Memorisation meets compositionality in natural language processing

Verna Dankers

Doctor of Philosophy
Institute for Language, Cognition and Computation
School of Informatics
University of Edinburgh
2025

Abstract

In deep learning, the perspective on memorisation of training examples is undergoing a paradigm shift. Previously linked to overfitting and poor generalisation, memorisation is now seen both as beneficial when it enhances deep neural networks' generalisation capabilities and as concerning when it comes to specific examples that should not be memorised. This shift raises questions about when memorisation is beneficial, what models memorise and should memorise, and how memorisation is implemented internally. Although these questions might be relevant for deep learning problems in general, I consider them to be particularly relevant for language learning and the field of *natural language processing* (NLP). After all, language itself requires both syntax-driven, generalisable meaning compositions *and* memorisation capabilities, thanks to its dichotomous nature of being both compositional – in terms of freely generated language – and non-compositional – due to the pervasiveness of fixed formulaic sequences.

This dissertation is divided into two parts, each studying memorisation in transformer models from a different angle. Within each part, I focus on the data first and then elaborate on model-internal mechanisms for memorisation. The first part examines memorisation broadly, identifying which examples require more memorisation, whether memorisation aids generalisation and where memorisation occurs in multi-layered models. Firstly, using the task of translation, various source-target language pairs and graded memorisation metrics, examples are placed on a 'memorisation map' to explore features predictive of high memorisation and their impact on model performance. Secondly, using classification tasks, memorisation localisation is examined at the level of the layers.

In the second part, I approach memorisation through the lens of natural language's compositionality, focusing on idioms as prime examples of non-compositional phrases requiring memorisation in neural networks. Using translation tasks, I analyse how models acquire idiom translations over the course of training while also monitoring models' compositional abilities. I then examine pretrained translation models for various source-target language pairs, separating idiom translations into paraphrases and word-for-word translations, and analysing the role of transformer's attention and changes to the hidden states in translating idioms non-compositionally.

By combining insights from data analysis and internal mechanisms, this dissertation examines the link between memorisation and generalisation. I firstly show that memorisation is not a mysterious phenomenon, but is predictable based on examples' features. Secondly, I establish that model-internal mechanisms for memorisation emerge in a dispersed manner: memorisation is implemented over a range of layers, and generalisation and memorisation capabilities are intertwined. Finally, I demonstrate that memorising certain training examples can aid generalisation, but also that models still face challenges with both compositional generalisation and non-compositional memorisation.

Lay Summary

When learning a language, we want students not to memorise too many specific sentences encountered in the course material. Instead, they should use the examples to learn the grammar and build up their vocabulary so that they can create new sentences in new contexts when speaking the language in the world. When humans use the meanings of words as building blocks and compose them to create sentences, we refer to that skill as *compositionality*. Yet, in some cases, memorising sentences from the course material is necessary to succeed; for instance, if someone learning English encounters the idiom "kick the bucket" for the first time, which means that someone has passed away. A phrase like that is what we call *non-compositional*, because the meanings of the words cannot be used as building blocks to understand the meaning of the phrase. As is the case with human students, when computational models learn language, we also do not want them to memorise too many specifics about the course material unless it will help the models understand the language after their initial training phase.

This thesis focuses on models' memories. I examine them in two separate parts, discussing model memories in general and memories of idioms in particular. In the first part, I firstly look at all of the data presented to a model as learning material when it learns to translate text, to study which of the examples are memorised the most, and whether this helps the model when presented with new examples after the training phase. Secondly, I look at the model itself, which consists of many parts and is organised in a layered way. Where in all of the model's layers does it memorise an example if we force the model to learn the wrong label? For instance, if we train the model to predict that "I love that you found a new job" expresses negative sentiment.

In the second part of the thesis, I focus specifically on the skill of compositionality and models' memories of idioms. I train models to translate text, monitoring when they learn idioms and how well they can build sentences by just treating the words as building blocks. Later on, I again look at the models themselves to understand which parts are the most important for memorising paraphrases of idioms during translation. For instance, if the model has to translate "My grandfather has kicked the bucket" as "Mijn grootvader is overleden" in Dutch, which internal mechanisms contribute to that?

Together, these experiments firstly show that it is quite predictable which sentences from the training material the model will or will not memorise. Secondly, they reveal that models' memories are not stored in one specific location but across many components spread throughout the model. Finally, they indicate that memorising certain examples from the training material can help translation models after the training phase. At the same time, when it comes to idioms, models have not memorised them well enough. Idioms are often translated word for word rather than being paraphrased, which leads to translation errors.

Acknowledgements

When I started my PhD back in September of 2020, the world was a different place, and if I look around beautiful Edinburgh today, it's a difference of night and day. Lockdown, quarantine, vaccination, and red lists all seem like alien concepts now. The best way to heal from the trauma of that first year, locked up in my new, foreign home, while attempting to be a researcher, was to go out and explore Scotland. If you're reading this and considering life in the UK as a PhD, postdoc, or otherwise, I can only say that when the highlands and the islands are calling: go. There is no way to recover from *CL deadlines like standing on the edge of Unst, Harris, or Skye.

Supportive from day one was the best supervisor and a phenomenal all-around person: Thanks, Ivan, for allowing me to pursue my own creative interpretability/compositionality/memorisation projects, even if they were not exactly what you expected when I arrived here in 2020. Your knowledge of the field is both broad and deep; your feedback can be in-depth and technical, or high-level and abstract, fine-tuned to what's needed. Meetings with you are always pleasant and inspiring, and it's been an honour to learn from you. I could use many, many more of those meetings.

A massive *dankjewel* to Dieuwke, my internship manager and long-time collaborator. I have jokingly called you my honorary supervisor, but it's not a joke that I owe you a lot. Your enthusiasm and optimism about research are unparalleled, and you've modelled for me that I should shoot for the moon. I've thoroughly enjoyed the many brainstorms, discussions, and laughs we've had over the years, for manifesto (2.0), my internship paper, and GenBench. You are a huge inspiration to me.

Supportive from the sidelines has been my secondary supervisor, Chris Lucas. Chris taught me to always return to the overarching story and the underlying research questions before getting lost in technical details or results. My data annotation studies have been successful mainly because of your advice! Thanks to Chris, Kenny Smith and Adam Lopez for reviewing my yearly progress and steering me in the right direction for how to design the thesis's storyline.

Many thanks to Tal Linzen and Edoardo Ponti for reading these pages in June of 2025 and for the fruitful discussions during my unique virtual viva about previous, current, and future research. Thanks for challenging me about what memorisation even is, critically assessing the experimental choices I've made, and discussing the implications of my research.

Thanks to the collaborators I've had over the years: the co-authors of my CoNLL 2021 paper (Anna, Kate, Adina), manifesto 2.0 (Elia), the students that I supervised, among whom Maike and Márk, the GenBench NMI team, my GenBench and L2M2 workshop co-organisers and my wonderful Microsoft manager, Vikas, who supported me throughout the unlucky summer of 2024.

PhD journeys are rocky roads, and mine is no exception. I owe much to the travel companions I met in Edinburgh along the way; you're all gems of humans. Thanks to Agostina and Matthias for long lunches, for fun procrastination sessions about random topics, for sharing the day-to-day, for great memories made during conferences, for feedback on write-ups and presentations, and for giving a boost whenever imposter syndrome came along. Thanks to Henry for teaching me about language evolution and compositionality over coffee. Thanks to Victor for the best coffees and brownies, which celebrated good days and cured bad days. Thanks to Aida for beautiful conversations and bonding over wine and chocolate. Thanks to Syd and Ivan for inspirational and enthusiastic "compositionality is a continuum" research and the fanciest dinner parties. Thanks to the office mates I've had over the years, particularly Radi for her joyful presence, Wanqiu for gorgeous Chinese souvenirs and lovely conversations, and Anil for co-working during evenings and weekends. Thanks to the CDT in NLP (and Data Science CDT) crowd, among whom Amr, Coleman, Gautier, Giulio, Laurie, Nick F., Nikita, Rim, Rohit, Steph, Tom and Zheng. You managed to make every event more enjoyable, because I knew I'd see your friendly faces, both on (Forum/Wilkie lunch, CDTalks, ILCC cookies, etc.) and off campus (Firbush, writing retreat, one-on-one covid meetups in the early days of my PhD, etc.). Thanks to Sally for supporting the CDT with so much enthusiasm and answering countless questions of mine over the years. Thanks to Ivan's team for great group meetings and pub nights, both regular attendees, including Antonio, Bailin, Dennis, Guillem, Jason, John, Masaru, Mattia, Neel, Pedro, Rochelle, Xinnuo, and Zeyu, and visitors, including Hosein and Max; I learnt a lot from you. Thanks to my fellow research interns, whom I met in Paris in 2022 and Seattle in 2024, and Ric in particular. Thanks to Siva Reddy's group for welcoming me with open arms and celebrating my viva and the final weeks of my PhD with me; I look forward to the many new (intellectual) adventures we'll have together.

A shout-out to friends that supported me from afar, especially Martin, Matthijs and Amarens, and to my family. *Lieve pappa en mamma, Janine, Gerdine, Arina, iedereen die bij jullie hoort en mijn schoonfamilie: bedankt voor jullie steun vanuit Nederland. Het is spannend om je dochter/zusje te zien emigreren en moeten missen, terwijl ze haar eerste stappen zet in een voor jullie onbekende, academische wereld. Ik houd super veel van jullie, en ben heel trots om hierna doctor Dankers te zijn.*

Finally, I'd like to express my gratitude to my love. Wietze, more than a decade ago, it was you who suggested I study AI, and it's safe to say that your suggestion was spot on. There are no words that would do justice to your support. Thanks for crossing seas and oceans just to be with me, for all the care, for all the love, and for being much more AI-optimistic than I am. You are one of a kind.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Verna Dankers*)

Table of Contents

Chapter 1

Introduction

The field of deep learning is showing evidence of a paradigm shift when it comes to the role of *memorisation* in the development of computational models. Traditionally, in machine learning, memorisation was intricately linked to overfitting and associated with a lack of generalisation capabilities. In 1995, Dietterich (1995, p.326) already discussed the concern of "fit[ting] the noise in the data by memorising various peculiarities", and the development of active schemes to combat overfitting is still alive and kicking (Bejani and Ghatee, 2021). Yet, in recent years, deep neural networks trained on ever-growing datasets have shown that strong generalisation to evaluation data can coincide with the memorisation of training data, a situation referred to as *benign overfitting* (Bartlett et al., 2020). This shift has become particularly salient within the field of *natural language processing* (NLP) with the rise of *large language models* (LLMs), for which generalisation and memorisation capabilities appear to increase together (Carlini et al., 2022; Biderman et al., 2023). It has been suggested that excelling at tasks whose data is characterised by long-tailed distributions simply requires memorisation of atypical (but correct) training examples for optimal generalisation capabilities (Feldman, 2020; Feldman and Zhang, 2020; Zheng and Jiang, 2022), and LLMs are expected to memorise factual information about the world (Lee et al., 2022; Zhao et al., 2024b).

If memorisation is not necessarily something to combat, that introduces a range of new questions, such as: (When) is memorisation beneficial? What do LMs memorise, and what should they have memorised? Which neural mechanisms enable memorisation, and how can we improve those mechanisms? While these questions might be relevant for deep learning problems in general, I consider them to be particularly relevant for NLP. Language learning requires modulating analytic processing and memorisation capabilities, thanks to natural language's dichotomous nature of being both compositional and non-compositional. It is compositional in terms of freely generated language, and non-compositional due to the pervasiveness of fixed, or *formulaic*, sequences, such as proverbs and idioms (e.g. Svensson, 2008).

In this dissertation, I investigate memorisation in computational models of language, both as a general phenomenon and through the lens of formulaic sequences, by using non-compositional idiomatic expressions for memorisation case studies. I also study the relation between memorisation and generalisation, to better understand to what extent they are at odds with one another. In this section, I first introduce the main topics under discussion to the reader, by providing an overview of the current NLP landscape in terms of memorisation studies (§1.1) and laying out how memorisation and compositionality are connected for natural language (§1.2). This is followed by a thesis outline (§1.3) detailing the research questions under investigation and a summary of my findings. Lastly, §1.4 lays out how the chapters relate to my published articles.

1.1 Memorisation in NLP and beyond

Deep learning and the pretraining of deep neural networks (as opposed to training them from scratch) were catalysts for investigations concerning memorisation, particularly in the last five years. But to the one label of 'memorisation', many different meanings and interpretations have been assigned. I identify at least four distinct, albeit related, ways in which memorisation has been discussed that are relevant to keep in mind throughout the thesis: Firstly, in deep learning articles, in general, memorisation has been used as a synonym for (1) **training set interpolation** (e.g. Zhang et al., 2017; Arpit et al., 2017; Chatterjee, 2018) – i.e. memorisation of entire training sets – and (2) as a **model descriptor**, used to distinguish memorising networks from generalising networks, based on how well they generalise to evaluation data. Memorising networks have been trained on purpose, using manipulated data (e.g. Morcos et al., 2018), but also naturally emerge. For instance, during *grokking*, networks interpolate the training set, but still morph from memorising networks into generalising networks through training beyond the point of training loss convergence (Power et al., 2022). The observation that training set interpolation can occur in generalising networks was the one that caused Bartlett et al. (2020) to first coin the term *benign overfitting*, and instigated work in understanding the circumstances under which overfitting is, in fact, benign (Sanyal et al., 2020). Work in this direction is of a more theoretical nature, often adopting *computer vision* (CV) tasks. It is less prominent in contemporary NLP, because the vast sizes of pretraining corpora do not allow LLMs to fully interpolate that data. Nonetheless, it is relevant to our discussions because this work sets the stage for the question of the extent to which memorisation and generalisation are at odds with one another.

At a more fine-grained level, (3) **data memorisation** of individual training examples has been discussed, and this work is more prominent in contemporary NLP studies where models memorise only a subset of the (pre)training data. Work in this direction has focused on developing metrics to quantify memorisation, understanding which models or training techniques yield more data memorisation (e.g. Carlini et al., 2022; Mireshghallah et al., 2022; Biderman et al., 2024), and analysing what types of examples are memorised (e.g. Carlini et al., 2021; Zheng and Jiang, 2022; Zhang et al., 2023). Data memorisation can be considered concerning, benign or beneficial depending on what is memorised. Verbatim memorisation of copyright-protected text (Chang et al., 2023) or *personally identifiable information* (PII) (Huang et al., 2022), for instance, is concerning. Other types of verbatim memorisation can be benign (e.g. in case of canonical text from copyright-free news headlines or wiki entries, Carlini et al., 2021), and memorisation of factual information (Geva et al., 2023) or atypical but accurate examples (Zheng and Jiang, 2022) is beneficial.

Finally, the (4) **implementation of memorisation** in deep neural networks has drawn attention, resulting in work studying how memorised examples affect models internally. Some articles focus on pinpointing individual layers or neurons that encode memories (e.g. Dai et al., 2022; Chang et al., 2024; Stoehr et al., 2024), whereas others mean to provide a more comprehensive explanation of how multiple layers or multiple layer subcomponents (i.e. transformer's self-attention and feedforward modules) cooperate to retrieve memories (Haviv et al., 2023; Geva et al., 2023). In addition to localising memories, there is work on editing them (e.g. De Cao et al., 2021; Meng et al., 2022, 2023), which often focuses on the feedforward modules as those are assumed to encode models' memories following seminal work by Geva et al. (2021). To this day, however, it is unclear whether memories tend to be stored in specific layers of a model; there are conflicting results from both CV and NLP, where there was initially a consensus that deeper layers of deep neural networks were more involved in storing memories than earlier layers (e.g. Stephenson et al., 2021; Cohen et al., 2018; Dai et al., 2022). More contemporary work, however, puts more emphasis on earlier layers and the fact that memories may be encoded in a distributed rather than a localised manner (e.g. Geva et al., 2023; Haviv et al., 2023; Maini et al., 2023).

This brief overview of the literature demonstrates the multi-faceted nature of memorisation, and I further elaborate on it in §2.2. Paradoxically, memorisation is desirable for some datapoints and undesirable for others. Developing computational models of language, therefore, involves both encouraging memorisation where needed, and favouring learning strategies that generalise, otherwise. In the thesis, I contribute to a better understanding of both data memorisation and the implementation of memorisation, using datapoints that models *do* memorise, while also noticing that for other examples (specifically those containing formulaic language) models lack memorisation. In the next subsection, I further elaborate on the connection between memorisation and formulaicity.

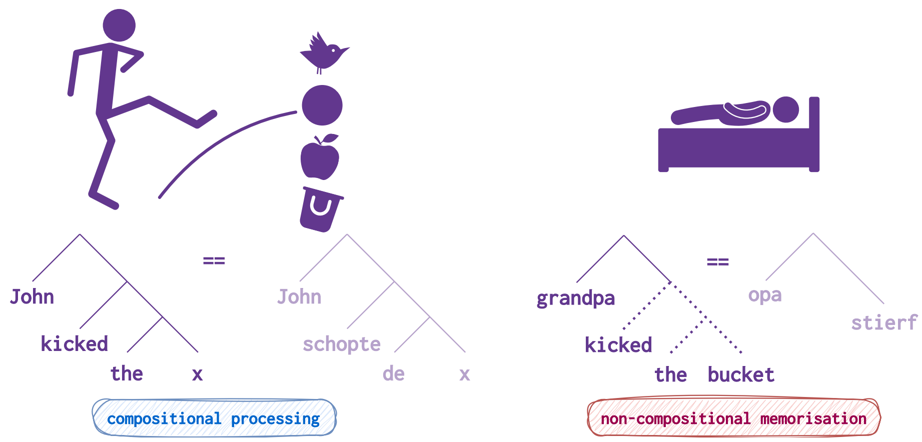


Figure 1.1: An illustration of the relation between compositionality and memorisation: compositional processing of "<person> kicked the <object>" explains how language can be used productively. Yet, exceptions exist where phrases need to be memorised as one unit instead of being decomposed according to the syntax, as is the case for "grandpa kicked the bucket". This affects NLP tasks, such as translation, as demonstrated in the illustration using Dutch translations.

1.2 Memorisation and compositionality

Compositionality is a property that a language may have or may lack. Many arguments for why natural languages would have this property have been put forth, among which are the systematicity and productivity of natural languages (Fodor, 1987), but also their learnability and intersubjectivity (Pagin and Westerståhl, 2010b). Speakers of a natural language can produce sentences they have never heard before by recombining words or phrases they know in a new way, and other speakers of that language would understand those sentences, thanks to the compositionality of language. Partee (1984, p.153) proposed one of the most well-known definitions of **the principle of compositionality**:

> *The meaning of an expression is a function of the meanings of its parts and of the way they are syntactically combined.*

At first glance, this definition may suggest that language is a little bit like arithmetic: we can determine the meaning of a linguistic expression based on the meanings of the inputs (words) and the operators and structure (syntax) of an expression, as visualised in Figure 1.1 for "John kicked the <object>". The same operators describe the meanings when inserting different words, such as "ball", "bucket", "apple" or "bird". To understand why that is somewhat naive, let us first consider a thought experiment.

A compositionality thought experiment Think of a story you recently read in the newspaper. Now imagine a pupil from a different time and place who knows little about the world's celebrities or historical events. They do not speak your language, but have learnt the language of this story by memorising the dictionary and a grammar guide. They might grasp much of the story because they can use the words' meanings and the structure of the text, as they taught themselves. Still, they will unavoidably run into parts that cannot be explained by stringing together the meanings of individual words. Numerous issues come to mind, but the primary one is that words do not function in isolation: At the extra-sentential level, *contextuality* shapes meaning beyond what a dictionary can convey – cultural, temporal, and situational factors all play a role. At the intra-sentential level, words do not function in isolation either since many expressions derive meaning from being a part of a fixed group, e.g. for prototypical formulaic multi-word expressions ("stealing my thunder"), but also named entities ("Purgatory Pool", an English lake), or even entire quotes from Shakespeare's plays. The pupil would struggle to understand or translate our text without the wider context and more knowledge about such groups of words. As Fillmore et al. (1988, p.504) notes: "an idiomatic expression or construction is something a language user could fail to know while knowing everything else in the language". This is where **non-compositionality** comes into play. Returning to Figure 1.1, our hypothetical pupil might understand the literal meanings, but would likely miss the non-compositional reading of "grandpa kicked the bucket" as "grandpa passed away".

In spite of such counterexamples, the fact that natural language is largely compositional is undeniable, and this property has long intrigued NLP researchers. Ideally, computational models of language would mirror language's compositional nature. Studies that analyse whether NLP models do so refer to the type of evaluation they perform as **compositional generalisation** evaluation.[1] The most appropriate definition of compositionality has been widely discussed in linguistics and the philosophy of language, because of the many counterexamples that exist in natural language (Zadrozny, 1994; Horwich, 2001; Pagin and Westerståhl, 2010b; Szabó, 2012; Baggio et al., 2012, i.a.). These definitions vary in strictness to accommodate counterexamples; I return to this literature and related work on compositional generalisation in §2.3.1 and §2.3.2, respectively. My goal is not to resolve this debate, but to emphasise that not all of natural language is compositional. In many ways, NLP models are like our hypothetical pupil: trained on decontextualised text, broken up into words. As NLP researchers, we assume end-to-end neural training on text corpora allows models to capture the richness of language, but whether and how it does remains an open question – one to which this thesis contributes.

The pervasiveness of formulaicity Compositionality explains how infinitely many new sentences can arise from a finite syntax and semantics. However, its importance may be

[1] Although the term was not used until 2018 by Loula et al., the concept had been used in cognitive science decades prior. Systematic generalisation and compositional generalisation are sometimes used interchangeably, with the latter now being more commonly used than the former.

overstated when considering actual language use. Of all possible sentences, there is only a small proportion that native speakers would actually utter (Pawley and Syder, 1983), partly due to the prevalence of **formulaic sequences**, defined by Wray (2002, p.9) as:

> *A formulaic sequence is a sequence, continuous or discontinuous, of words or other elements, which is, or appears to be, prefabricated: that is, stored and retrieved whole from memory at the time of use, rather than being subject to generation or analysis by the language grammar.*

To account for the pervasiveness of formulaic sequences alongside compositional text, dual-system models have been proposed (Wray, 1992), suggesting that compositional processing in the brain competes with *holistic processing*, which relies on prefabricated strings stored in memory. Systems of this kind have been supported by cognitive evidence in both healthy subjects (e.g. Sidtis et al., 2018) and clinical populations (e.g. Sidtis et al., 2009; Zimmerer et al., 2016; Torrington Eaton and Thomas, 2024), pointing towards the presence of different neural substrates for the two types of processing. The prominence of holistic processing means that language production and comprehension lean heavily on humans' memory, and motivates taking a closer look at memorisation during computational modelling of language, as well. While memorisation is not sufficient for adequately capturing the non-compositional nature of language – i.e. it does not actually explain all issues outlined during the thought experiment above, or compositionality problems listed in the literature (Pagin and Westerståhl, 2010b) – it is a necessary component of language learning, in humans and machines alike. When exposing our pupil to tasks that involve natural language understanding, such as translation, memorisation is needed on top of compositional processing, to memorise that an idiom needs to be translated as a unit instead of decomposing it according to syntax (as illustrated in Figure 1.1). Formulaic language has been the proverbial pain in the neck of NLP for decades because models have struggled with acquiring non-compositional meanings (e.g. Sag et al., 2002; Rayson et al., 2010; Shwartz and Dagan, 2019). A wide body of research has been dedicated to understanding to what extent computational models can adequately detect and understand formulaic phrases, and use them in downstream tasks. I return to related work in this direction in §2.3.3.

1.3 Thesis outline and research questions

In this thesis, I will first focus on memorisation (and generalisation) in generic terms (Part I), followed by a case study around the (non-)compositional nature of natural language (Part II). In the process, two types of tasks will be used: firstly, the sequence-to-sequence task of *neural machine translation* (NMT). NMT models are trained on handcrafted or automatically sourced training corpora representative of the natural variation observed in language, and compositionality is traditionally well-studied and motivated for MT (Rosetta, 1994; Janssen and Partee, 1997; Janssen, 1998). When working with idiomatic expressions, NMT is of particular interest because different languages have very different formulaic sequences, ensuring that idiom translation *must* involve memorisation. In addition to NMT, I also use generic language understanding classification tasks in the scenario where higher control over the training data is desired. All analyses adopt the transformer architecture (Vaswani et al., 2017), albeit in different setups, varying training set sizes, model sizes and whether or not the model is pretrained. In §2.1, I review the fundamentals of this architecture and the tasks and models that are considered throughout the thesis.

Together, the different chapters from the two parts of this thesis will contribute to answering the following research questions:

RQ1. *What characterises memorised examples?*

RQ2. *Which model-internal mechanisms enable memorisation?*

RQ3. *To what extent are memorisation and generalisation at odds with one another?*

Part 1. Memorisation in transformer This first part focuses on memorisation in generic terms: within a dataset, some examples require more memorisation than others. Which examples do models memorise, and where in these multi-layered networks can memorisation be localised?

1. *The data viewpoint* (chapter 3): For the task of NMT, I study data memorisation to understand what types of examples transformer memorises most. Adopting graded memorisation metrics (as opposed to binary ones), all training examples are positioned on a 'memorisation map' for five source-target language pairs, comprising a resource of memorisation metrics which could be used by future work. I investigate which surface-level features are predictive of high memorisation scores, identifying that these are mostly datapoints showing natural variation, and are not simply noise (RQ1). Features that are predictive of memorisation in one language predict memorisation in other languages well, too. A brief intermezzo is included, foreshadowing findings discussed in Part II, to establish that examples with formulaic phrases are memorised less than control stimuli. Finally, I establish that examples with high memorisation scores are beneficial for models' generalisation to unseen data (RQ3).

2. *The model viewpoint* (chapter 4): Afterwards, I shift focus to localising memorisation in transformer. To do so, four LMs, four localisation methods, a new visualisation technique (*centroid analysis*), and twelve natural language classification tasks are used, for which a subset of the labels is perturbed to perform

layer-wise memorisation localisation when fine-tuning the LMs. This controlled setup ensures that the examples under investigation are, in fact, memorised. I identify that memorisation cannot be localised to individual layers, but is a co-operative process of many layers, through which memorised examples gradually become more distinct (RQ2). Contrary to what previous work suggests, models' deepest layers do not play a special role in that. Which layers are most involved is, however, dependent on the task being inspected, and task difficulty correlates with where memorisation is located most.

Part 2. (Non-)compositionality: a memorisation–generalisation case study In the second part, I shift focus to the topic of compositionality, both evaluating compositional generalisation and the processing of idiomatic expressions. How does this reflect the tension that exists between memorisation and generalisation, and how does memorisation of idioms affect models internally?

1. *Evaluating (non-)compositional generalisation* (chapter 5): In this chapter, I focus on compositional generalisation first, and memorisation of idioms second. Compositionality is often studied using artificial datasets, in which a strict, bottom-up approach to composing meaning is guaranteed to be successful. I, instead, redefine three tests from the literature for the evaluation of the (non-)compositionality of transformer NMT systems trained on *natural* data. I focus on one source-target language pair, introducing new evaluation data and adopting it for our tests. The results indicate that NMT systems, paradoxically, both do not exhibit enough compositional generalisation when it comes to robustness to input perturbations, while also being *too compositional* in other cases, for instance, in the case of idioms (RQ3). During training, idiomatic translations are acquired in two phases: an overgeneralisation and memorisation stage. Yet, for many idioms, NMT systems remain in the overgeneralisation stage if the model has finished training, exemplifying one way in which memorisation and generalisation are at odds with one another.

2. *Mechanisms for idiomatic translations* (chapter 6): Afterwards, I solely focus on idioms, analysing how transformer NMT systems for seven source-target language pairs translate them, establishing that they are often translated overly compositional. To determine this, a heuristic way to distinguish (non-compositional) paraphrased translations from (compositional) word-for-word translations is leveraged. I point out that frequency influences how idioms are translated (RQ1). This is followed by a range of analyses that study transformer's internal mechanisms that enable paraphrased translations of idioms (RQ2). I discuss the role of the encoder's attention, the encoder-decoder cross-attention and hidden states' trans-

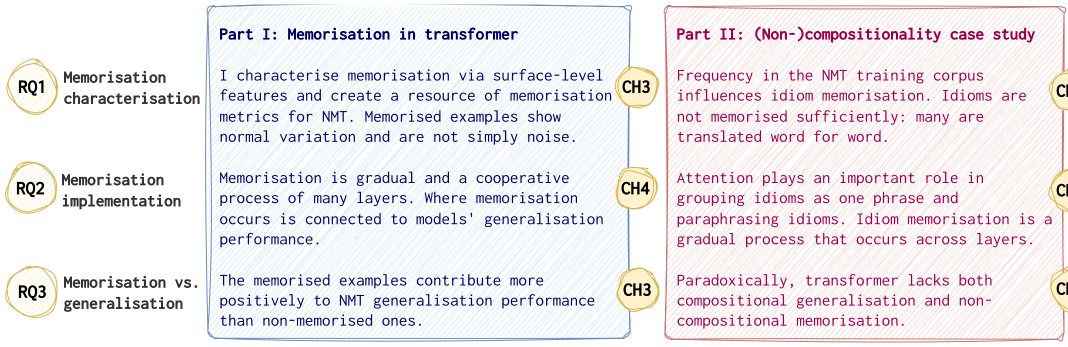


Figure 1.2: An overview of how the different chapters address the three central research questions, both for memorisation in general (in blue) and for our (non-)compositionality case study (in red).

formations over layers. Key findings are that when paraphrasing figurative idioms, transformer's encoder is found to group words within the phrase more strongly compared to literal phrases, and that transformer's decoder processes the input in a way that is partially disconnected from the encoder. In line with results from chapter 4, memorisation is found to be a gradual process in which many layers cooperate to set idiomatic examples apart.

Figure 1.2 succinctly summarises the contributions and conclusions from the different chapters for each of the research questions under investigation. I conclude that although transformer models retain substantial information from their training data – which is useful for generalisation to unseen examples – they still fall short in capturing formulaic language, and do not adequately process inputs compositionally, where possible, and non-compositionally, where needed. Although memorisation mechanisms do arise naturally as a cooperative process of transformer's many layers, these mechanisms are insufficient for many idiomatic expressions. In chapter 7, I will further reflect on these findings and propose directions for future work.

1.4 Published works

The chapters in this thesis are primarily based on the following papers:

- **Chapter 3:** Verna Dankers, Ivan Titov, and Dieuwke Hupkes. 2023. Memorisation cartography: Mapping out the memorisation-generalisation continuum in neural machine translation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8323–8343
  Dieuwke Hupkes (DH) and Ivan Titov (IT) provided supervision and contributed to conceptualisation and deciding on the methodology. DH and IT contributed to paper writing through reviewing and editing. DH contributed to data analysis. I developed the software, executed experimentation, analysis and visualisation, and wrote the paper. The work was partially conducted during an internship at Meta.

- **Chapter 4:** Verna Dankers and Ivan Titov. 2024. Generalisation first, memorisation second? Memorisation localisation for natural language classification tasks. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 14348–14366
  Ivan Titov (IT) provided supervision, contributed to conceptualisation, deciding on the methodology and paper reviewing and editing. I developed the software, executed experimentation, analysis and visualisation, and wrote the paper.

- **Chapter 5:** Verna Dankers, Elia Bruni, and Dieuwke Hupkes. 2022a. The paradox of the compositionality of natural language: A neural machine translation case study. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4154–4175
  Dieuwke Hupkes (DH) and Elia Bruni (EB) provided supervision and contributed to conceptualisation and deciding on the methodology. DH trained the NMT systems evaluated in this chapter, contributed to the manual data analysis and paper writing. EB contributed to paper reviewing and editing. I created the new evaluation datasets, developed the software for the analysis and visualisation, executed experimentation and contributed to paper writing.

- **Chapter 6:** Verna Dankers, Christopher Lucas, and Ivan Titov. 2022b. Can transformer be too compositional? Analysing idiom processing in neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3608–3626
  Ivan Titov (IT) and Christopher Lucas (CL) provided supervision and contributed to conceptualisation and deciding on the methodology. CL supervised the data collection. IT contributed to paper reviewing and editing. I developed the software, set up the data collection infrastructure, executed experimentation, analysis and visualisation, and wrote the paper.

In addition to the articles listed above, I was the first author of the following publications throughout my PhD:

- Verna Dankers*, Anna Langedijk*, Kate McCurdy, Adina Williams, and Dieuwke Hupkes. 2021. Generalising to German plural noun classes, from the perspective of a recurrent neural network. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 94–108. *Equal contribution

- Verna Dankers and Ivan Titov. 2022. Recursive neural networks with bottlenecks diagnose (non-)compositionality. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4361–4378

- Verna Dankers and Christopher Lucas. 2023. Non-compositionality in sentiment:

New data and analyses. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5150–5162

- Verna Dankers and Vikas Raunak. 2025. Memorization inheritance in sequence-level knowledge distillation for neural machine translation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 760–774

Chapter 2

Background

Throughout the thesis, transformer models (Vaswani et al., 2017) are examined through the lenses of memorisation and compositionality. Before diving into new results, let us review the background on these topics. Firstly, in §2.1, by reviewing the composition of the model architectures employed, the tasks that we will consider and the wide range of interpretability methods we will later rely on. Afterwards, I go over related work on memorisation (§2.2), and we end with background information on the continuum of (non-)compositionality in natural language, and what is known about how computational models and humans balance compositional with non-compositional processing (§2.3).

2.1 A primer on transformer and tasks

NLP at large heavily relies on **language modelling**: the task of estimating a probability distribution over sequences of words. I will more formally define it in §2.1.2, but what is relevant for now is that LMs are predominantly trained using next-word prediction, where they are provided with examples of the form (x_1^n, y). Given the context of length n (consisting of $x_1, x_2, \ldots, x_n$), they are trained to predict the next word y, processing text in the order of reading. In **sequence classification** tasks, the objective is to estimate a conditional probability distribution over abstract classes given an input sequence, where the total number of classes is typically small – e.g. when performing sentiment classification, these classes could be 'negative', 'positive' and 'neutral'. Training pairs are of the form (x_1^n, y), with y being an abstract target instead of the next token. In the task of **machine translation**, on the other hand, we train models to estimate a conditional probability of a translation, based on an input sequence. Training pairs are of the form (x_1^n, y_1^m), containing an input sequence x_1^n in one language (the **source language**) and translation y_1^m in another language (the **target language**). For these tasks, examples are typically created in an automated way for language modelling – since one can simply take existing text excerpts and break them down into $(x_1^n,\ y)$ pairs –

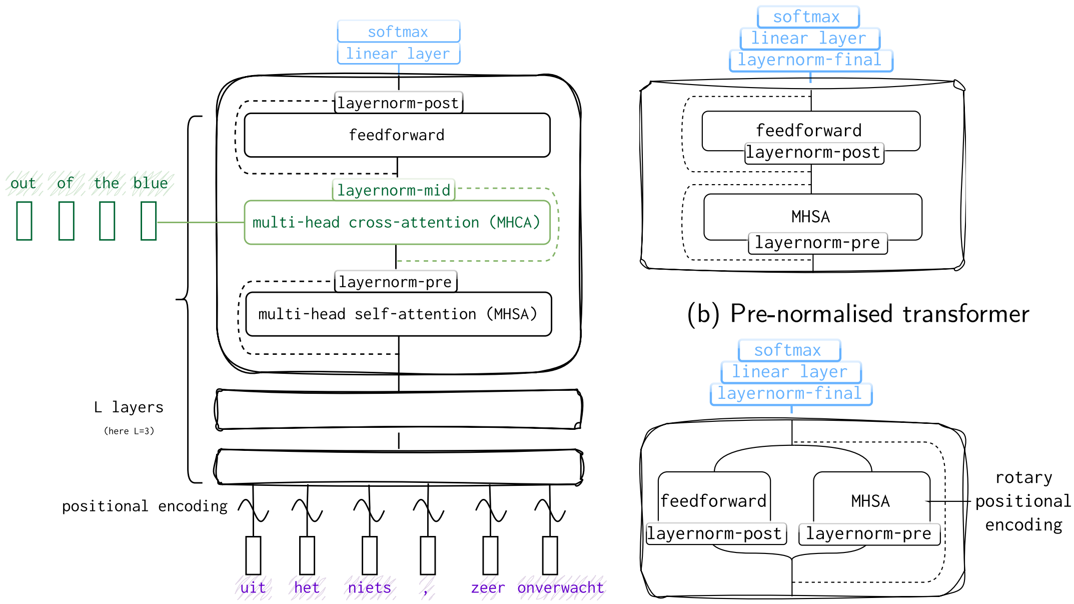
and in a semi-automated or manual way for MT and natural language classification – e.g. by collecting examples from content creators on the web, or asking human annotators to provide translations or label examples. We then use a computational model Θ to fit the task-specific probability distribution based on these training examples.

2.1.1 Transformer

We return to the individual tasks in §2.1.2, but first focus on the model architecture dominant in NLP since its presentation in 2017 by Vaswani et al.: *transformer*. Transformer is used throughout this thesis in various forms.

Encoding a sequence of words Before transformer can process $x_1^n \in V_x$, we first separate the sequence into tokens using a **tokenisation scheme** that maps every natural language word to a sequence of symbols from the vocabulary of the model V'_x. This vocabulary typically contains tokens representing words, but also subwords and even individual letters or symbols, such that new words can be processed without encountering out-of-vocabulary issues. Words that are frequent in the training corpus would often be assigned their own token, whereas less frequently occurring words are more likely to be segmented more. For instance, "out of the blue" would be tokenised as "_out _of _the _blue" but "this is gobbledygook" would be processed as "_this _is _go bb led yg ook". The main families of tokenisation schemes are based on *Byte-Pair Encoding* (BPE) (Sennrich et al., 2016), and Unigram LMs (Kudo, 2018). BPE builds a subword vocabulary by repeatedly merging frequent character pairs, while the Unigram LM prunes a large set of candidate subwords using a probabilistic model to choose likely segmentations. The vocabulary also contains special tokens (such as the *beginning-of-sentence* (BOS) and *end-of-sentence* (EOS) token) which can be used to explicitly delineate parts of the input. Transformer stores d-dimensional representations for each token in a separate row of a vocabulary weight matrix $W_{V'_x} \in \mathbb{R}^{|V'_x| \times d}$ and stores representations for positional indices in which tokens can occur in a sequence with a max length of I in a positional weight matrix $W_{P_x} \in \mathbb{R}^{I \times d}$. Together, these weights represent **the input embedding layer**, parametrising θ_{Emb}. After tokenisation, transformer encodes the input in a high-dimensional space $\mathbf{x}_1^n \in \mathbb{R}^{n \times d}$ by simply summing the rows that correspond to the tokens and their positions.[1] Vaswani et al. originally considered both learning the positional weight matrix and fixing the weights up front by encoding positions using sine and cosine functions of different frequencies.

[1]Where, for simplicity, I assume length n does not change post tokenisation here. By convention, matrices are denoted with uppercase italicised letters. Note, however, that I use lowercase bold symbols to denote the matrix of hidden representations for multiple tokens, to emphasise the relationship to individual hidden representations.



(a) Transformer as per Vaswani et al.

(b) Pre-normalised transformer

(c) Parallelised transformer

Figure 2.1: Schematic overview of transformer, adapted from Vaswani et al. (2017), along with two layer variants used in the thesis. Because layer variants (b) and (c) are only used without MHCA, we omit the green block there. Dashed lines indicate residual connections.

The sequence passes through the model Our encoded sequence now passes through a composition of L transformer layers, where each layer has its own parameters, and the output of layer l feeds into layer $l+1$. Figure 2.1 depicts transformer, along with two layer variants employed by models in the thesis. We will introduce these variants and other modifications that apply to models used in the thesis later on, when discussing the model component to which they apply.

One individual transformer layer, as per the original definition by Vaswani et al. consists of four or six components, depending on whether the current transformer is stand-alone or whether it receives input $\mathbf{s}_1^m$ from another transformer (we mark modifications required for the latter in dark green in equations and figures).[2] When two transformers are chained, they are referred to as the **encoder** and **decoder**, respectively. A stand-alone transformer is referred to as an encoder-only or decoder-only model.

$\mathbf{x}_1^n$ firstly passes through a *multi-head self-attention* (MHSA) module, followed by a normalisation operation with a residual connection. This ensures that tokens are contextualised within the input sequence they are a part of, and that their representations are a combination of the current layer and the previous layer. If transformer receives input from another transformer, that input $\mathbf{s}_1^m$ then passes through the *multi-head cross-attention* (MHCA) module, followed by another normalisation operation with a

[2]The introductory paragraph referred to the source and target sequences as x_1^n and y_1^m. To discuss how various architectures rely on the same underlying transformer layer, we now refer to the input of the current transformer as x_1^n, temporarily referring to the source sequence from another model as s_1^m. We return to referring to these two sequences as x_1^n and y_1^m in the next subsection.

residual connection. This ensures that representations for the input are combined with relevant prior context (e.g. the source sequence in NMT). Finally, the representations go through a feedforward layer that enriches the now contextualised representations. This is followed by another normalisation operation with a residual connection, again ensuring representations are a combination of previous (sub-)layers and the feedforward layer. The paragraphs that follow further detail these components using equations.

Multi-head self-attention In the MHSA module, the representation of a current token in position i ($\mathbf{x}_i$) is combined with all tokens through a weighted sum of tokens' representations in the self-attention function A. Self-attention involves linear projections Q, K and V (producing query, key and value vectors) that transform their inputs into k-dimensional vectors ($k = \frac{d}{h}$). Self-attention compares the query vector for token i to the key vectors for all tokens, computing attention weights using the softmax based on that comparison. Using the weights, the new representation for token i is a mixture of the value vectors. This is performed h times, where each time is referred to as an attention head. The outputs of heads are combined by concatenation, after which a final output projection is applied (W_O). The full module is detailed in Equations (2.1)-(2.5).

$$\mathbf{u}'_i = [\mathrm{A}^{(1)}(\mathbf{x}_1^n, \mathbf{x}_1^n)_i; \ldots; \mathrm{A}^{(h)}(\mathbf{x}_1^n, \mathbf{x}_1^n)_i] W_O \qquad \texttt{MHSA} \qquad W_O \in \mathbb{R}^{hk \times d} \tag{2.1}$$

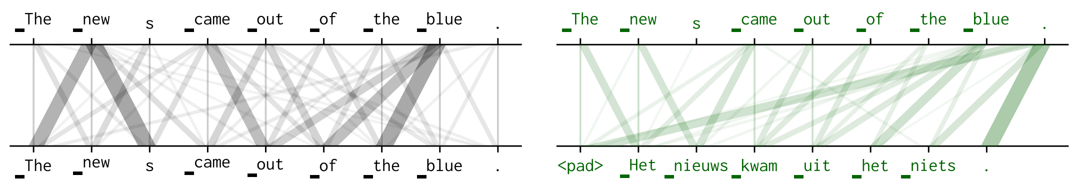
$$\mathrm{A}(\mathbf{x}_1^n, \mathbf{z}_1^m) = \mathrm{softmax}\left(\frac{\mathrm{Q}(\mathbf{x}_1^n)\mathrm{K}(\mathbf{z}_1^m)^\top + M}{\sqrt{k}}\right)\mathrm{V}(\mathbf{z}_1^m) \qquad \texttt{attention} \qquad M \in \mathbb{R}^{n \times m} \tag{2.2}$$

$$\mathrm{Q}(\mathbf{x}_1^n) = \mathbf{x}_1^n W_Q \qquad \texttt{query vectors} \qquad W_Q \in \mathbb{R}^{d \times k} \tag{2.3}$$

$$\mathrm{K}(\mathbf{x}_1^n) = \mathbf{x}_1^n W_K \qquad \texttt{key vectors} \qquad W_K \in \mathbb{R}^{d \times k} \tag{2.4}$$

$$\mathrm{V}(\mathbf{x}_1^n) = \mathbf{x}_1^n W_V \qquad \texttt{value vectors} \qquad W_V \in \mathbb{R}^{d \times k} \tag{2.5}$$

Vaswani et al. were not the first to introduce an attention mechanism; it had previously been proposed in NMT by Bahdanau et al. (2015) as a mechanism that allowed a model to selectively focus on specific words on the source side when translating a word on the target side. Yet, Vaswani et al. were the first to build a model that only has attention as a mechanism for token-mixing – as opposed to more traditional recurrent models, that processed tokens in order, combining consecutive tokens through weighted gating mechanisms (e.g. Hochreiter and Schmidhuber, 1997). Generally, the role of MHSA is to combine tokens that are related in various ways, such as tokens that are close together in the input (e.g. Raganato and Tiedemann, 2018; Clark et al., 2019b; Wang et al., 2022; Ferrando and Voita, 2024) or tokens that are syntactically or semantically related (e.g. Clark et al., 2019b; Chen et al., 2023). More niche roles have also been identified, such as heads attending to rare words (Voita et al., 2019c). Although post-hoc analyses have identified such roles, the MHSA develops as the model is trained, without researchers having control over or absolute clarity about what the attention captures.



(a) Self-attention (b) Cross-attention

Figure 2.2: Illustrative example of attention between tokens in transformer, for MHSA and MHCA. The underlying data is averaged over layers for the En-Nl NMT model used in chapter 6, with edges below 0.05 suppressed. EOS tokens are omitted to improve visibility.

For a concrete example of attention between tokens, inspect Figure 2.2a, where "the news came out of the blue" is captured using (sub)words, with attention here averaged over the layers of a model from chapter 6 of the thesis. Line thickness indicates the magnitude of the attention weight: we observe a clear impact of proximity, where tokens that are close together in the input attend more to one another, and also observe an effect at the phrase level, since "_out _of _the" all strongly attend to "_blue".

Modification 1 Equation (2.2) contains the masking matrix M. The masking matrix can be used to suppress interactions between tokens that do not satisfy the causality property; $M_{i,j}{=}0$ if $j \leq i$ and is set to a large negative number, otherwise (note that in the MHSA $n{=}m$). Without masking, transformer is considered bidirectional, whereas with masking, it is referred to as being autoregressive (e.g. in `GPT-2`, Radford et al., 2019). Vaswani et al. only used the mask in the decoder of their encoder-decoder NMT model. A stand-alone, bidirectional transformer is referred to as an encoder-only transformer; a stand-alone, autoregressive transformer is referred to as a decoder-only transformer.

Modification 2 In the original formulation, every token can attend to every other token (or every token up to i in the autoregressive transformer). Child et al. (2019) introduced multiple variants of sparse attention, where tokens can only attend to other tokens within a certain window. Some architectures, such as `GPT-Neo` from Black et al. (2021), alternate dense attention layers with sparse attention layers.

Modification 3 A final modification employed by later models is that, instead of encoding the position of an example in the embedding layer, positions are encoded in the attention modules. In Equation (2.2), this could be added by computing the numerator with a function that takes into account key and query vectors, as well as relative position differences. One such approach is that of the *Rotary Position Embedding* (RoPE) (Su et al., 2024), which rotates vectors so that the angle is based on the position index.

LayerNorm and the residual connection After the MHSA, the resulting representations are combined with the inputs through a residual connection and pass through LayerNorm

(Ba et al., 2016), see Equation (2.6). LayerNorm for $\mathbf{z} \in \mathbb{R}^d$ is defined in Equation (2.7).

$$\mathbf{u}_i = \text{LayerNorm}(\mathbf{u}'_i + \mathbf{x}_i; \boldsymbol{\gamma}_1, \boldsymbol{\beta}_1) \qquad \texttt{layernorm-pre} \qquad \boldsymbol{\gamma}_1, \boldsymbol{\beta}_1 \in \mathbb{R}^d \tag{2.6}$$

$$\text{LayerNorm}(\mathbf{z}; \boldsymbol{\gamma}, \boldsymbol{\beta}) = \boldsymbol{\gamma} \odot \frac{\mathbf{z} - \mu_z}{\sqrt{\sigma_z^2 + \epsilon}} + \boldsymbol{\beta} \tag{2.7}$$

$$\mu_z = \frac{1}{d} \sum_{j=1}^{d} z_j, \tag{2.8}$$

$$\sigma_z^2 = \frac{1}{d} \sum_{j=1}^{d} (z_j - \mu_z)^2 \tag{2.9}$$

LayerNorm normalises the representations of each token, independent of other examples in a training batch. This stabilises gradients during the backward pass (e.g. Xiong et al., 2020) and enhances the attention mechanism, enabling query vectors to attend to all keys equally, when needed, and ensuring all key vectors are, in principle, selectable by query vectors (Brody et al., 2023).

Modification 4 According to this formulation, LayerNorm occurs *after* the MHSA (and after the feedforward module discussed below). This variant of the transformer layer has, therefore, been referred to as **post-normalised**. One modification implemented later on uses **pre-normalisation** (e.g. Vaswani et al., 2018), which decouples the residual connection and LayerNorm, and applies LayerNorm right before the non-linear modules instead of after. Figure 2.1b depicts the layout of this layer variant. Pre-normalised layers were found to be more stable and faster to train (Xiong et al., 2020).

Multi-head cross-attention Transformers can be used to create architectures of various compositions. Vaswani et al. originally used transformer for NMT, using two chained transformers: the encoder and the decoder. The mechanism that includes encoder representations from source sentence $\mathbf{s}_1^m$ in the decoder is the MHCA module, see Equation (2.10). This module is similar to the MHSA discussed above, with the modification that the query vectors come from the current layer, but the key and value vectors come from layer L of the encoder.

$$\mathbf{v}'_i = [\text{A}^{(1)}(\mathbf{u}_1^n, \mathbf{s}_1^m)_i; \ldots; \text{A}^{(h)}(\mathbf{u}_1^n, \mathbf{s}_1^m)_i] W_O \qquad \texttt{MHCA} \qquad W_O \in \mathbb{R}^{hk \times d} \tag{2.10}$$

$$\mathbf{v}_i = \text{LayerNorm}(\mathbf{v}'_i + \mathbf{u}_i; \boldsymbol{\gamma}_2, \boldsymbol{\beta}_2) \qquad \texttt{layernorm-mid} \qquad \boldsymbol{\gamma}_2, \boldsymbol{\beta}_2 \in \mathbb{R}^d \tag{2.11}$$

Intuitively, this allows the decoder to look at the tokens in the encoder when deciding which token to output (or translate, in NMT) next. Figure 2.2b demonstrates this for our example of "The news came out of the blue". When translating that into Dutch ("Het nieuws kwam uit het niets") using an NMT model used in chapter 6, we observe both a look-back and look-ahead effect in the cross-attention – e.g. when the input token is "_Het" ("*the*"), the cross-attention mostly flows to both "_The" and "_new".

MHCA is again followed by LayerNorm and a residual connection (Equation (2.11)), where the order of the attention, LayerNorm and residual connection would be modified in a pre-normalised transformer. MHCA is not included in a stand-alone transformer.

Feedforward layer Finally, the hidden representations pass through two linear layers with a ReLU non-linearity in between them, see Equation (2.12), followed again by a residual connection and LayerNorm (Equation (2.13)), where the order would be modified in a pre-normalised transformer.

$$\mathbf{z}'_i = \begin{cases} \text{ReLU}(\mathbf{v}_i W_1)W_2 \\ \text{ReLU}(\mathbf{u}_i W_1)W_2 \end{cases} \quad \texttt{feedforward} \quad W_1 \in \mathbb{R}^{d \times p}, W_2 \in \mathbb{R}^{p \times d} \tag{2.12}$$

$$\mathbf{z}_i = \begin{cases} \text{LayerNorm}(\mathbf{z}'_i + \mathbf{v}_i; \boldsymbol{\gamma}_3, \boldsymbol{\beta}_3) \\ \text{LayerNorm}(\mathbf{z}'_i + \mathbf{u}_i; \boldsymbol{\gamma}_3, \boldsymbol{\beta}_3) \end{cases} \quad \texttt{layernorm-post} \quad \boldsymbol{\gamma}_3, \boldsymbol{\beta}_3 \in \mathbb{R}^d \tag{2.13}$$

The first layer projects the hidden representations of dimensionality d into higher-dimensionality vectors (with dimension p), and the second one brings them down to d dimensions, again. As a result, the feedforward layers represent approximately two-thirds of the total number of parameters of a full transformer model.

The functionalities implemented by feedforward layers have not been studied as extensively as those of the attention mechanisms, yet one example of an influential account is that of Geva et al. (2021). They posited that feedforward layers perform key-value memory retrieval, with the columns of W_1 (the keys) as pattern detectors over the input sequence, detecting patterns such as n-grams or semantic topics, that provide weights for the rows of W_2 (the values). Other work has focused on explaining what individual neurons in this module capture, identifying, for instance, neurons that fire if certain facts are in the input (e.g. Dai et al., 2022).

Modification 5 A modification made to the feedforward layer by many LMs proposed post `GPT-2` (Radford et al., 2019) is the use of the GeLU non-linearity (Gaussian Error Linear Unit) instead of the ReLU, to yield more smooth activations.

Modification 6 A final modification worth mentioning is that Wang and Komatsuzaki (2021) modified the self-attention and the feedforward layer to be parallel to one another, instead of being applied consecutively. Figure 2.1c depicts this setup. This modification was primarily to improve the efficiency of training transformer, and has been shown to yield minimal performance degradation (Chowdhery et al., 2023).

Classification or unembedding layer After the representations for $\mathbf{x}_1^n$ have passed through the L layers, they can then be used for classification using the **output** or **unembedding layer**, parametrised by θ_{Unemb}. When using an encoder-decoder model, only the decoder would have an output layer. The output layer contains a weight matrix $W_{V'_y} \in \mathbb{R}^{d \times |V'_y|}$, which projects the hidden representations into the output vocabulary

V_y' and applies softmax to obtain probabilities per output class. For a pre-normalised transformer, this would be preceded by one final LayerNorm. When generating a sequence, as is the case for language modelling and translation, the output classes are tokens, and classification is performed repeatedly. The output vocabulary V_y' would thus be equal to the input vocabulary of the decoder, and one could even use the same weight matrix for the embedding and unembedding layers as Vaswani et al. did for NMT. In that case, the weight matrices are referred to as 'tied'. When using transformer for sequence classification, the output vocabulary is typically much smaller.

Bringing it all together We have now reviewed all of transformer's components, along with modifications made in the years since 2017 that apply to one or multiple models used in this thesis. Transformer Θ thus consists of parameters θ_{Emb}, $\theta_1, \cdots, \theta_L$, and θ_{Unemb}, and when chaining two transformers within one encoder-decoder model as is the case for MT, $\Theta = \{\theta^{enc}_{Emb}, \theta^{enc}_1, \cdots, \theta^{enc}_L, \theta^{dec}_{Emb}, \theta^{dec}_1, \cdots, \theta^{dec}_L, \theta^{dec}_{Unemb}\}$. The corresponding hyperparameters are hidden dimensionality d, the number of heads h, the hidden dimensionality k within the attention modules ($k = \frac{d}{h}$), the hidden dimensionality p within the feedforward layer, and the number of layers L. Vaswani et al. use $d = 512$, $k = 64$, $p = 2048$, $h = 8$ and $L = 6$ for their `transformer-base` architecture. We adopt this original architecture in chapters 3, 5 and 6, and use other transformers in chapter 4.

2.1.2 Language models and fine-tuning

During the last six years, transformers have been widely adopted to train LMs: machine learning models capturing a probability distribution over word sequences. LMs can be used to assign a probability to an entire sequence x_1^n (Equation (2.14)), or predict the most probable continuation x_i^{n*} of a given input x_1^{i-1} (Equation (2.15)), typically by factorising that probability (the likelihood) into a product of conditional probabilities.

$$P(x_1^n|\Theta) = \prod_{i=1}^{n} P(x_i|x_1^{i-1}, \Theta) \tag{2.14}$$

$$x_i^{n*} = \underset{x_i^n}{\arg\max} \prod_{t=i}^{n} P(x_t|x_1^{t-1}, \Theta) \tag{2.15}$$

When using LMs to generate text, computing the most probable continuation is often intractable. Taking the argmax at every timestep is one example of a way to approximate the global argmax (also referred to as **greedy decoding**). To learn model parameters Θ, we use a dataset of m examples $D = \{x^{(i)}\}_1^m$ and learn Θ that maximises the log-likelihood of D via *maximum likelihood estimation* (MLE):

$$\Theta^{LM}_{MLE} = \underset{\Theta}{\arg\max} \sum_{i=1}^{m} \sum_{t=1}^{|x^{(i)}|} \log P(x_t^{(i)}|x_1^{(i),t-1}, \Theta) \tag{2.16}$$

For contemporary LMs, D consists of billions of tokens, and training the model on those tokens is referred to as **pretraining**. However, not all LMs used in the thesis adhere to this objective's causal constraint. BERT (Devlin et al., 2019), elaborated on below, was trained using *masked language modelling* (MLM), where for example i a set of positions $O^{(i)}$ is introduced that contains all token positions from $x^{(i)}$ that were manipulated in $\tilde{x}^{(i)}$. Tokens can be manipulated by, for instance, being replaced with a special token ([MASK]) or by being swapped with a different vocabulary item. We want to learn Θ such that it maximises the probabilities of the manipulated tokens:

$$\Theta^{MLM}_{MLE} = \arg\max_{\Theta} \sum_{i=1}^{m} \sum_{t \in O^{(i)}} \log P(x_t^{(i)} | \tilde{x}^{(i)}, \Theta) \tag{2.17}$$

These two respective types of language modelling correspond to the autoregressive and bidirectional designs of transformer as discussed in the previous subsection. In the autoregressive setup, when making predictions based on input $x_1^{(i),t-1}$, the mask in the MHSA in Equation (2.2) prohibits token interactions beyond time step $t-1$. In the bidirectional design, the entire (manipulated) input sequence $\tilde{x}^{(i)}$ is given to the model, and none of the attention flow is blocked. Figure 2.3 depicts the architectural differences of the LMs and translation models used throughout the thesis, and demonstrates the order in which the different LMs were presented, providing context for why they have certain architectural differences.

Finding the MLE solution is an optimisation problem, which is equivalent to finding the solution that minimises the *negative log-likelihood* (NLL) loss:

$$\mathcal{L}^{LM}(\{x^{(i)}\}_1^m, \Theta) = -\sum_{i=1}^{m} \sum_{t=1}^{|x^{(i)}|} \log P(x_t^{(i)} | x_1^{(i),t-1}, \Theta) \tag{2.18}$$

This optimisation problem is non-convex and is approached by applying mini-batch stochastic gradient descent, an iterative process that starts from a model initialisation and iteratively updates the parameters using gradients computed for a batch of the data until reaching a stopping criterion. The models used throughout the thesis all rely on the Adam and AdamW optimisers (Kingma and Ba, 2015; Loshchilov and Hutter, 2017). We do not further experiment with the optimiser, and, therefore, refer the reader to the respective papers for more information.

BERT Devlin et al. (2019) proposed *Bidirectional Encoder Representations from Transformers* (BERT), one of the first widely used transformer-based LMs. BERT consists of one bidirectional transformer with layer type (a) from Figure 2.1, and was released for two model sizes: a base model (with 12 layers and 110M parameters) and a large model (with 24 layers and 340M parameters). BERT was trained using two objectives: an MLM objective for predicting which token in the input was randomly manipulated, and a next sentence prediction objective for predicting whether two text segments preceded each other in the training corpus. The models have a vocabulary constructed using the WordPiece algorithm (Wu et al., 2016), a BPE-like algorithm that builds a vocabulary

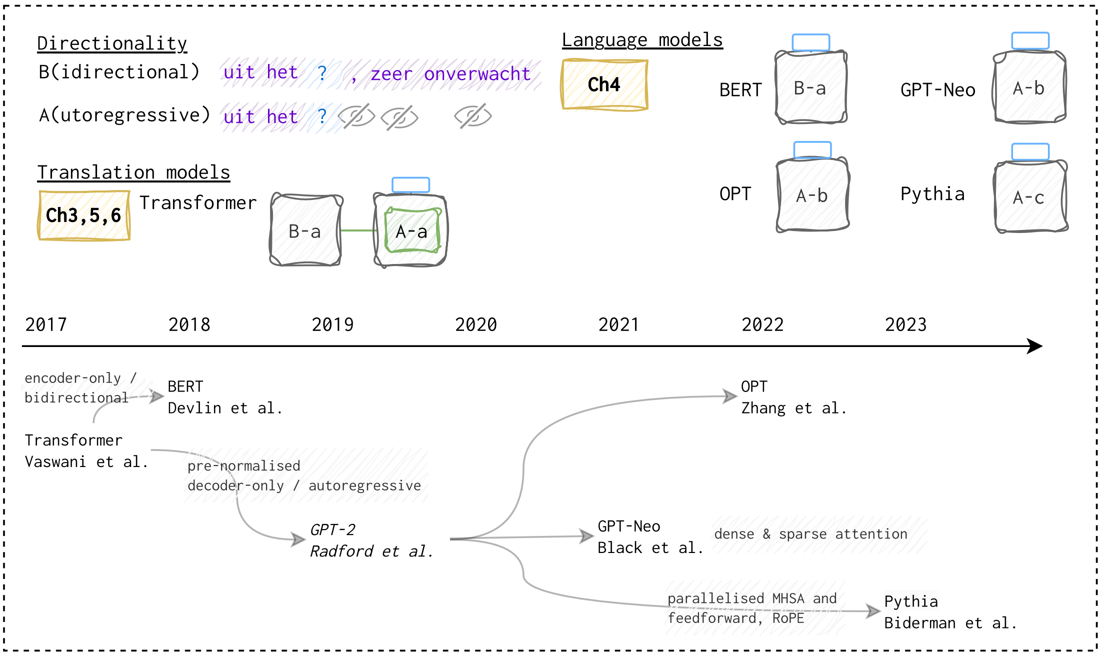


Figure 2.3: Overview of the models used in the thesis, with a generic indication of their architectural differences and the way models relate to one another, temporally. GPT-2 is the only model we do not consider, but it was influential for architectural changes to LMs proposed after BERT. a, b and c refer to the subfigures from Figure 2.1.

of subword units by iteratively merging pairs of tokens to maximise the likelihood of a simple LM. When encoding a sequence, every input starts with a special symbol `[CLS]`, and segments are delineated by using the `[SEP]` token in between the two parts and by adding special segment embeddings to transformer's standard input embeddings. BERT was pretrained using 3.3B tokens from the BooksCorpus (Zhu et al., 2015) and English Wikipedia.

To use LMs for tasks other than language modelling, Devlin et al. proposed a straightforward **fine-tuning** recipe that was widely adopted afterwards: the classification layer of the model used during pretraining is replaced with a randomly initialised classification layer whose number of output classes matches the task of interest, and the model and this layer are trained jointly on a new task. To demonstrate the superiority of BERT compared to models that were considered the state of the art in 2019, Devlin et al. evaluated the model on benchmarks GLUE (Wang et al., 2019b), SQuAD (Rajpurkar et al., 2016) and SWAG (Zellers et al., 2018), that evaluate natural language understanding, question answering and common-sense inference using a wide range of tasks. We do not further elaborate on these tasks here, but return to benchmark tasks for fine-tuning in chapter 4, where we, too, use them in our experiments. When fine-tuning, models are trained to minimise the NLL loss using data different from the pretraining data, in a dataset with input and output pairs: $D = \{(x^{(i)}, y^{(i)})\}_1^m$. Inputs are sentences in natural language, and the output classes come from a typically small vocabulary V_y:

$$\mathcal{L}^{FT}(\{(x^{(i)},y^{(i)})\}_1^m,\Theta) = -\sum_{i=1}^{m}\log P(y^{(i)}|x^{(i)},\Theta) \tag{2.19}$$

In the years that have passed since BERT appeared, other types of fine-tuning have been proposed, such as parameter-efficient fine-tuning (e.g. Houlsby et al., 2019) that fine-tunes a small set of model parameters or newly introduced parameters together with the classifier while freezing the remainder of the model. Alternative approaches have established that, particularly for models that far exceed BERT in size, downstream tasks can be performed by rephrasing them as a language modelling task using a technique referred to as prompting, or in-context learning (Brown et al., 2020). The task is then introduced in input x_1^{t-1}, and the model's continuation x_t^n is post-processed to extract the prediction. In this thesis, we will only consider fine-tuning as proposed by Devlin et al., which is why we do not further elaborate on these alternative approaches here.

GPT-2 and beyond Prior to and following the release of BERT, other authors and teams proposed a range of transformer-based models. Very influential was the release of GPT-2 by Radford et al. (2019), a scaled-up version of its predecessor GPT-1 (Radford et al., 2018). GPT-2 is architecturally different from BERT in that it uses pre-normalisation and autoregressive language modelling, and a range of models proposed afterwards adopted similar architectures. The release of GPT-2 was followed by the announcement of GPT-3, but its parameters were never released to the public. What followed was that various initiatives developed models inspired by the GPT series, either developed to create (open) models that could compete with them in terms of performance, or to better understand LMs and their scaling behaviour. Models from three such initiatives will be employed in this thesis:

- GPT-Neo: GPT-Neo (Black et al., 2021) was one of the first open-science projects that aimed to train autoregressive LMs that could compete with GPT-3. The models vary in size from 125M to 2.7B (12-32 layers), and were pretrained on approximately 300B tokens from the Pile (Gao et al., 2020) – an open-source dataset composed of 22 smaller datasets, including academic texts, web content, books, code, and more. GPT-Neo uses layer type (b) from Figure 2.1, applying pre-normalisation, but with the attention changed to integrate dense and sparse attention.

- OPT: Zhang et al. (2022b) presented *Open Pretrained Transformer* (OPT), a family of decoder-only transformer LMs that vary in size from 125M to 175B parameters, and vary in layers from 12 to 96 layers. OPT uses layer type (b) from Figure 2.1 and was pretrained on a 180B corpus that combines data from multiple corpora, among which the BooksCorpus (Zhu et al., 2015) and the Pile (Gao et al., 2020).

- `Pythia`: Biderman et al. (2023) presented `Pythia`, a family of decoder-only transformer LMs that vary in size from 70M to 6.9B parameters, and vary in layers from 6 to 36 layers. `Pythia` uses layer type (c) from Figure 2.1, thus having a setup that employs RoPE, and in which the MHSA and feedforward modules are parallelised. `Pythia` was trained on 300B tokens from the Pile (Gao et al., 2020).

All of these model families use a tokenisation scheme referred to as byte-level BPE (Wang et al., 2020). Byte-level BPE merges at the level of bytes instead of characters. Upon presentation, the models were primarily evaluated using prompting techniques, to demonstrate performance that closely resembles `GPT-3` for the larger model sizes, yet, similar to `BERT`, they can also be used in the fine-tuning paradigm, as we will further elaborate on in chapter 4.

2.1.3 Neural machine translation

Ever since the presentation of `BERT`, when evaluating or analysing models on downstream tasks, one would typically start by fine-tuning an LM rather than starting training from a randomly initialised transformer. NMT has, for a long time, been an exception to this rule, and in the thesis (in chapters 3, 5 and 6) we will review NMT systems that thus also train from scratch directly on data relevant to the task. In MT, given our sequence in the source language x_1^n (from vocabulary V_x), we aim to find the translation in the target language, y_1^{k*} (from vocabulary V_y):

$$y_1^{k*} = \arg\max_{y_1^k} \prod_{t=1}^{k} P(y_t | x_1^n, y_1^{t-1}, \Theta) \tag{2.20}$$

Akin to language modelling, we factorise the probability of the target into a product of conditional probabilities over tokens, such that the probability of the next translated token depends on the source and all translated tokens observed so far. Note that, again, finding the global argmax is intractable in practice, so when generating a translation with an NMT model, one approximates the global argmax using greedy decoding or otherwise, as we elaborate on below. We learn model parameters Θ using a dataset of m example source-target pairs, $D = \{(x^{(i)}, y^{(i)})\}_1^m$, via MLE, and the NLL loss, applying length normalisation for the targets:

$$\Theta_{MLE}^{MT} = \arg\max_{\Theta} \sum_{i=1}^{m} \frac{1}{|y^{(i)}|} \sum_{t=1}^{|y^{(i)}|} \log P(y_t^{(i)} | x^{(i)}, y_1^{(i),t-1}, \Theta) \tag{2.21}$$

$$\mathcal{L}^{MT}(\{(x^{(i)}, y^{(i)})\}_1^m, \Theta) = -\sum_{i=1}^{m} \frac{1}{|y^{(i)}|} \sum_{t=1}^{|y^{(i)}|} \log P(y_t^{(i)} | x^{(i)}, y_1^{(i),t-1}, \Theta) \tag{2.22}$$

In practice, these sequences typically consist of BPE tokens (Sennrich et al., 2016), where the BPE tokens are estimated using D, using either one vocabulary for each language, or one large joint vocabulary.

The architecture Vaswani et al. (2017) implement the model Θ using two chained transformers: the encoder and the decoder. Both transformers have layer type (a) from Figure 2.1. The encoder is a bidirectional transformer without the MHCA module; it processes the source sequence x_1^n and its outputs are the hidden representations from transformer layer L. The decoder is an autoregressive transformer with the MHCA module; it encodes all target tokens seen so far by taking into account the encoded source sequence via the MHCA, as previously depicted in Figure 2.2b. Only the decoder has an unembedding layer, which transforms the output of decoder layer L into a vector of size $|V_y|$, that, following the application of the softmax function, is a probability distribution over the tokens from V_y.

Training and inference During the training phase, we optimise the model parameters using the NLL loss function applied within batches that fit up to a pre-specified number of tokens. Following Vaswani et al., it became standard to apply label smoothing during training, assigning a small portion of the target probability mass to incorrect classes to keep models from being overconfident.

When using the model after training to predict translations for new inputs, we compute an approximation of y_1^{k*}, but not necessarily the translation that is the most likely, because the number of possible translations grows exponentially as the translation gets longer. Instead, translations are generated using **beam search**, where one sets a beam size and stores only that number of hypotheses at a time. With each time step, one considers extending all current solutions in the beam with every vocabulary token, storing only the candidates allowed within the beam size. Generation ends for an individual translation when the EOS token is generated. The selected translation is the one with the highest overall (log-)probability. Setting the beam size to one yields **greedy decoding**, where the highest-probability token is selected at each time step.

Evaluation In NLP, it is standard practice to use train and test sets that are randomly sampled from a larger data pool; yet, for NMT, this is not the case. NMT training corpora are often semi-automatically collected, e.g. by scraping the web for examples that are likely to be translations (e.g. Schwenk et al., 2021b). When randomly sampling test data, one may thus end up with low-quality translations. To avoid overestimating models' performance, gold-standard human-translated text is used instead.[3]

The quality of models' translations has since long been measured using the *Bilingual Evaluation Understudy* (BLEU) (Papineni et al., 2002) metric. It calculates n-gram overlap between translations and targets (typically for $n = 1$ to 4), computing the geometric mean of the n-gram overlap multiplied by a brevity penalty. BLEU has been

[3]The Conference on Machine Translation (e.g. Kocmi et al., 2022, 2023), for instance, releases new test sets with each shared task, to ensure that the test data is unseen and of high quality.

criticised for not capturing the semantic adequacy of translations and being insensitive to fluency and grammaticality. Alternative metrics have been proposed that capture token- or character-level overlap of targets and translations (e.g. METEOR, or chrF, Banerjee and Lavie, 2005; Popović, 2015), or that use neural models trained specifically to estimate translation quality (e.g. COMET, Rei et al., 2020). In spite of this, BLEU remains a widely adopted metric for evaluation in NMT; to this day, it facilitates standardisation in the evaluation across systems and papers.

In chapter 3, we will not only consider models' quality in terms of BLEU and COMET but also look at their tendency to *hallucinate*. Hallucinations are translations with certain n-grams repeated over and over, or translations with fluent (sub-)sentences that do not align to the source material (Guerreiro et al., 2023). Lee et al. (2018) were one of the first to study this extensively for NMT, by inserting tokens into source sentences and measuring for how many sentences this could lead to a hallucination. Raunak et al. (2021) later posited that memorised examples can more easily lead to a hallucination than non-memorised ones. We adopt the approach of Lee et al. in chapter 3.

2.1.4 Interpretability methods

The rise of deep neural networks sparked interest in what their layers, parameters, and internal representations capture, and how model-internal signals can explain model outputs. Methods that were developed to answer these questions are broadly referred to as **interpretability methods** (see Madsen et al., 2022, for an overview). More recently, **mechanistic interpretability methods** (Saphra and Wiegreffe, 2024) gained traction, which put more emphasis on explaining model behaviour through causal interventions instead of sticking to post-hoc analyses. Many such methods are not specific to transformer, but have been widely applied to transformer in recent years. In chapters 4 and 6, the methodology used relies on some of these methods, as summarised in Figure 2.4, which we will now briefly review. Our review primarily covers techniques we, ourselves, employ and is not meant to be an exhaustive overview.

Behavioural probes When processing an input, **hidden representations** are computed that are passed from layer to layer as reviewed in §2.1. A wide range of efforts has investigated what those representations capture, and what they can teach us about what the model has learnt when it comes to, for instance, linguistic structure. The predominant paradigm for doing so has employed **probing classifiers** (Alain and Bengio, 2017; Conneau et al., 2018; Hupkes et al., 2018). Probing classifiers are typically external, small neural networks that take the hidden representations as inputs, and are trained on a task of choice. If the probe can make accurate predictions, it is assumed that the hidden representations that were fed as input encoded the property that the

	internal modifications	interpreting model-internals post-hoc	external interpretations
hidden representations	amnesic probes, Ch6	CCA analyses, Ch4, Ch6	behavioural probes, Ch4, Ch6
attention		attention weight analysis, Ch6	
model parameters	layer replacement/ re-initialisation, Ch4	gradient analysis, Ch4	

Figure 2.4: An overview of the interpretability methods applied in the thesis, the entity to which they apply and the extent to which they operate inside or outside of the model.

task of choice measured. The primary criticism of this paradigm is that it is not guaranteed that the model encoded the property, since a probe of sufficient size might, given sufficient training data, learn the task even if the original model did not encode the property probed for (e.g. Hewitt and Liang, 2019; Ravichander et al., 2021).

Considerations for use When applying probes, this should thus be handled with care, for instance by severely limiting the capacity of the probe, using control tasks and only relying on the probe's increase in performance compared to the control task (Hewitt and Liang, 2019), or adopting an information-theoretic approach to probing (Voita and Titov, 2020). We refer to probes that are trained to identify properties or explain model behaviour using those properties as **behavioural probes**, and adopt such probes in chapters 4 and 6.

Amnesic probes A second criticism of probing is that even if the hidden representations appear to encode a property, that property may not causally influence the model's predictions. **Amnesic probing** (Elazar et al., 2021) extended conventional probing in a way that establishes that causal connection, by removing information from the hidden representations based on probes, and monitoring the change in behaviour of the base model. *Iterative null-space projection* (INLP), proposed by Ravfogel et al. (2020), was used to remove information from the representations, by training k probing classifiers to predict a property from hidden representations H. After training probe i, parametrised by W_i, the vectors are projected onto the null-space of W_i, using projection matrix $P_{N(W_i)}$, such that $W_i P_{N(W_i)} H = 0$. The projection matrix of the intersection of all k null spaces can then remove features found by the k classifiers. We complement analyses of behavioural probes with amnesic probes in chapter 6.

Considerations for use When intervening in a model, one should make minimal changes to avoid degrading its overall performance. This can be checked by monitoring evaluation metrics other than the reduction of the model's ability to perform the task that the probe measures. For instance, one could measure the change in perplexity when applying INLP to an LM. It should also be taken into account that INLP can only remove linearly decodable information – i.e. the representation may still contain information about the target attribute, just not in a linear form.

Intrinsic probes Instead of training probes externally, alternatives for behavioural probes analyse model internals directly. Such **intrinsic probes** lay out how linguistic information is structured within representations. Examples are techniques that select dimensions that most encode morphosyntactic attributes such as tense, gender, animacy or case (Hennigen et al., 2020), or similarity-based analyses that align representations from different models or layers and then examine how representations differ with respect to a target attribute (e.g. Raghu et al., 2017; Morcos et al., 2018; Voita et al., 2019a; Saphra and Lopez, 2019). Similarity-based analyses often relied on variants of *Canonical Correlation Analysis* (CCA) (Hotelling, 1936), and we adopt this method in chapters 4 and 6. Assume matrices $A \in \mathbb{R}^{d_A \times n}$ and $B \in \mathbb{R}^{d_B \times n}$, that are representations for n datapoints, drawn from two different sources with dimensionalities d_A and d_B – e.g. different layers of one network. CCA linearly transforms these subspaces $A' = WA$, $B' = VB$ such as to maximise the correlations $\{\rho_1, \ldots, \rho_{\min(d_A, d_B)}\}$ of the transformed subspaces.

Considerations for use Because CCA does not train an external network, the criticisms that apply to probing do not apply as much here. Still, one should ideally monitor that differences present for the different categories are not also observable within one category, and the dataset size should exceed the number of dimensions in the representation (Kornblith et al., 2019).

Although probing is still applied to this day, other analysis methods for interpreting hidden representations have gained popularity in recent years. One such method is the use of sparse auto-encoders (Huben et al., 2024, i.a.), that reconstruct hidden representations in a sparse way to ease the interpretability of the resulting features. Another method is the projection of hidden representations from lower layers into the vocabulary space using the unembedding layer (nostalgebraist, 2020), to interpret what representations capture in terms of the output vocabulary. We do not further elaborate on them here, but refer the reader to Ferrando et al. (2024) for a comprehensive overview.

Attention weight analyses When a transformer processes an input, it applies attention modules, and the per-token weights those modules produce have been widely studied in the interpretability literature (Figure 2.2). The weights have been used for **input attribution** since they are assumed to capture the relevance of tokens in the model's predictions; the higher the attention weight on token x_i when predicting target y, the more important that token is assumed to be. The extent to which attention is a faithful explanation of the reasoning process of the model has been widely debated. Most notably, Jain and Wallace (2019) pointed out that, for tasks such as sentiment analysis and natural language inference, attention weights can be modified without changing models' predictions and that attention weight analysis does not always agree with other methods for input attribution. In response, Wiegreffe and Pinter (2019) argued that while attention is not always faithful, it is often a plausible explanation.

Analyses of attention weights are not unique to transformer models, but the widespread usage of the model has been a catalyst for this type of research, since transformer relies on attention in all of its layers. To get to the bottom of interpreting how transformer performs certain tasks, we thus simply *must* analyse what these attention mechanisms capture, while taking care when making causal claims about attention's role. We previously included examples of attention weight analyses in §2.1, and include one such analysis in this thesis when relating attention mechanisms to idiom processing in chapter 6.

Considerations for use To make attention patterns more reliable, alternative ways of analysing them have been proposed. For instance, taking into account attention weights from transformer's layers $l < 1$ when analysing layer l (Abnar and Zuidema, 2020), jointly analysing the weights with the norms of transformer's value vectors to which the weights apply (Kobayashi et al., 2020), or applying interventions within the attention mechanism to study causal connections to the model's output, e.g. through attention knockout that disables certain attention mechanisms (Geva et al., 2023).

Gradient-based analyses of model parameters Instead of relying on the by-products of a forward pass (i.e. hidden representations and attention weights) one can also directly try to interpret what the underlying model parameters capture. Obvious signals to inspect are the gradients with respect to the loss, on a selected data subset, where a larger gradient norm would suggest a higher relevance. While gradient-based methods have been widely used as alternatives to attention weights for input attribution (Bastings and Filippova, 2020), they have also been used to study what certain layers within models capture, as we will do in chapter 4. Maini et al. (2023) and Stoehr et al. (2024), for instance, used gradient-based methods to study layer relevance for memorised examples, on which we further elaborate in §2.2.3.

Considerations for use Care should be taken when interpreting the norm as layer relevance directly, since this post-hoc signal does not always align with other methods that intervene within the model (Maini et al., 2023). For instance, one could inspect the maximum within a layer or model component (Stoehr et al., 2024), or inspect the norm's ratio for two different data subsets of interest (Stephenson et al., 2021).

Modifying model parameters Instead of analysing model parameters post-hoc, one can change a subset of the parameters directly and monitor the effect on model behaviour

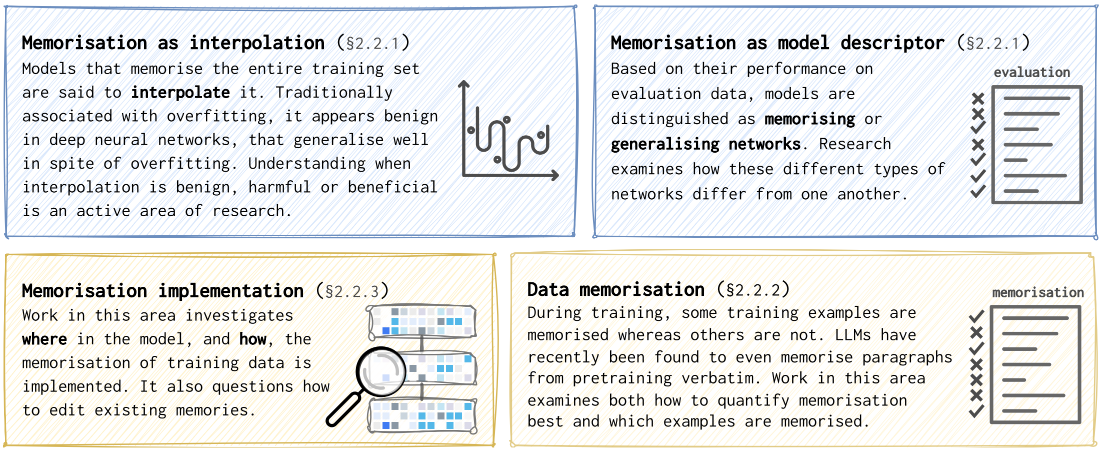


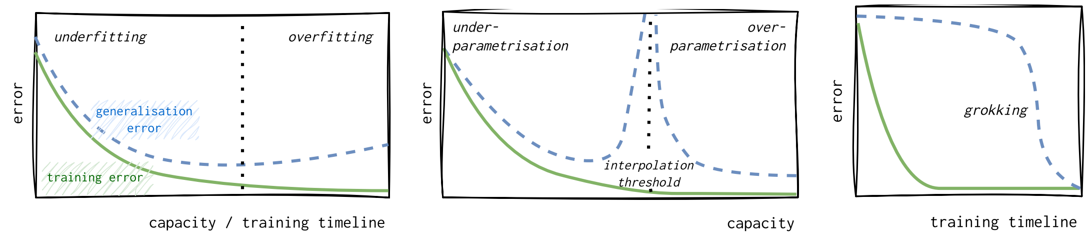
Figure 2.5: Summary of the different ways in which memorisation has been discussed in the literature. In the thesis, we contribute new work to the directions marked in yellow.

with respect to the phenomenon of interest. If the model's behaviour remains unchanged, the modified subset of parameters likely did not encode that phenomenon. Stephenson et al. (2021), Zhang et al. (2022a), Maini et al. (2023), for instance, applied layer re-initialisation, re-randomisation and retraining to study layers' roles in image classification for transformer and deep *convolutional neural networks* (CNNs), and Mosbach et al. (2020, 2021) applied layer re-initialisation to examine how fine-tuning affects BERT's layers. We apply similar methods in chapter 4.

Considerations for use When applying these methods, it is important to recognise that internal modifications to the model may affect not only the targeted capability but also its overall behaviour. The more the overall behaviour changes, the less certain one can be that the specific change focused on is meaningful. As mentioned for the amnesic probes, it is thus a good practice to monitor evaluation metrics other than performance on the examples or task of interest.

2.2 Memorisation metrics and findings

Now that we have reviewed the models, tasks, and interpretability methods we will consider, we can dive into the topic of memorisation. Here, we first review a brief history of how memorisation became a multi-faceted topic within deep learning in recent years (§2.2.1), after which we will move on to discussing memorisation of individual datapoints (§2.2.2), and how memorisation affects models internally (§2.2.3). Figure 2.5 summarises the different ways in which memorisation has been discussed in the literature, highlighting the directions to which the thesis directly contributes.



(a) Overfitting (b) Double-descent (c) Grokking

Figure 2.6: Illustrations of phenomena related to memorisation observed in classical machine learning (a), and deep learning (b, c).

2.2.1 A paradigm shift around memorisation

The broader topic of memorisation traces back to the core principles of machine learning of optimisation, evaluation and generalisability.[4] By applying optimisation techniques, we learn models that minimise the training error, but the quality of models is ultimately determined by their generalisation or test error – the expected error on new examples. A common assumption is that training and test data share the same underlying distribution and are *independent and identically distributed* (IID). A model that perfectly captures the underlying data distribution would then have equal expected training and test errors. However, in practice, learning (imperfect) models based on training data yields a test error that exceeds the training error. Models' success depends on both minimising the training error and narrowing the gap between the training and test error. A model that achieves low training error but fails to generalise exhibits **overfitting**, memorising properties of the training set that do not transfer to new data. Conversely, **underfitting** occurs when a model neither captures training data well nor generalises. These challenges have been central to machine learning for decades; for example, Dietterich (1995, p.326) already highlighted concerns about "fit[ting] the noise in the data by memorizing various peculiarities". Figure 2.6a illustrates these phenomena. Under- and overfitting can be discussed within the context of training a single model for many epochs until it starts to overfit, but also within the context of different models of increasing sizes, that, as they grow in capacity, become more likely to overfit (according to classical machine learning theory, at least).

To address overfitting and balance underfitting and overfitting, key strategies include adjusting model capacity (avoiding over-parametrisation), applying regularisation techniques (explicit, like weight decay, or implicit, like early stopping), and increasing the training dataset size. Yet, deep neural networks have defied the assumption that over-parametrisation (and thus memorisation) cannot co-occur with strong generalisation skills. Zhang et al. (2017), for instance, trained pretrained deep CNNs on image

[4] I refer to Goodfellow et al. (2016) for elaboration on the core principles of deep learning, in general, and the notions of overfitting and underfitting in their section 5.2, specifically.

classification tasks, demonstrating that without altering the training procedure or the model, zero training error can be achieved on both (i) regular data and (ii) mislabelled variants of that data. This established that even though the network has the effective capacity[5] to memorise the training set when trained on (i), generalisation skills emerge. They interpret memorisation as **interpolation** of the entire training set – achieving 100% training accuracy and near-zero training error after training for multiple epochs. The coexistence of over-parametrisation with a low generalisation error is known as *benign overfitting* (Bartlett et al., 2020). Subsequent research has explored conditions enabling benign overfitting (Li et al., 2021b), qualitative differences in models trained on real vs noisy data (Arpit et al., 2017), and concerns that benign overfitting can actually be 'malign', harming generalisation when evaluating models outside of the IID evaluation paradigm (Sanyal et al., 2020; Wald et al., 2023). Benign overfitting is closely linked to the *double descent* phenomenon (Belkin et al., 2019) (Figure 2.6b), where the test error initially decreases when increasing model capacity, then rises (as predicted by traditional machine learning theory), but ultimately drops again once the model becomes over-parametrised for the given data, exhibiting benign overfitting.

While interpolation depends on both the model and its training set, memorisation has also been discussed as a qualitative **model descriptor**, based on train and test performance. A model that interpolates the training set can be either a *memorising network* or a *generalising network* in case of a high or low generalisation error, respectively. In the aforementioned work, but also in articles discussed later on in §2.2.3, memorising networks are often intentionally trained using label randomisation or input corruption to study the properties that these networks have (e.g. Zhang et al., 2017; Arpit et al., 2017; Morcos et al., 2018). However, memorising and generalising networks can also emerge naturally during training through grokking (Power et al., 2022): initially, the model interpolates the training set but generalises poorly, until – after training for many additional epochs – the test accuracy suddenly increases long after the training data has been interpolated (see Figure 2.6c). In this process, a memorising network transitions into a generalising one. While the topics of grokking and double descent have mostly been studied in isolation, Davies et al. (2023) suggest they can be viewed as two sides of the same coin.

Research on overfitting, grokking, and double descent typically examines trends across model groups rather than the properties of specific memorised datapoints (§2.2.2) or how memorisation affects a model internally (§2.2.3). We do not explore these topics further here, but the above work provides context for the discussions ahead.

[5] The set of hypotheses, i.e. model instantiations, that are reachable by applying a specific learning algorithm to a specific dataset (Arpit et al., 2017).

2.2.2 Data memorisation

In the previous subsection, we discussed memorisation as a phenomenon that either applies to the training set as a whole or as a qualitative descriptor of models that have memorised their training set without generalising to test data. Yet, memorisation is a multi-faceted phenomenon, and within NLP, the type of memorisation that has received the most attention in recent years is that of memorisation of individual datapoints from the pretraining or the subsequent fine-tuning phase. Given the vast sizes of pretraining corpora, it is impossible for contemporary LLMs to fully interpolate their pretraining data; especially since pretraining now often only involves a single pass over the data, successfully avoiding overfitting in the traditional sense (Xue et al., 2023). And yet, even within single-pass pretraining, individual examples from pretraining *are* still memorised. Related work, therefore, focuses on identifying how much of the pretraining or fine-tuning data models memorise, what information those memorised examples contain, and establishing under which conditions memorisation increases. In doing so, articles typically distinguish between binary and graded memorisation metrics.

Binary memorisation metrics When approaching memorisation as something that applies to only a small subset of datapoints, a widely used definition is that of **extractable memorisation** (Nasr et al., 2023), which applies to generative models:

> *Given a generative model f parametrised by θ, an example x from a training set $\mathbb{X}$ is extractably memorised if one can construct a prompt p that makes the model produce x, i.e. $f_\theta(p) = x$.*

More restricted variants of this definition are *discoverable memorisation* (Nasr et al., 2023), where p comes from the training set and directly precedes x, *k-eidetic memorisation* (Carlini et al., 2019), which focuses on cases where x appears $\leq k$ times in $\mathbb{X}$, and *exact memorisation* (Tirumala et al., 2022), which focuses on the final token in x. More loosely defined variants relax the constraint that x needs to be fully reproduced, e.g. in the case of *approximate memorisation* (Ippolito et al., 2023). The general type of memorised content that these metrics all capture – the literal reproduction of text from the training set – is called **verbatim memorisation**.

Carlini et al. (2021) were among the first to identify that LMs have memorised certain problematic content verbatim: out of 200,000 examples generated with `GPT-2`, there were 604 exact matches, containing text from news articles, copyright notices and Wiki entries, but also universally unique identifiers and PII such as names, phone numbers, and email addresses. McCoy et al. (2023) later found that `GPT-2` even memorises passages that are over 1000 words long, although such cases are quite rare. Related work has primarily focused on extracting memorised content at a larger scale (e.g. Carlini et al., 2022; Nasr et al., 2023) and improving the techniques to extract the sequences, such as attaching a constant soft prompt to the prefix (e.g. Wang et al., 2024).

Binary metrics outside of verbatim memorisation have used **membership inference** techniques that rely on the loss of individual examples and a set threshold to decide whether it was likely that a specific example was included in the training set (e.g. Kharitonov et al., 2021; Mireshghallah et al., 2022). And, finally, a third category of work uses **phenomenon-specific definitions** of binary memorisation: Chang et al. (2023) relied on the close accuracy of named entities to identify that ChatGPT has memorised content from copyright-protected books, Haviv et al. (2023) measured memorisation based on verbatim reproduction of specific words within idioms, and Raunak and Menezes (2022) define extractive memorisation in NMT to refer to examples for which models have memorised to produce the full translation after seeing only a prefix of the source sentence.

Together, studies adopting various definitions of memorisation have identified data properties, model features, and experimental setups that increase the chances of examples being memorised, in models trained from scratch, pretrained models, and fine-tuned models. Dominant contributing factors are large model sizes, data duplication, and using longer prompts during extraction (Raunak and Menezes, 2022; Tirumala et al., 2022; Carlini et al., 2022; Biderman et al., 2023), using larger input vocabulary sizes (Kharitonov et al., 2021), and fine-tuning the head of a model (as opposed to full fine-tuning or adapter tuning) (Mireshghallah et al., 2022).

Graded memorisation metrics and the long-tail theory Instead of considering memorisation as something that applies to only some examples in our training dataset, we can view it as something that happens to all examples, to some extent, thus adopting a graded metric. The most influential contribution to the discussion of graded memorisation metrics in recent years has been the account by Feldman (2020), whose theory was later referred to as the **long-tail theory** (LTT). To discuss this, we briefly step away from NLP again to think of deep learning, in general, and review related work from CV before returning to applications within NLP. In chapter 3, we will put the LTT to the test within the context of NMT.

Feldman explains why memorising training examples can co-occur with generalisation by emphasising that the data distributions that underlie deep learning tasks are often long-tailed. There can be many very infrequent output classes, and even within one class, subpopulations vary in frequency. For instance, in image classification, subpopulations of one class could differ by the object's visibility in the image. When observing a representative example of an atypical subpopulation in the training data, memorising that example could positively influence the test set accuracy for that subpopulation. While some atypical examples are true outliers instead of representing a rare subpopulation, these two types can be indistinguishable to the model. In data distributions where the number of outliers is not excessive, and the total weight of all atypical subpopulations is significant enough, memorisation of atypical examples can be a near-optimal strategy for generalisation.

In the LTT, there is thus a notion of *influence* (Feldman and Zhang, 2020) that training examples have on test examples. Assuming a learning algorithm $\mathcal{A}$ that is used to learn model f using dataset D, a training example (x_i, y_i), a test example (x_j, y_j) and a performance metric M comparing $f(x_i)$ to y_i, influence is defined as follows:

$$\texttt{infl}(\mathcal{A}, D, i, j) = \underset{f \leftarrow \mathcal{A}(D)}{M}(x_j, y_j, f) - \underset{f \leftarrow \mathcal{A}(D^{\setminus i})}{M}(x_j, y_j, f) \tag{2.23}$$

Memorisation is then simply the self-influence of an example i, referred to as *label memorisation* (Feldman, 2020; Feldman and Zhang, 2020) or **counterfactual memorisation** (CM) (Zhang et al., 2023):

$$\texttt{CM}(\mathcal{A}, D, i) = \underbrace{\underset{f \leftarrow \mathcal{A}(D)}{M}(x_i, y_i, f)}_{\text{'IN' performance}} - \underbrace{\underset{f \leftarrow \mathcal{A}(D^{\setminus i})}{M}(x_i, y_i, f)}_{\text{'OUT' performance}} \tag{2.24}$$

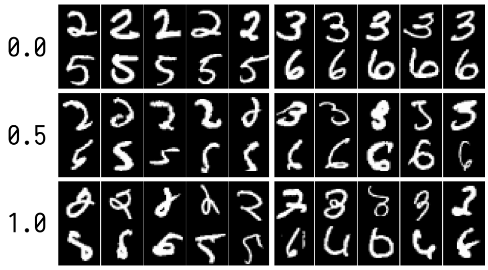
This memorisation metric thus contrasts the performance on example i when models have been trained on i ('IN' performance) to the performance the models would have had, had i not been a training example (the 'OUT' performance, hence the 'counterfactual' nomenclature). Note that this metric does not characterise which datapoints an individual model instantiation memorises, but expresses how likely a datapoint is to be memorised by a given architecture and training procedure.

Due to the leave-one-out notion that the definitions rely on, and the fact that deep learning datasets often contain thousands or millions of examples, Feldman and Zhang propose a method for approximating CM, training models on randomly sampled subsets of D. To then compute CM for example i, we collect every model l for which $(x_i, y_i) \in D_l$ in $\Theta^{IN,i}$ (containing m models total), and gather the remaining k models in $\Theta^{OUT,i}$:

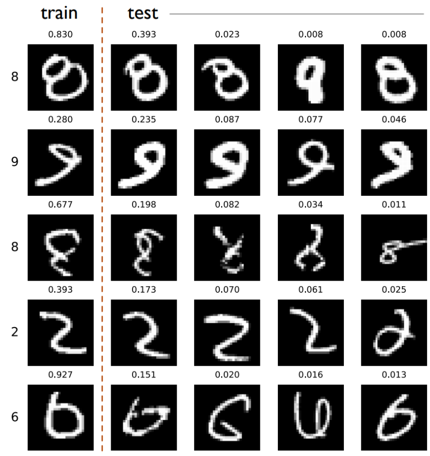
$$\texttt{CM}(i, \Theta^{IN,i}, \Theta^{OUT,i}) = \frac{1}{m} \sum_{j=1}^{m} M(x_i, y_i, \Theta_j^{IN,i}) - \frac{1}{k} \sum_{j=1}^{k} M(x_i, y_i, \Theta_j^{OUT,i}) \tag{2.25}$$

Feldman and Zhang measured the CM and influence metrics for image classification, using ResNet-50 (He et al., 2016) trained on Imagenet (Deng et al., 2009) and CIFAR-100 (Krizhevsky, 2009), and a simpler CNN trained on MNIST (LeCun et al., 1998). They identified that removing examples with high CM harms the generalisation accuracy more than removing random examples. Figure 2.7a includes examples from MNIST with three different CM scores: low CM examples are 'prototypical digits', whereas the others are atypical, ambiguous or mislabelled. Figure 2.7b illustrates the positive influence atypical, memorised digits can have on test examples that resemble them.

Zhang et al. (2023) were the first to explore CM in NLP, computing CM scores for 2M examples by training transformer-based LMs. They linked CM to text simplicity –



(a) CM scores for digits 2, 3, 5, 6



(b) CM ('train') and influence scores ('test')

Figure 2.7: Examples of CM and influence scores for MNIST data, as computed by Feldman and Zhang (2020).

finding that high-CM examples have intermediate simplicity, while low-CM examples are either very easy or very hard – examined CM per domain and determined the influence of the number of models and training epochs used to approximate CM. They identified that data duplication is associated with lower CM, confirming that CM measures a different type of memorisation than the binary metrics from §2.2.2. They also measured examples' influence on unseen data, establishing that high CM examples have a larger maximum influence than low CM examples. Although Zhang et al. highlight some features that might lead to high CM when discussing the different domains – e.g. the presence of non-English tokens and structured data like tabular texts – they did not systematically analyse CM in relation to datapoints' features, as we do in chapter 3.

Inspired by CM, Zheng and Jiang (2022) proposed a *self-influence* metric to quantify the change in parameters when down-weighting a training example, to measure memorisation in sentiment analysis, natural language inference and question answering. They confirmed that removing examples with high self-influence from the training set has a larger negative effect on generalisation performance than removing random examples.

Raunak et al. (2021) computed approximated CM scores for NMT systems, by training ten models on subsets of a 160k English-German translation dataset, with the primary aim of showing the relation between CM and hallucinations. They compute models' hallucination tendency by perturbing source sequences and measuring how often a perturbation can lead the model to emit a hallucination. They identify that hallucinations are more prominent among examples with high CM. We return to CM in chapter 3, where we study CM in NMT much more broadly, relating CM scores to models' generalisation performance, akin to Feldman and Zhang and Zheng and Jiang.

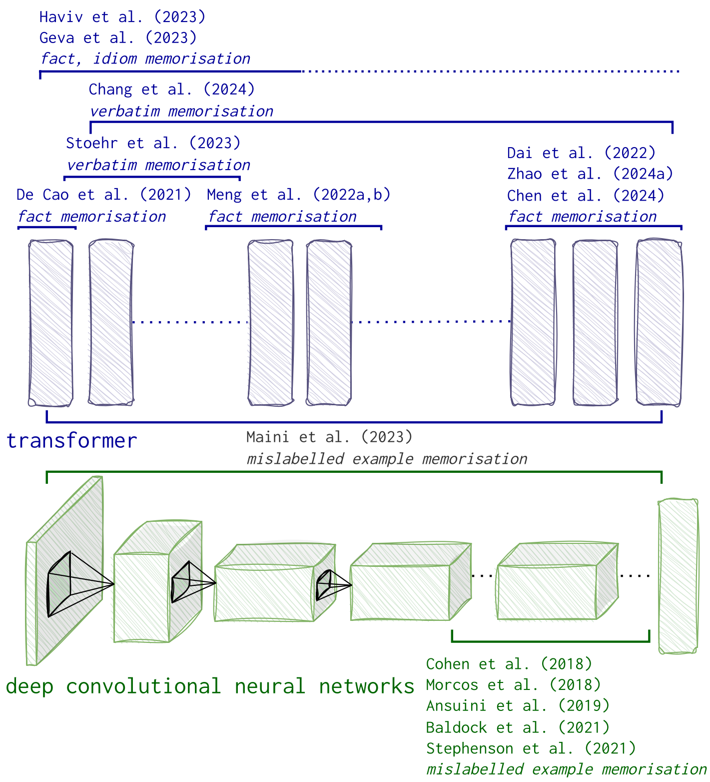


Figure 2.8: Generic indication of the range of layers that have been pointed out as storing memorised information.

2.2.3 Memorisation implementation

A final thread of related work we need to explore is that of the **implementation of memorisation**, questioning which parameters, neurons or layers store memories (focusing on *localisation*), how different components cooperate to retrieve memorised information (focusing on memorisation *mechanisms*) and how we can change that information (via *model editing*, since changing information might tell us something about how those memories were implemented initially). To review this work, we first step away from NLP again to learn from findings in CV, before elaborating on localisation, model editing and memorisation mechanisms for factual knowledge in NLP. We end with reviewing initial studies on verbatim memorisation, for which localisation is a new and active area of research at the time of writing the thesis. In chapter 4 we will focus on layer-based localisation of a specific type of memorisation, which is why this subsection has a particular focus on analysing the different conclusions regarding layer-based results, and we summarise those conclusions in Figure 2.8.

Noise memorisation in CV The observation of Zhang et al. (2017) that pretrained, deep CNNs can interpolate training sets sparked interest in how memorisation affects vision models internally. Two primary ways of studying that emerged, of which the first one applies label perturbation to an entire image classification dataset and trains memorising networks that interpolate that noise. Ansuini et al. (2019) trained a CNN from scratch on MNIST, showing that when randomly shuffling labels, the intrinsic dimensionalities of hidden representations are much larger than those of a control network in the final layers only. Cohen et al. (2018) compared multiple models, among which a Resnet-based model, to a k-nearest neighbour classifier (k-NN) fit using the models' hidden representations, effectively using k-NN as a probe. For the generalising networks, the k-NN probe starts to accurately match the models' predictions from lower layers onwards, but for the memorising networks, the predictions do not match until the final few layers, indicating that memorisation likely does not occur until those final layers. Morcos et al. (2018) trained 11-layer CNNs on CIFAR-10 using true labels and randomised labels, and measured the representational similarity of hidden representations using a CCA-based method (§2.1.4). They identified that networks that generalise well are much more similar to one another than to memorising networks, and that those differences are the most prominent in the final few layers.

Later work that studied memorisation localisation in CV uses individual networks instead of relying on comparisons between generalising and memorising networks, by comparing training data subsets. Baldock et al. (2021) established a positive correlation between prediction depth in image classification (the earliest layer that predicts the label) and example-level learning difficulty metrics. Stephenson et al. (2021) analysed the hidden representations for image classification using CIFAR-100 and tiny ImageNet, for multiple models among which ResNet-18. They permuted the labels of a training data subset and studied the geometry of the hidden representations of regular and memorised examples, measuring, for instance, the linear separability of classes. They reported that memorisation of mislabelled examples occurs abruptly in late layers (e.g. layer 15-20 in a 20-layer model) and late training epochs. Rewinding models' final or penultimate convolutional layer to earlier checkpoints partially reverted memorisation.

Contrary to earlier work, Maini et al. (2023) found that in partially mislabelled image classification datasets (e.g. CIFAR-10, MNIST), memorisation is not confined to specific layers but involves small sets of neurons dispersed across layers, for ResNet-9, ResNet-50, and a vision transformer.[6] Intervening by rewinding layers to earlier epochs or retraining individual layers with clean data showed no single layer alone drives memorisation. A gradient-based neuron search revealed that mislabelled examples required fewer neurons to be zeroed out than clean examples, with these neurons distributed across layers for both clean and mislabelled examples.

Although Maini et al. did not identify specific roles for deeper layers, their layer rewinding experiments, in particular, *do* suggest that rewinding mid to upper layers is more successful in reverting memorisation than rewinding lower layers. Figure 2.9

[6] The vision-transformer by Dosovitskiy et al. (2020) works akin to a regular transformer, but with the input representations being patches of an image instead of token embeddings.

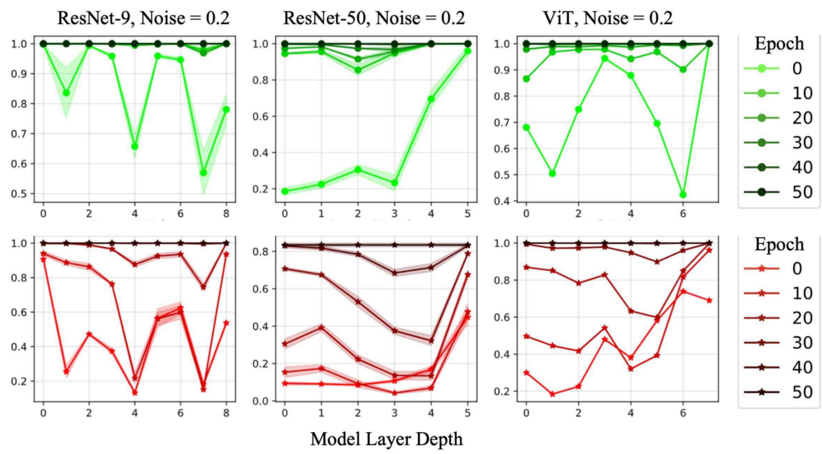


Figure 2.9: Figure 16c from the appendix of Maini et al. (2023), showing how the accuracy on clean (green) and mislabelled MNIST examples (red) decreases when rewinding individual layers to earlier epochs, for three models when mislabelling 20% of the examples. To simplify the visualisation, Maini et al. group layers into blocks for ResNet-50.

demonstrates this: if we disregard epoch zero (at which point the model has not learnt the task), the layers that show the largest accuracy decrease for mislabelled examples are four and seven, three and four, and four and five, for the three models, respectively. This holds across models, datasets, and noise levels, except for the vision transformer.

Memorisation of factual knowledge NLP memorisation localisation studies have primarily focused on factual knowledge. De Cao et al. (2021) first connected work from CV to fact memorisation in transformer, and trained a hypernetwork to edit facts. Their hypernetwork mostly edited the bottom layer of a six-layer transformer, and De Cao et al. pointed out the contrast to findings from Stephenson et al. (2021), suggesting this difference might be due to the change in modality.

Later work operated under the assumption of the *knowledge neuron thesis*,[7] assuming that facts are stored by and retrieved from transformer's feedforward layer weights, which act as a key-value memory (Geva et al., 2021), and that one may thus be able to identify knowledge neurons inside the feedforward layers (Dai et al., 2022). The knowledge neuron hypothesis inspired model editing techniques that mainly target the feedforward layers, such as the methods proposed by Meng et al. (2022, 2023). Meng et al. (2022) applied a two-step procedure of localising the layers that store facts, and editing them afterwards. Localisation was performed with *causal tracing*, a method that corrupts hidden representations with noise and assigns importance to layers in which restoring the original representations recovers the original prediction. Causal tracing was used to pinpoint a layer, after which the feedforward module in that layer

[7] The term was coined by Niu et al. (2024) to summarise the hypothesis underlying multiple related studies. Niu et al. criticised the thesis since it oversimplifies knowledge storage. Instead, they suggested focusing on network-wide circuits.

was edited to modify factual memories. Early to mid-layers were most often selected by causal tracing[8] – e.g. in a 48-layer model, the layer importance increases up until layer 15, after which it starkly decreases. Meng et al. (2023) later extended the approach to edit multiple layers at the same time, again based on the causal tracing results.[9]

In addition to identifying which layers are the most relevant, related work also directly attempted to identify individual knowledge neurons. Dai et al. (2022) identified them by integrating gradients using a fill-in-the-blank test for factual information, and mostly found knowledge neurons in the *top* layers of `BERT`. They then successfully used those neurons to update facts by partially modifying the second layer in the feedforward modules. Similar findings have been reported in later work on knowledge neurons by Zhao et al. (2024a), who selected neurons based on extreme activation values, and Chen et al. (2024), who also used integrated gradients. Although Zhao et al. echoed the finding of knowledge neurons primarily residing in the upper layers of multilingual `BERT`, they did observe a relative increase of knowledge neurons in lower layers when comparing results for Malay and Indonesian to English and German.

Work beyond model editing and knowledge neurons emphasises the relevance of lower layers for fact retrieval: Haviv et al. (2023) analysed the memorisation of facts and idioms in `BERT` and `GPT-2` by projecting hidden representations into the vocabulary space. They identified a two-phase process: the lower layers promote the correct prediction's rank, while the upper layers strongly increase the probability of that prediction. By intervening in a feedforward sublayer, Haviv et al. demonstrated the significance of the first phase, as earlier layer interventions were more likely to prevent the model from emitting a memorised token, with the first three layers having the greatest impact and the effect decreasing thereafter. Follow-up work by Geva et al. (2023) provided a more fine-grained description of the mechanisms behind fact retrieval when predicting an attribute based on a subject and relation: in `GPT-2-XL` and `GPT-J`, early feedforward layers enrich the subject's representation, early/mid attention layers pass information about the relation to the final token, after which mid/upper attention layers integrate information about the subject into the final token. By intervening in ten consecutive layers, they identified that the attribute prediction accuracy starkly decreases when intervening in the earliest feedforward layers. All in all, these two descriptions of the mechanisms underlying fact and idiom retrieval emphasise the role of early layers for storing memories, while explaining that all layers cooperate to produce correct memorised predictions at the output level.

[8]This result is not specific to transformer: Sharma et al. (2024) found early/mid-layers to be important when editing facts in `Mamba`.

[9]Note that Hase et al. (2023) found success in model editing to be unrelated to the layers selected by Meng et al.'s localisation method, which means that model editing might be an unreliable way to check where facts are stored.

Verbatim memorisation Memorisation localisation has predominantly focused on facts, but studying the implementation of verbatim memorisation has recently gained some traction thanks to open science initiatives publishing pretraining corpora. Chang et al. (2024) evaluated localisation methods on verbatim memorisation in `Pythia` and `GPT-2-XL`. They (1) trained models to memorise sequences using fixed neurons per layer and tested whether localisation identifies these, and (2) identified memorised pretraining sequences and tested whether dropping top-ranked neurons prevents recall. Most experiments use a fixed neuron count per layer, limiting insight into the layers' roles. Two additional experiments, however, touch upon the roles of layers: for (1) they also considered localising neurons globally instead of per layer, in which case localisation primarily points to bottom layers, and for (2) they considered dropping out neurons in individual layers instead of all layers, in which case preventing recall gradually improved with layer depth. Dropping neurons globally was more successful than targeting individual layers, which underscores memorisation's distributed nature. Stoehr et al. (2024) studied a 12-layer model (`GPT-Neo-125M`, also analysed in chapter 4) to identify components responsible for 50-token verbatim memorisation. They performed parameter gradient attribution, focusing on the maximum value per weight matrix, per layer. For memorised (but not non-memorised) paragraphs, those values were particularly large for weights from the attention module in the lowest layers, except the bottom-most layer, and those weights could be edited to revert memorisation. They pointed to a specific attention head in the second layer as attending to rare tokens, hypothesising that these tokens act as 'triggers' for retrieving memorised paragraphs.

When considering all of these findings for factual memories and verbatim memorisation, it is evident that many conflicting conclusions have been drawn regarding memorisation localisation in transformer-based LMs. Yet, it is unclear where the disagreement comes from – e.g. using different models, different localisation methods or different evaluation metrics. As a result, we cannot determine whether these findings truly contrast with the noise memorisation conclusions from CV. In chapter 4, we will contribute a missing piece to this puzzle by analysing noise memorisation in NLP.

2.3 Compositionality and formulaic language

We will now review the topic of compositionality and its relation to formulaic language (§2.3.1). Interest in the compositionality of language has, since the late 2010s, sparked interest in how well NLP models' generalisation capabilities reflect language's compositional nature, giving rise to a new type of evaluation dubbed *compositional generalisation* evaluation. The general introduction will be followed by a summary of work on compositional generalisation (§2.3.2) and work on how NLP models handle formulaic language (§2.3.3) that appeared prior to the publication of chapters 5 and 6.

2.3.1 Compositionality as a continuum

Gotlobb Frege is generally taken to be the first to have formulated the principle of compositionality along with the claim that it is an essential feature of natural language (Frege et al., 1892), but the general idea underlying the principle has been discussed by many others (Carnap, 1947; Katz and Fodor, 1963; Putnam, 1975, i.a.). Intuitive arguments exist in favour of natural language being compositional, of which productivity (we can produce and understand infinitely many new meaningful sentences) and systematicity (our ability to understand certain expressions, such as “John loves Mary” being linked to our ability to understand others, such as “Mary loves John”) are the most well-known (e.g. Fodor, 1987). Pagin and Westerståhl (2010b) outline these arguments, along with many others. Rather than providing arguments for why compositionality would hold, let us focus on definitions of compositionality and the relation to formulaic language understanding, which are of relevance to this thesis.

Definitions and problem cases Let us consider the definition of Pagin and Westerståhl (2010a) as a starting point: assume that a grammar $\mathbf{E}$ consists of a set of linguistic expressions E, a set of atoms (words) A, and a set of syntactic functions Σ that can be recursively applied to generate expressions from atoms.[10] Using this terminology, we can build grammatical terms, for instance, assuming $\beta, \gamma, \delta \in \Sigma$ representing functions to construct a noun phrase, a verb phrase and a sentence, “The man kicked the ball” can be represented as $\delta(\beta(the, man), \gamma(kicked, \beta(the, ball)))$. We then need a function μ to represent the semantics of our grammar, yielding what has been referred to as the *function version* of the principle of compositionality (Pagin and Westerståhl, 2010a, p.254):

> *For every rule $\alpha \in \Sigma$ there is a meaning operation r_α such that if $\alpha(u_1, \ldots, u_n)$ has meaning, $\mu(\alpha(u_1, \ldots, u_n)) = r_\alpha(\mu(u_1), \ldots, \mu(u_n))$.*

The definition presupposes that sub-terms of meaningful terms are also meaningful. In formal semantics, a standard approach to defining meaning compositions has been through function application, e.g. via lambda calculus. More recent approaches, however, adopt richer lexical meaning representations than lambda terms can offer, such as through distributional semantics (Martin and Baggio, 2020).

Applying the basic definition of Pagin and Westerståhl recursively[11] and bottom-up yields a rather strict interpretation of compositionality. Weaker versions relax the

[10] This is a simplification, omitting the value function V that can accommodate homonymous atoms mapping to the same expression. For a more precise description, see Pagin and Westerståhl (2010a).

[11] Note that the definition is given by recursion over syntax, but that standard semantic theories also define μ to be recursive (Pagin and Westerståhl, 2010a, p.254). The type of recursion considered here is somewhat different from the typical interpretation in computer science; due to μ mapping a term to an abstract semantic representation, it cannot be applied to its own outputs.

requirement that the meaning of a complex term is computed based on the meanings of the immediate sub-terms and the top-level syntactic operation, with the extreme case assuming that the meaning of an entire expression depends only on the meanings of the atoms and the global syntactic structure. We can relate this to the less formal definition introduced in chapter 1, by Partee (1984, p.153):

> *The meaning of an expression is a function of the meanings of its parts and of the way they are syntactically combined.*

In Partee's definition, no explicit restrictions are placed on the relationship between compound expressions and their parts. The type of *function* that relates those two components is unspecified and could thus take into account, for instance, the sentence's global syntactic structure, or even include external arguments, such as discourse. Furthermore, it is ambiguous what '*they*' refers to: the parts themselves or the meanings of the parts – i.e. do parts receive meaning in isolation or in context? A permissive reading is, therefore, similar to the weakest version discussed above (see Szabó, 2012, for an elaborate discussion of the many readings of Partee's definition). The many definitions have been referred to as ranging from weak to strong, but also from top-down to bottom-up (Baggio et al., 2012) or from global to local (Szabó, 2004). In chapter 5 we primarily refer to 'local' vs 'global', but the underlying considerations are similar to what is outlined above, considering strict vs weak (with the exception that we do not consider global compositionality to include extra-sentential inputs within the context of analysing computational models).

The appropriate definition of compositionality for natural language and natural language problem cases for compositionality have been widely discussed. Commonly mentioned issues are that the principle is trivially true (e.g. Horwich, 2001) or formally vacuous (e.g. Zadrozny, 1994) – since for any language we could construct syntactic functions and meaning operators that make the language compositional – or that radical contextualism means that the literal meaning of expressions is never adequate if their meaning always varies from context to context (e.g. Travis, 1985). If meaning is always contextual, we cannot capture infinitely many contexts in a finite way.

Examples of problem cases are belief sentences ("You believe she is a child doctor" can be true and "You believe she is a paediatrician" can be false, even if 'child doctor' and 'paediatrician' are synonymous; Pelletier, 1994), quotation ("I like Tully" and "I like Cicero" are synonymous if Tully is Cicero, yet, that does not mean "I like 'Tully'" and "I like 'Cicero'" are synonymous; Pagin and Westerståhl, 2010c) and fictional discourse (the fact that the two peanuts in "The peanut was in love" and "The peanut was salted" do not appear synonymous demonstrates the context-sensitivity of meaning compositions; Baggio et al., 2012). Lastly, most relevant to the chapters in this thesis, idioms have been pointed out as a problem case (Westerståhl, 2002). If we consider the

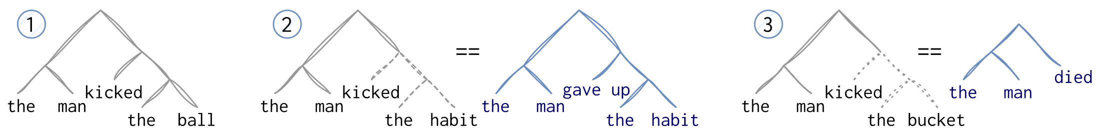


Figure 2.10: Three simple phrases with similar syntactic constructions but very different meanings. The meaning of (1) appears fully compositional; the meaning of (2) is less compositional but can be made compositional when interpreting "kicked" as "gave up". In (3), "kicked the bucket" can only be explained by treating the phrase as one atomic term.

meaning of $\delta(\beta(the,man),\gamma(kicked,\beta(the,ball)))$, we cannot use the same functions β and γ (and meaning operators r_β and r_γ) when representing the meanings of "the man kicked the bucket" and "the man kicked the habit" – meaning that the man has died or has given up a habit, respectively. We would have to extend the syntax and semantics to account for the different types of compositions involved in these new phrases, either by adding new operators or introducing some phrases as atoms within our grammar. There is no one-size-fits-all solution, since idioms themselves are not uniform, as Figure 2.10 illustrates. We might be able to represent the meaning of "kick the habit" with a new function that transforms "kicking" into "giving up", but we cannot treat "kick the bucket" in the same way. As a result "kick the habit" appears more compositional than "kick the bucket". Idioms are just one example here posing issues for compositionality, but similar arguments could be made for other types of figurative and formulaic language, such as proverbs or non-compositional compounds.

Nonetheless, it is widely accepted that natural language *is* compositional, even if we cannot agree on one definition (Dowty, 2007). Context and figurativeness influence expressions' meanings, but there is value in capturing literal meaning through syntax and semantics. Baggio (2021, p.8), for instance, proposes a competence version of the compositionality definition, stating that there is at least one interpretation of a compound expression that is determined only by the meanings of its parts and the way they are syntactically combined.[12] At the same time, it is well-known that not all phrases are equally compositional. We could extend our grammars to include all known figurative and formulaic phrases (e.g. Westerståhl, 2002, shows how to incorporate idioms in compositional semantics), but what if a new metaphor comes along? Do we continually invent new meaning operators? Instead of claiming that natural language is fully compositional or choosing a definition, I acknowledge that natural language exists along a compositionality continuum. I will default to referring to idioms as non-compositional, but realise that that is a matter of nomenclature rather than principle.

For the purpose of analysing computational models of language, instead of being hung up on an exact definition of compositionality, it is important to acknowledge the

[12] Whether or not this is true could in itself be debated, when one considers phrases like "by a dint of", "spick and span", and "odds bodkins", which all contain words we no longer use outside of these contexts. As a result, these phrases do not appear to have a literal default reading.

representational co-existence that humans have (Baggio, 2021), where they can interpret most natural language expressions both compositionally and non-compositionally. That this applies to human formulaic understanding is elaborated upon in the next paragraph, and that this is (or should be) the case in computational models is revisited in chapters 5 and 6.

The representational co-existence of formulaic sequences In §1.2, I previously introduced the notion of formulaic sequences. They differ from novel sequences in primarily four properties (van Lancker Sidtis, 2012): Firstly, in terms of *form*: they are not as flexible as other sequences since they only preserve their formulaic meaning when presented in a certain form, e.g. "he kicked a bucket" no longer means 'dying'. Secondly, in terms of *meaning*, since they signal a complex meaning that goes beyond a straightforward composition of the terms contained in the sequence. This meaning is conventionalised and shared by speakers of a language. Thirdly, there are *contextual conditions* for when usage of formulaic sequences is appropriate and context determines whether or not a sequence is formulaic ("The man kicked the bucket off the pavement" is not formulaic). Lastly, speakers need *personal knowledge* of a formulaic sequence; they need to memorise the conventionalised meaning and the properties in order to use or understand the sequence appropriately. While the connection to memorisation might not be as apparent to native speakers of a language, the wide range of literature dedicated to strategies for teaching formulaic sequences to second language learners (e.g. see Pellicer-Sánchez and Boers, 2018, for an overview), and the evidence of dedicated neural substrates for formulaic sequences (Wray, 2002) underscore their special nature.

'Formulaic sequence' is an umbrella term for many different types of sequences, that themselves are on a continuum of being more novel (compositional) to more reflexive (non-compositional): collocations (e.g. "to make the bed") for instance are considered more novel than idioms, proverbs and expletives (e.g. "good heavens"), which are themselves considered to be more novel than pause fillers, cries and vocal gestures (van Lancker Sidtis, 2012). Because the thesis will primarily focus on idioms (in chapters 5 and 6, with proverbs and non-compositional compounds also being considered in chapter 3), I do not elaborate on the different types here. For idioms, their formulaicity is dependent on their figurativeness: while figurative language and formulaic language are sometimes taken to be synonymous, there are differences, for instance, in the case of metaphors. Although all metaphors are figurative, only conventionalised metaphors (such as "couch potato") are formulaic, since only conventionalised phrases can be memorised, whereas novel metaphors need to be processed on-the-fly. The context modulates whether the words in an idiom are used figuratively or literally, and they are only to be understood non-compositionally in a figurative context.

Because of this potential competition between interpretations, idiom processing in humans has been widely discussed. Opposing views on the matter can be characterised as literal-first vs figurative-first. An example of the former is the *standard pragmatic view* suggesting humans attempt a literal interpretation first and only consider a figurative one in case of a contextual discrepancy (Bobrow and Bell, 1973; Grice, 1975, 1989). An example of the latter is the *direct access view*, suggesting that the figurative meaning can immediately be retrieved (Gibbs Jr et al., 1994) since figurative phrases have processing advantages over literal ones. However, this advantage mainly holds for very conventionalised figures of speech and ignores the impact of context. Context can facilitate fast processing of both the literal and figurative meaning, depending on whether the context is literally or figuratively biased (Holsinger, 2013). The more modern *hybrid view* posits that idioms are simultaneously processed as a whole and word for word (Caillies and Butcher, 2007). The processing speed and retrieval of the figurative meaning then depend on the idiom's semantic properties, and the context (Cain et al., 2009; Vulchanova et al., 2019). Examples of semantic properties are familiarity (how often one encounters the idiom) or meaningfulness (how well one knows the idiom's meaning) (e.g. Bulkes and Tanner, 2017). The most widely discussed property is decomposability or compositionality (Nunberg et al., 1994). The prototypical example of "kick the bucket" would be non-compositional, whereas, as discussed above "kick the habit" is more compositional once the metaphorical meaning of "kicking" is known.

Thus, not only do types of formulaic phrases lie on a compositionality continuum, but individual idioms exist along a continuum, too. In chapter 6, we lack the appropriate resources to subcategorise the idioms analysed based on their properties. However, we do ground our analyses in this literature by studying (1) the role that context plays in idiom disambiguation, and (2) the extent to which words in an idiom act as one phrase.

Compositionality and idioms in the context of translation Before moving on to discussing related work from NLP, I will comment on how (non-)compositionality has played a role in MT and motivate MT as a testbed for evaluating models' compositional behaviour. The principle of compositionality has explicitly informed the creation of some traditional MT systems (des Tombe et al., 1985; Rosetta, 1994). Janssen (1998, p.51) states the principle of compositionality of translation as follows:

> *The translation of a compound expression is a function of the translations of its parts and of the rule by which the parts are combined.*

Different from other NLP tasks – e.g. summarisation, which removes input content, or sentiment analysis, where meaning beyond sentiment is irrelevant – MT should be precisely meaning-preserving, which a compositional translation would be. However, designing a system for compositional translation encounters challenges akin to defining compositionality itself. Strictly compositional translations often yield nonsensical outputs: a correct translation might have different syntactic structures than the source sequence, and overly compositional translations of special terms in the source – such the quoted term in "'Bucket' has six letters", or the idiom in "The man kicked the bucket" – are simply incorrect. In translation, the 'function' from the definition above should be flexible enough to handle such cases yet strict enough to capture translations of a potentially infinite number of sentences in a finite system – e.g. when conjoining two sentences, we probably do want the translation of the conjunction to be a conjunction of the translation of the parts. Thanks to these nuances and MT's meaning-preserving nature, we deem it a suitable and interesting testing ground for evaluating models' compositional capabilities when dealing with natural language in chapter 5.

While translating idioms compositionally is often incorrect, it is not always clear what the correct strategy would be. Baker et al. (1992) discuss strategies for human translators: using an idiom from the target language of similar meaning and form, using an idiom from the target language with a similar meaning and a different form, copying the idiom to the translation, paraphrasing the idiom or omitting it. In the absence of idioms with similar meanings across languages, paraphrasing is most common. Since no comprehensive database aligns the world's idioms, automating translation strategies for idioms is challenging, and Baker et al. argue that native speakers are best suited for the task. One project studying the cross-lingual occurrences of idioms is that of Piirainen (2012), which identified about 500 idioms that (mostly Indo-European) languages have in common. When we analyse idiom translations in chapter 6, we unfortunately cannot accurately distinguish between all translation types of Baker et al. in an automated way. Instead, our main focus will be on distinguishing paraphrases from literal translations, but the reader should remember that compositional translations of idioms are not always incorrect due to the occasional cross-lingual sharing of figurative units.

2.3.2 Compositional generalisation evaluation

Humans can produce and understand language in a compositional manner, but does the same hold for computational models? This has been widely debated for predecessors of the modern neural network: Fodor and Pylyshyn (1988) argued that *connectionist models*, as associative mechanisms with distributed representations, could not capture the structure and combinatorial nature underlying language. Reviewing their arguments and the discussions that followed is beyond the scope of this thesis, but it should be noted that what became known as compositional generalisation evaluation is essentially a 'second wave' of interest in models' compositional abilities. In NLP, this second wave started around 2018. Liška et al. (2018) presented a lookup tables task, demonstrating that recurrent sequence-to-sequence models struggle with composing atomic functions, and that although generalising models can be found when training thousands of networks, they are extremely rare. Lake and Baroni (2018) and Loula et al. (2018) presented the widely adopted SCAN task, which constitutes a simple navigation task with inputs such as "`jump twice and walk`", mapped to a sequence of actions ("`JUMP JUMP WALK`"). Recurrent models struggled to generalise to longer sequences than the ones seen during training, failed to perform zero-shot generalisation for actions only seen without context during training, and failed to compose function words in novel contexts.

In the years following, a range of datasets were proposed, and the corresponding results underscored models' inadequacy to generalise outside of the IID paradigm, for both recurrent and transformer models. Yet, different from other types of *out-of-distribution* (OOD) evaluation (e.g. across-domain generalisation, see Hupkes et al., 2023, for an overview of OOD evaluation types in NLP)[13], compositional generalisation is in itself underspecified due to the potentially infinitely many new ways in which language can be composed: how exactly do we test whether models have human-like compositional abilities then? The evaluations of Liška et al. and Lake and Baroni highlight the importance of recombining training commands in test data, length generalisation and zero-shot generalisation. In Hupkes et al. (2020)[11], my co-authors and I took a more principled approach by grounding five experiments in the literature on compositionality (a small subset of which we reviewed in §2.3.1), proposing the following five tests:

1. *Systematicity*: do models understand input token combinations unseen during training?
2. *Productivity*: do models generalise to longer sequence lengths than observed during training?
3. *Substitutivity*: do models assign the same output following synonym substitution in the input?
4. *Localism*: do models assign the same output to local and global compositions of the same input?
5. *Overgeneralisation*: do models overgeneralise exceptions to rules?

Hupkes et al. apply these tests to the PCFG SET dataset, a sequence-to-sequence task that implements string manipulation, e.g. "`remove_first A B C , echo D E F`" yields "`D E F F`", demonstrating strong performance decreases for the first three tests when compared to IID evaluation. Using the final two tests, they show that the neural models evaluated are not locally compositional, and that while they initially overgeneralise exceptions, they do memorise them later on in training. We adopt three of these tests in chapter 5, when evaluating compositional abilities of NMT systems.

In addition to artificial sequence-to-sequence tasks being used to test compositional generalisation (e.g. Bastings et al., 2018), semantic parsing and MT emerged as popular testing grounds. For semantic parsing, Finegan-Dollak et al. (2018) published new train-test splits for existing datasets that do not repeat the same SQL query in both

[13] I was a core contributor of this article, yet it is not a part of this thesis.

train and test data. Kim and Linzen (2020) and Keysers et al. (2019), instead, present entirely new datasets. Kim and Linzen design COGS, a semantic parsing dataset with OOD evaluation targeted at assessing a wide range of lexical and structural types of generalisation. Instead of targeting manually selected types of generalisation, Keysers et al. measured compositional generalisation using a maximum compound divergence metric, a corpus-wide metric that quantifies the novelty of token combinations in the evaluation set with respect to the training data. For MT, Lake and Baroni (2018) presented a toy task for evaluating the generalisation for a primitive introduced during training, Li et al. (2021a) designed CoGnition, an English-Chinese dataset with simple sentences and a generalisation set with novel token combinations, and Raunak et al. (2019) investigated the systematicity and productivity properties of MT systems through analyses of their hidden representations. The general consensus within this line of work is that when presented with many types of tests for compositional generalisation, performance drops substantially compared to IID evaluation.

What most of these datasets have in common is that the data used to study compositional generalisation is either fully synthetic (not even using natural language) (e.g. Liška et al., 2018; Hupkes et al., 2020), is generated from scratch via a grammar or heuristics using a vocabulary with words from natural language (to generate simple but natural-sounding sequences) (e.g. in case of Kim and Linzen, 2020; Keysers et al., 2019), or is selected because of its simplicity, excluding constructions that contribute to the complexity of natural language, such as polysemous words or metaphors (Li et al., 2021a). To address this issue for semantic parsing, Shaw et al. (2021) adopted a variant of Keysers et al.'s metric but applied to conventional semantic parsing datasets. Yet, at the time of writing chapter 5, no existing approach had discussed compositional generalisation in the context of MT for systems trained on regular, natural language corpora. MT is well-aligned with the principle of compositionality thanks to its meaning-preserving nature, while also addressing a practical and important problem, without oversimplifying the (non-)compositional nature of language. Redefining tests for that context is one of the contributions of that chapter.

Parallel to the introduction of datasets for compositional generalisation, a range of methods have been proposed to improve models' compositional abilities, with notable trends being the development of specialised architectures (e.g. Korrel et al., 2019; Li et al., 2019; Russin et al., 2020; Liu et al., 2020), data augmentation and manipulation methods (e.g. Andreas, 2020; Herzig et al., 2021; Akyürek et al., 2021; Oren et al., 2021; Qiu et al., 2022), auxiliary training objectives (e.g. Jiang and Bansal, 2021) and meta-learning approaches (e.g. Lake, 2019; Conklin et al., 2021), and the identification of simple tips and tricks for the experimental setup that can already boost performance (Csordás et al., 2021; Ontanon et al., 2022). Since many of these approaches cannot be applied when training models such as NMT systems on natural language, or have not been shown to scale to more complex tasks than originally evaluated on (Furrer et al., 2020), further reviewing these methods lies outside of the scope of this thesis.

2.3.3 Formulaic language understanding in NLP

Reports of how formulaic language poses problems for NLP systems have been around for decades. Sag et al. (2002) laid out challenges related to English (idiomatic) multi-word expressions and strategies for how to explicitly encode them in a grammar. In the years that followed, the tasks of multi-word expression identification and disambiguation gained traction, with the research being propelled by initiatives such as dedicated workshops (e.g. Tanaka et al., 2004; Moirón et al., 2006; Anastasiou et al., 2009). Models for idiom understanding evolved from rule-based systems to corpus-driven statistical methods in the 2000s (see Rayson et al., 2010, for an overview of approaches), to (contextualised) embedding-based approaches in the last decade (e.g. Shwartz and Dagan, 2019). In chapters 5 and 6, we examine idioms in the context of NMT, focusing on how translations evolve during training and which internal mechanisms support non-compositional outputs.

Idioms in machine translations Idiom understanding is most commonly investigated through the task of idiomaticity detection directly. However, models' lack of formulaic language understanding negatively affects performance in other tasks, too, although this is not always well-documented due to a lack of resources for quantifying those effects. MT is one of the tasks negatively affected by this – e.g. Santos (1990) already proposed a specialised system to handle non-compositional translations. Only a limited number of articles have reported on the effects of idioms on NMT systems, among which are Isabelle et al. (2017), who created idiom challenge sets for English-French translation, Shao et al. (2018), who compiled a dataset of Chinese-English idiom translations and Rikters and Bojar (2017) and Fadaee et al. (2018), who filtered existing MT datasets for idiomatic examples for English-Latvian/Czech and English-German MT, respectively. The results all demonstrate a decrease in performance for idiomatic examples compared to conventional test sets[14] and Shao et al. emphasised a common error type in idiom translations, namely, overly literal translations. They introduced a blocklist with literal translations of the words contained in idioms, used to identify such errors. In chapters 5 and 6, we adopt a similar method when analysing transformer's translations of idioms.

At this point, it is worth adding a caveat. Most articles discussing low-quality idiom translations rely on standardised NMT evaluation metrics based on n-gram overlap between system and gold-standard translations, such as BLEU. As reviewed in §2.3.1,

[14]These articles predominantly focus on recurrent models; we include them for completeness, but focus on transformer, ourselves. In §6.6 we elaborate on work that appeared after the publication of chapter 6 that does adopt transformers.

multiple strategies exist for translating idioms, and when an equivalent target idiom does not exist, paraphrasing is preferred. However, there is often more than one valid paraphrase. For instance, in case of "out of the blue", which could map to "uit het niets" ("*out of nothing*"), "plotseling" ("*suddenly*"), or even "als donderslag bij heldere hemel" ("*like a thunderclap in a clear sky*"). As a result, n-gram overlap might underestimate the quality of systems' translations when models paraphrase idioms differently from the gold standard. While metrics' insensitivity to paraphrasing is not unique to idioms, it is likely more prominent for non-compositional than for compositional phrases, although this has not been quantified by previous work. Applying semantics-aware neural evaluation metrics is expected to partially alleviate this problem. At the same time, even if our evaluation metrics cannot exactly quantify the magnitude of the problem, we can still confidently say that models *do* struggle with idiom translations; studies that include human annotators in the evaluation process confirm this (e.g. Isabelle et al., 2017; Avramidis et al., 2019).

Techniques for improving idiomatic translations include extending the training set with more examples of idiomatic translations (Rikters and Bojar, 2017; Zaninello and Birch, 2020), explicitly annotating the idiom in the input (e.g. by encoding an idiom as one input word, Zaninello and Birch, 2020) and fusing representations from a pretrained model like BERT with NMT systems' representations (Huang et al., 2021).

Analysing transformer's representations for formulaic language Little is known about what enables the detection of idioms within NMT systems. For this reason, we draw on more general findings concerning how formulaic and figurative language influence the hidden representations of transformer – a key focus of chapter 6. Several studies have used hidden representations to distinguish figurative from literal occurrences of various linguistic tropes. Shwartz and Dagan (2019) studied verb-particle constructions, light verb constructions, *noun compounds* (NCs) and adjective-noun pairs. Using contextualised representations (among which those of BERT), they identified that models are better at detecting figurative meaning shift than at predicting implicit meaning – e.g. a model might be able to identify that "a hot argument" is figurative, but struggles with determining whether the phrase conveys temperature. Kurfalı and Östling (2020) detected idioms based on the dissimilarity of BERT's representations of the potentially idiomatic expression and its context, assuming that contextual discrepancies indicate figurative usage. García et al. (2021a,b) examined how contextualised representations (including those of BERT) represent figurative and literal NCs, using in-context and out-of-context representations. They demonstrated that the context of NCs hardly affects whether the representations encode figurativeness, and that the representations do not adequately capture non-compositionality since synonym replacements hardly affected the representations of non-compositional NCs. In search of the *idiomatic key* of verb compounds (the part of the input that cues idiomatic usage), Nedumpozhimana and Kelleher (2021) trained a probing classifier to distinguish literal usage from figurative usage. They then compared the effect of masking the compound in the input to masking the context on the classifier's performance and concluded that the idiomatic key mainly lies within the compound itself, although some information comes from the surrounding context. Taken together, these analyses suggest that hidden representations can, to some extent, be used to determine whether transformers capture idiomaticity, but the context may be insufficiently leveraged when disambiguating figurative expressions.

In chapter 6 we both analyse hidden representations and examine transformer's attention patterns for idioms. Little is known about how formulaic language affects attention mechanisms, though related research has examined ambiguous nouns. That type of ambiguity differs from idiomaticity due to the absence of non-compositionality, but the challenge of disambiguation is similar. Tang et al. (2019) demonstrated that noun ambiguity can be reliably predicted from hidden representations, and transformer's self-attention patterns in the encoder reflect this ambiguity through more distributed weights across the context for ambiguous nouns. However, cross-attention in transformers appears not to contribute meaningfully to disambiguation; Tang et al. (2018) found that ambiguous nouns received more cross-attention than non-ambiguous ones during translation, suggesting that cross-attention weights alone do not facilitate disambiguation.wen

Part I

Memorisation in transformer

Now that we have reviewed the background on transformer models, and the general topics of memorisation and (non-)compositionality, let us dive into three key open questions pertaining to memorisation: why are certain datapoints more likely to be memorised than others, is data memorisation beneficial for models' generalisation performance, and which internal parts of a model are the most crucial for memorising specific datapoints? The two chapters that follow explore these questions through distinct experimental setups. Whereas chapter 3 focuses on memorisation from the data perspective, chapter 4 shifts the focus to model internals, examining which of transformer's layers implement memorisation most.

When focusing on the data perspective first, it is evident that for some examples, models produce the same prediction independent of whether they were seen during training, whereas other examples must be memorised during training, or else models would never predict the corresponding targets. This is not a binary distinction; when presenting a model with millions of training examples, those examples will vary in the extent to which they require memorisation. In chapter 3, based on Dankers et al. (2023), I quantify that variance for the task of NMT by using continuous metrics to put training examples on so-called 'memorisation maps'. Using those maps, I study which examples are memorised most, how surface-level features influence a datapoint's position on the map, and how the presence of examples with specific memorisation scores in the training set is related to models' generalisation performance.

I turn to model-internal mechanisms in the chapter that follows: chapter 4, based on Dankers and Titov (2024), examines which layers are most involved in the memorisation of mislabelled examples, when performing natural language classification using transformer-based LMs. The focus on mislabelled examples provides a controlled setting to isolate the effects of memorisation; it allows us to focus on the internal memorisation mechanisms without having to worry about the fact that memorisation is not always that clear-cut (as emphasised in chapter 3). In this chapter, too, we focus on the memorisation-generalisation connection by studying a hypothesis suggesting that earlier layers are more involved in implementing generalisable features and that deeper layers specialise and memorise.

Chapter 3

A memorisation–generalisation continuum of data

3.1 Introduction

When training neural networks, we aim for them to learn a generic input-output mapping that does not overfit on the examples in the training set and generalises to unseen examples. In other words, we expect models to *generalise* without fully *memorising* the training set. However, adequately fitting a training dataset that contains natural language data inevitably means that models will have to learn the idiosyncrasies of that data, and it, therefore, does require memorising subsets of the training set (Feldman, 2020; Zheng and Jiang, 2022; Zhang et al., 2023). What models do and do not memorise, and how that relates to their performance, remains quite elusive. Both memorisation and task performance increase with model size (Carlini et al., 2022), and models memorise more than we would want them to. ChatGPT can recall detailed information from its training data, such as named entities from books (Chang et al., 2023), `GPT-2` memorises PII (Carlini et al., 2021), and for some source sentences that share a common prefix, transformer NMT systems have memorised to always emit the same translation (Raunak and Menezes, 2022). And yet, models do not always memorise what we would want them to – e.g. similar to other NMT systems, `mBART` was found to provide overly literal translations of idioms, signalling a *lack* of memorisation (Baziotis et al., 2023). These examples illustrate the multi-faceted relation between memorisation and generalisation. As discussed in §2.2, memorisation is no longer considered solely detrimental in NLP, and deep learning more broadly. Memorisation of random data can be benign when models maintain a low generalisation error (Zhang et al., 2017; Bartlett et al., 2020), and memorisation of long-tailed data distributions has been argued to be beneficial for generalisation performance (recall Feldman's LTT discussed in §2.2.2).

In this chapter, we get a step closer to understanding the relation between memorisa-

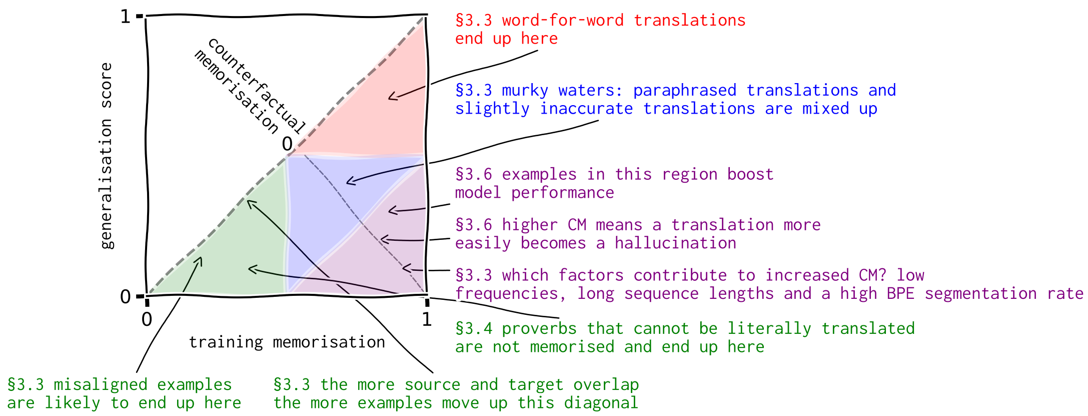


Figure 3.1: Illustrative summary of findings for different areas of the memorisation map. Counterfactual memorisation subtracts the y-coordinate from the x-coordinate.

tion and datapoints' characteristics for NMT, and put the LTT to the test for the NMT setting, thus contributing to the thesis's overarching questions RQ1 "*What characterises memorised examples?*" and RQ3 "*To what extent are memorisation and generalisation at odds with one another?*". We train NMT systems from scratch and put 5M examples from five language pairs on a memorisation map (as detailed in §3.2), constituting a continuum between memorisation and generalisation. The map is centred around the **counterfactual memorisation** (CM) metric (Feldman and Zhang, 2020) (see §2.2.2). Using the map, we address the following sub-questions, for which we illustrate takeaways and interesting findings in Figure 3.1:

1. *How do characteristics of datapoints relate to their position on the memorisation map?* In §3.3, we compute 28 quantitative features and annotate a data subset manually using 7 additional features. We discuss how features such as source-target similarity, input and output length, token frequency and tokens' segmentation relate to the memorisation map.
2. *How do datapoints containing formulaic phrases stand out on the memorisation map?* Although the majority of the experiments in this chapter pertain to all training datapoints, we include an intermezzo in §3.4, to examine memorisation scores of source sequences that contain proverbs, idioms and non-compositional compounds. While we would want these examples to be memorised *more*, they are memorised *less*, emphasising that what is actually memorised is not necessarily what should be memorised.
3. *Can we approximate memorisation metrics using datapoints' characteristics?* In §3.5, we use datapoints' characteristics to predict memorisation values using small regression models to consolidate findings from §3.3, to compare different languages to one another and to understand whether resource-intensive memorisation computation has cheaper approximates. We find that the regression

models generalise cross-lingually: characteristics' relation to memorisation is largely language-independent for the five language pairs we include.

4. *How does training on examples from different regions of the memorisation map change models' performance?* Finally, we relate different parts of the map to the quality of NMT systems in terms of BLEU, COMET, targets' log-probability and models' hallucination tendency (§3.6). Our results confirm previous work from other tasks – examples with high CM are most relevant for models' performance – yet there are caveats worth mentioning, in particular for the hallucination tendency.

After elaborating on the experiments, we end the chapter with a discussion (§3.7), commenting on the limitations of our approach, relevant work that appeared after the publication of this chapter, and how we think our results and data could benefit work going forward. In §2.2.2, we reviewed prior work for this chapter and already noted that, for NMT, memorisation is not that well explored. Most relevant is the work of Raunak and Menezes (2022) and Raunak et al. (2021). Raunak and Menezes computed extractive memorisation, a binary metric that identifies source sentences with a prefix for which models generate the same translation, independent of the prefix's ending. Raunak et al. computed CM scores in a low-resource NMT setup to show that hallucinations are more prominent among examples with higher CM values. We, too, treat memorisation as a graded phenomenon by using CM-based metrics. Whereas Raunak et al. solely explore CM in the context of hallucinations, we build a multilingual resource of memorisation metrics, examine the characteristics of datapoints that influence their position on the memorisation map, and investigate the relation to models' performance.

3.2 Experimental setup

This section details the memorisation metrics employed and the experimental setup for the model training that is required to compute those metrics.

Memorisation metrics To obtain a graded notion of memorisation, we employ the **CM** metric of Feldman and Zhang (2020) and Zhang et al. (2023) as detailed in Equation (2.24). The original CM formulation relies on a leave-one-out principle, which is too expensive computationally. We thus approximate CM as laid out in Equation (2.25): we train models on 50% of the training set and, afterwards, compute the 'IN' and 'OUT' quantities using all models that did and did not include this example in their training subset, respectively. Equation (2.25) relies on a generic performance metric M, instantiated by Feldman and Zhang (2020) using a target probability for image classification. Since NMT concerns the generation of sequences, we do not rely on the target's probability directly for M since that involves multiplying the probabilities of individual target tokens. As a result, sequences of different lengths would be incomparable. Instead, we adopt a *likelihood* metric (LL) which aggregates probabilities over tokens in the target sequence y_1^m of length m using the geometric mean, for a given source sequence x_1^n and model θ:

$$\mathsf{LL}(x_1^n, y_1^m, \theta) = \left(\prod_{t=1}^{m} P(y_t | y_1^{t-1}, x_1^n, \theta) \right)^{\frac{1}{m}} \tag{3.1}$$

The geometric mean is preferred over the arithmetic mean of target token probabilities, because it underscores probability's multiplicative nature – i.e. one wrong word can affect the entire translation's correctness – and is preferred over the arithmetic mean of log-probabilities because it is bounded. Alternatively, one could rely on a generation-based metric for M; in §3.3.2, we replace the likelihood measure with BLEU scores for greedily decoded hypotheses and reproduce a subset of the findings with those alternative maps.

CM is computed by taking the 'IN' performance and subtracting the 'OUT' performance; in this chapter, we refer to these two components as **training memorisation** (**TM**, which expresses how well a model performs if the example is in its training set) and the **generalisation score** (**GS**, which expresses how well a model performs when the example is unseen). CM is thus high for examples that can be predicted correctly if they are in the training set, but that a model cannot generalise to if they are not. Instead of approaching CM as a one-dimensional metric, we examine patterns that underlie TM, GS *and* CM.

Data Even when leaving out data subsets, computing the memorisation metrics is still resource-intensive. To balance the efficiency of the computation with the quality of the NMT systems, we use corpora with 1M examples for five language pairs: English (En) is the source language, and the target languages are German (De), Dutch (Nl), French (Fr), Spanish (Es), and Italian (It).[1] To enable direct comparison between languages, we collect parallel data by taking sentence pairs from the intersection of the OPUS corpora for these languages (Tiedemann and Thottingal, 2020). The use of multiple language pairs helps to ensure that the conclusions are not language-specific. The raw intersection of the language pairs mentioned above contained 4M sentence pairs, from which we select sentences based on four criteria:

1. The length of the source divided by the length of the target is between $\frac{2}{3}$ and $\frac{3}{2}$;
2. The punctuation ratio of the source and target sequence lies below 0.5;
3. Less than 30% of the words in the source can appear verbatim in the target, as well;
4. More than 90% of the digits contained in the target should also appear in the source sequence.

[1]Throughout the thesis, we refer to languages using their ISO 639-1 codes.

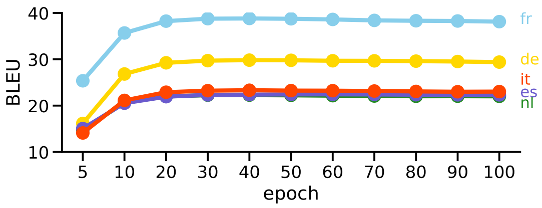


Figure 3.2: BLEU on the evaluation dataset FLORES-200, when training to obtain the memorisation scores.

The resulting data is relatively clean since the criteria above filter out potentially misaligned examples. What happens when using a *random* OPUS subset with much more noisy data? §3.3.2 elaborates on this. We tokenise our data using BPE tokenisation (Sennrich et al., 2016).

Training models to obtain memorisation metrics We train 40 models to compute our metrics repeatedly on a randomly sampled 50% of the training data, while testing on the remaining 50%. The models are `transformer-base` models (Vaswani et al., 2017), trained with `fairseq` for 100 epochs (Ott et al., 2019). To ensure that this leads to reliable CM scores, we compute CM scores based on two disjoint sets of models and examine how those scores compare, similar to Zhang et al. (2023). We compute CM scores based on seeds 1 to 20, and then compute CM scores based on seeds 21 to 40: these scores correlate with Pearson's r=0.94. When combining 40 seeds, the metrics are thus even more reliable.

We evaluate our models using the FLORES-200 'dev' set (Costa-jussà et al., 2022), a dataset created by expert translators for Wikimedia data. Figure 3.2 illustrates the BLEU scores on a development set over the course of training. At the time of writing, the top FLORES-200 'dev' performances on the OPUS-MT leaderboard were 40.4, 27.1, 51.5, 28.0 and 29.2 for DE, NL, FR, ES and IT, respectively. Of course, our models trained on a fraction of SOTA MT datasets underperform, but our relative differences in BLEU across languages are similar. In Appendix A we elaborate on the technical setup used for model training, and the licenses of the datasets used throughout this chapter.

3.3 Data characterisation: what lies where on the memorisation map?

We now have values for our memorisation metrics for 5M source-target pairs across five language pairs. We can view each source-target pair as a coordinate on a map based on the TM and GS scores associated with that example; the offset of the diagonal indicates the CM. Figure 3.3 illustrates the coordinate system for EN-ES. It represents datapoints using scattered dots, coloured according to CM. As is to be expected, the

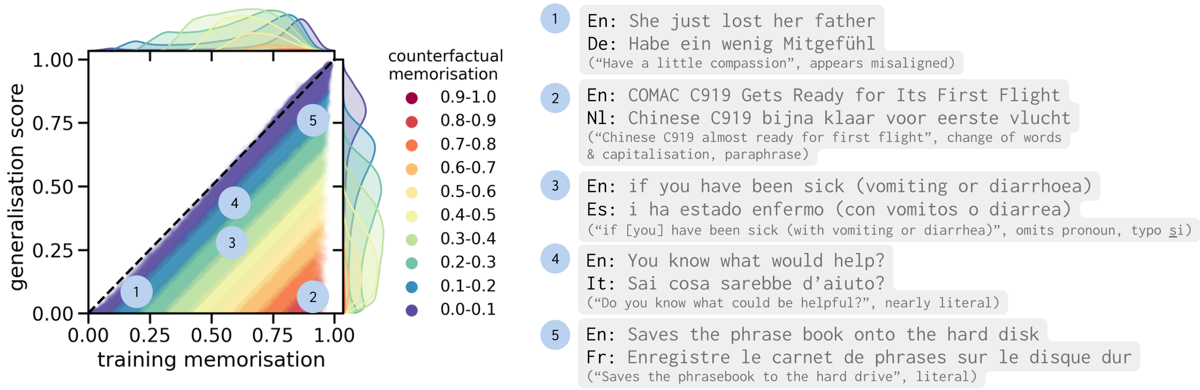


Figure 3.3: The memorisation map for En-Es, with five examples from the five language pairs and their approximate position on their respective memorisation maps.

TM values exceed the GS values, meaning that generating an input's translation is easiest when that example is in the training set. Examples with high CM are rare: few examples are *very* easily memorised during training while also having a *very* low GS. The figure provides five examples from the different language pairs, with an indication of where those example are on the language pairs' respective memorisation maps. The first example in the bottom left appears misaligned; the second example with very high CM demonstrates a case of changed formatting, paraphrasing and word replacement; examples three and four are straightforward translations but with slight differences between the source and target, and the fifth example is a very literal, word-for-word translation. Our interactive demo can be used to further examine individual examples on the map.

To better understand which characteristics influence a datapoint's position on this map, we next analyse the correlations between datapoints' features and the three metrics.

3.3.1 Analysis of feature groups

We compute 28 language-independent features that cover a broad spectrum of surface-level features of both the source and target. 16 features describe the source and target based on length and frequency (either before or after subword tokenisation), and 7 features capture the source-target overlap, using the source-target Levenshtein *edit distance* (ED), the Levenshtein ED of the source and the target's backtranslation (computed with models from Tiedemann and Thottingal, 2020), ratios of unaligned words, word/token verbatim overlap and the alignment monotonicity (as per the Fuzzy Reordering Score, Talbot et al., 2011). The remaining features are target repetitions, the BPE segmentation rate of the source or target, and digit and punctuation ratios of the source. The full list of features and details on the implementation of these features are included in Appendix A. For each feature, we compute Spearman's rank correlation (ρ)

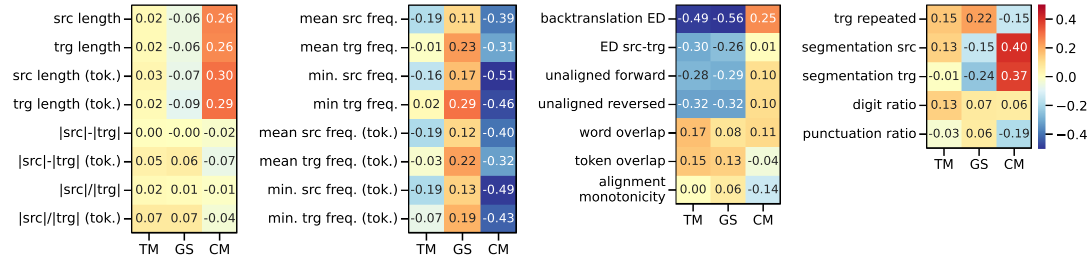
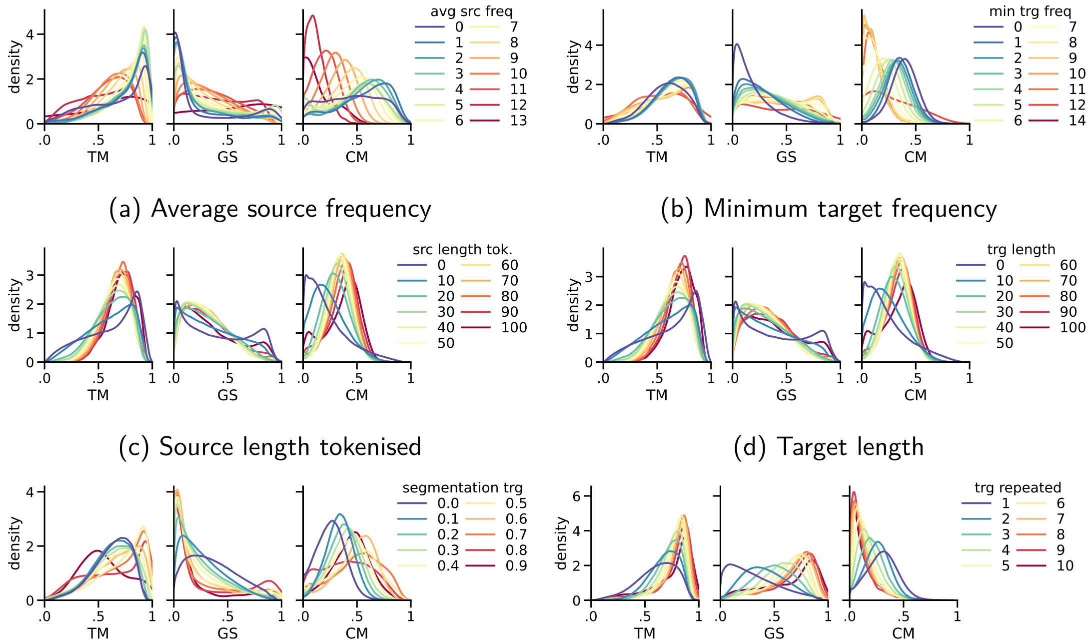


Figure 3.4: Correlations between memorisation metrics and features (Spearman's ρ), separated into length-, frequency-, and overlap-based features, and the remaining features.



(a) Average source frequency

(b) Minimum target frequency

(c) Source length tokenised

(d) Target length

(e) Segmentation degree target

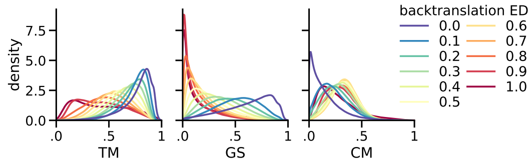
(f) Target repeated

Figure 3.5: Illustration of how six of the features capturing frequency, length, segmentation degree and target repetition relate to TM, GS and CM.

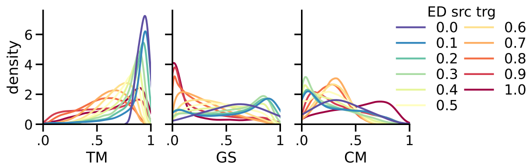
for TM, GS and CM, combining datapoints from all five language pairs. All correlations are contained in Figure 3.4; we will now review the most noteworthy patterns.

Frequency The frequency features are moderate predictors for CM (e.g. for the minimum target log-frequency feature, ρ_{CM}=−0.46, depicted in Figure 3.5b). Examples with low-frequency tokens can be learnt during training, but models are much less likely to assign a high probability to targets with low-frequency tokens during testing.

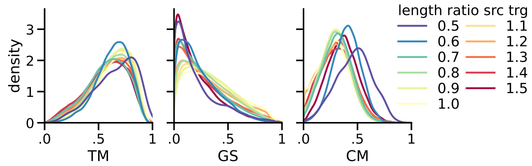
Length The length characteristics correlate more strongly with CM than with TM or GS (e.g. for the tokenised source length, ρ_{CM}=0.30, also visualised in Figure 3.5c). This means that longer sequences tend to have a larger difference in performance between training and testing time, compared to shorter sequences.



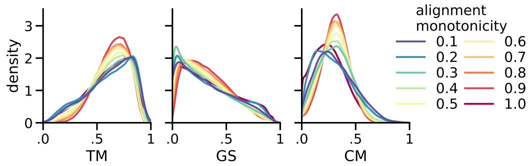
(a) Backtranslation ED



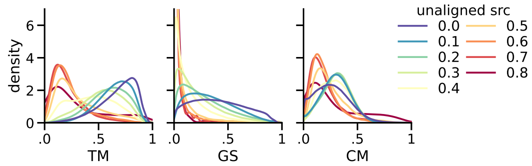
(b) src-trg ED



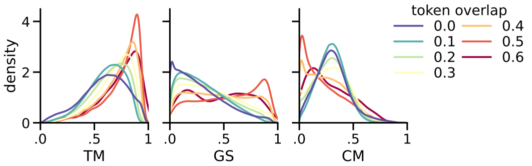
(c) Length ratio src/trg (tok.)



(d) Alignment monotonicity



(e) Ratio of unaligned source words



(f) Token overlap

Figure 3.6: Illustration of how six features capturing source-target overlap relate to TM, GS and CM.

Token segmentation Thirdly, the segmentation of words into tokens moderately correlates with CM (ρ_{CM}=0.40/ρ_{CM}=0.37 for source/target segmentation, respectively), as is shown in Figure 3.5e. The segmentation compares the number of white-space-based tokens to BPE tokens: $1 - \frac{|s|}{|s_{\text{BPE}}|}$. This is in line with the effect observed for frequency, since the BPE tokenisation scheme learns tokens based on frequency.

Repetitions A feature that is a weak positive predictor for TM and GS is the repetition of the target (ρ_{TM}=0.15, ρ_{GS}=0.22, ρ_{CM}=−0.15, see Figure 3.5f). This is expected, considering that similar targets have similar sources and are thus more easily memorised. Previous work already noted that repetition-related characteristics (repeated sentence 'templates') lead to high TM (Zhang et al., 2023).

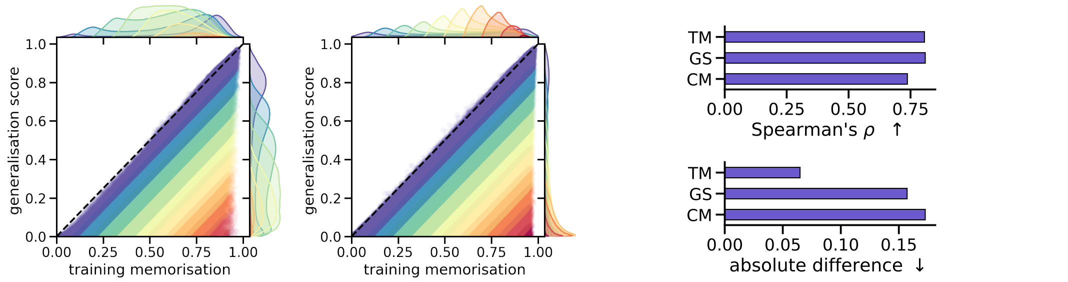
Source-target overlap The remaining features that correlate weakly or moderately with TM and GS are: the target's backtranslation ED to the source (ρ_{TM}=−0.49, ρ_{GS}=−0.56, see Figure 3.6a), the source-target ED (ρ_{TM}=−0.30, ρ_{GS}=−0.26, see Figure 3.6b), and the fraction of unaligned tokens (ρ_{TM}=−0.32, ρ_{GS}=−0.32 for target tokens, ρ_{TM}=−0.28, ρ_{GS}=−0.29 for source tokens, see Figure 3.6e). Apart from negative correlations, there are weak positive predictors, e.g. token overlap (ρ_{TM}=0.15, ρ_{GS}=0.13, see Figure 3.6f). These features express (a lack of) source-target overlap: source words are absent in the target, or vice versa. Because they are predictive of both TM and GS, they are not that strongly correlated with CM: they predict where along the diagonal an example lies but not its offset to the diagonal. While you might expect that examples with little source-target overlap *require* memorisation, their TM values remain low throughout the 100 epochs. The only relation to CM we observe is that, typically, examples in the mid-range (i.e. with some overlap) have higher CM than examples with extreme values (i.e. full or no overlap). CM thus highlights what models can memorise in a reasonable amount of training time. This provides a lesson for NMT practitioners: models are unlikely to memorise the noisiest examples, which might be one of the reasons why semi-automatically scraped corpora, rife with noisy data, have driven the success behind a range of NMT systems (e.g. Schwenk et al., 2021a,b).

3.3.2 Memorisation map variations

The memorisation maps we discussed are based on parallel OPUS and the target-likelihood metric. How do the maps change if we vary the corpus or the metric?

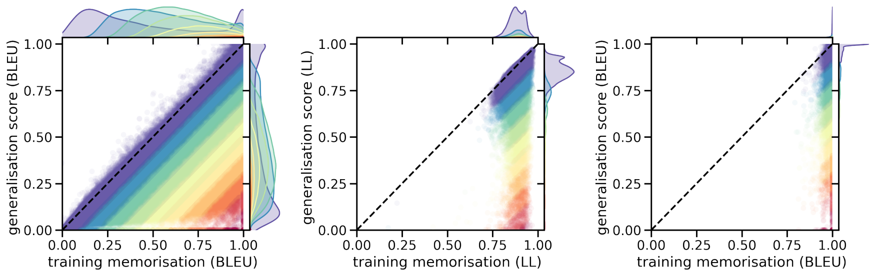
Firstly, we change the corpus from the 1M OPUS datapoints filtered as laid out in §3.2 to a random 1M examples, using the OPUS-100 subset for En-Nl of Zhang et al. (2020). Figures 3.7a and b show the memorisation maps for our parallel En-Nl OPUS and OPUS-100, respectively. The most striking differences are: OPUS-100 has many more datapoints with a CM score close to 1 (dark red, in the bottom right), and there are many more examples with a low GS. This is due to the more heterogeneous nature of the random corpus that includes more source-target pairs with unexpected tokens in the target (recall that we are computing a geometric mean over the target tokens' probabilities). For examples that the parallel OPUS corpus and OPUS-100 have in common, Figure 3.7c shows how the TM, GS and CM scores differ. They strongly positively correlate, but are still quite different in terms of absolute numbers. Hence, when varying the corpus, the memorisation trends are similar, but the exact score assigned to an example depends on the dataset composition. When we measure the correlations between the features we assigned to datapoints and our three metrics, the same patterns emerge: 95% of the correlations from Figure 3.4 have the same sign when replacing the corpus, with correlations' absolute values being slightly stronger across the board for OPUS-100 (+0.04). The most notable differences are that length is more strongly negatively correlated with GS, that length differences are now positively correlated with CM, and that frequency is now negatively correlated with TM.

Secondly, we change the performance metric M in Equation (2.25) from the target-based likelihood (LL) to a hypothesis-based metric by generating hypotheses using greedy decoding and measuring BLEU scores. Figure 3.8a displays the new BLEU-based memorisation map for En-Nl data (the map should be compared to Figure 3.7a). The examples generally lie closer to the diagonal, and the computation of our metrics is less stable across models: comparing CM scores from models with 20 seeds to those of 20 other seeds leads to Pearson's r=0.84 (it was 0.94 for the LL-based scores). When



(a) Parallel OPUS (b) OPUS-100 (c) Examples that are in both corpora

Figure 3.7: Memorisation maps for En-Nl computed using parallel OPUS vs. OPUS-100, and the differences in scores for examples the corpora have in common.



(a) BLEU-based map (b) High TM-BLEU (c) High TM-LL

Figure 3.8: The memorisation map for En-Nl when computed using BLEU instead of LL, and illustrations of where highly memorised examples from one map reside in the other.

comparing the two sets of LL- and BLEU-based metrics, the TM and GS metrics correlate very strongly with Spearman's ρ of 0.89 and 0.80, respectively, although the CM correlation is more moderate ($\rho = 0.54$). Examples that the model fully memorises (BLEU>99 or LL>0.9) reside in the same area on the two maps, as shown by Figures 3.8b and 3.8c. When we again examine the correlations between the datapoints' surface-level features and the three metrics, and compare those to the results from Figure 3.4, 93% have the same sign, with the absolute values of correlations being lower across the board for the BLEU-based metric (-0.07). The most noticeable qualitative differences are that CM is less strongly correlated with length features, and that the backtranslation source-target ED is no longer a weak positive predictor of CM.

In the remainder of the chapter, we will rely on the original memorisation map that uses parallel OPUS, and the target-based LL metric.

3.3.3 Manual annotation

In the previous subsections, we discussed coarse patterns that relate datapoints' features to memorisation metrics. To understand whether similar patterns appear when we qualitatively examine source-target pairs, we now annotate 250 En-Nl examples, uniformly sampled from different parts of the coordinate system, with lengths l for

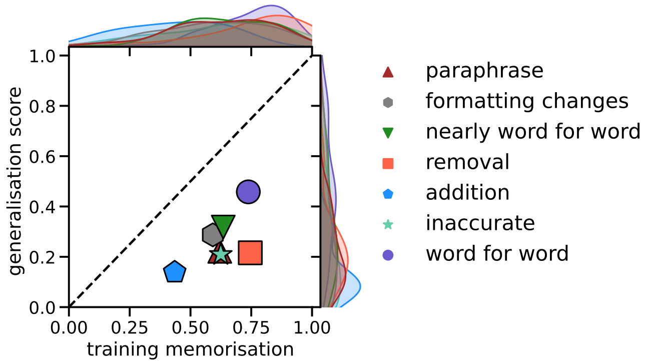


Figure 3.9: Centroids and marginal distributions of examples grouped through the manual annotation for En-Nl.

which $10 < l < 15$. We annotate them using the following labels, where multiple labels can apply to the same example:

- **Word for word**: if the target is almost a word-for-word translation of the source, with very minor rephrasing or change in word order, for instance, Example (1) below. After annotations were completed, we further subdivided these examples into **'word for word'** if no other labels were assigned, and **'nearly word for word'** if other labels were assigned.
- **Paraphrase**: if the target generally expresses the same meaning as the source but uses different wording, e.g. see Example (5) below.
- **Inaccurate**: if the target is an incorrect translation or discusses something that the source does not warrant, e.g. see Example (3) below.
- **Addition**: if the target introduces new information, e.g. see Example (8) below.
- **Removal**: if the target removes content from the source, e.g. see Example (6) below. When including removals or additions in the results, we only count those not labelled 'almost word for word', to focus on cases where the change affects the meaning.
- **Formatting changes**: if the target changes the punctuation or the capitalisation, such as in Example (9).

Figure 3.9 summarises the results. Firstly, these results consolidate the observation regarding source-target overlap: word-for-word translations, e.g. Examples (1) and (2), are positioned closer to the top right corner compared to inaccuracies, e.g. Examples (3) and (4), and paraphrases, e.g. Example (5).

(1) *s* En: In only days, without food or water, Society collapses into chaos.
t Nl: In slechts enkele dagen, zonder eten of drinken, stort de maatschappij in chaos. (TM=0.74, GS=0.47)

(2) *s* En: Leave a few empty rows and columns on either side of the values.
t Nl: Laat enkele rijen en kolommen leeg aan beide zijden van de waarden.

(TM=0.85, GS=0.55)

(3) *s* EN: The last 2 years of my life has been one big lie.
t NL: "De afgelopen twee jaren van mijn leven zijn een grote leven geweest. (*leven != lie*, TM=0.28, GS=0.14)

(4) *s* EN: We ask you to form a worldwide front against war and NATO.
t NL: Wij vragen u om een wereldwijd front tegen de oorlog en sancties te vormen. (*NATO != sancties*, TM=0.31, GS=0.19)

(5) *s* EN: I don't know how she did it, but she did it.
t NL: Geen idee hoe, maar ze deed 't. (*underlined portions are paraphrases*, TM=0.23, GS=0.14)

Yet, paraphrases and inaccurate translations have similar centroids on the map; the differences between those two types are subtle and are not well reflected in the memorisation metrics. Lastly, what is not that easily captured by one automated feature, but does show up in these results, is that targets that remove content from the source, e.g. Examples (6) and (7), are easier to memorise during training than those that add content, e.g. Example (8).

(6) *s* EN: Then we had our little adventure up in Alaska and things started to change.
t NL: Toen waren we in Alaska en begonnen dingen te veranderen. (TM=0.80, GS=0.22)

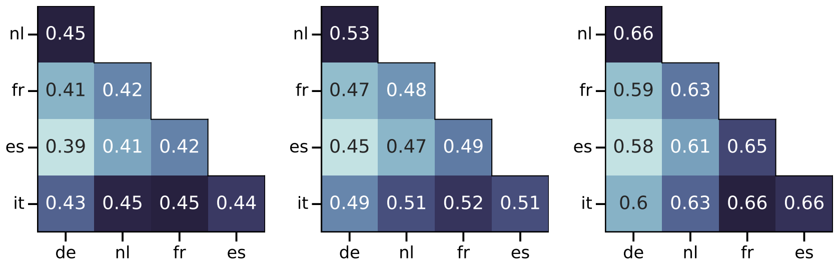
(7) *s* EN: He married his beloved wife, Penny, in 1977 and raised a family.
t In 1977 trouwde hij met Penny en samen brachten ze een gezin groot. (TM=0.50, GS=0.04))

(8) *s* EN: There are periods and stages in the collective life of humanity.
t NL: Evenzo zijn er perioden en fasen in het collectieve leven van de mensheid. (TM=0.45, GS=0.34)

(9) *s* It is difficult to negotiate with people who CONFUSE AUSTRIA WITH AUSTRALIA.
t Samenwerken met mensen die Oostenrijk verwarren met Australië is lastig. (TM=0.72, GS=0.01)

3.3.4 Comparing metrics for five languages

The trends of what complicates or eases memorisation are consistent for all five languages, upon which we elaborate in §3.5. Does this mean that one source sentence will have a very similar TM, GS or CM score across the five language pairs in our corpus? Not necessarily, as is shown in Figure 3.10, which, for the three metrics, reports the



(a) TM (b) GS (c) CM

Figure 3.10: Comparison of the memorisation metrics across the five languages, using Pearson's r.

correlation between scores associated with the same source sequence (but different target sequences) across the different languages. These correlations vary from moderate to strong, suggesting that for the same source sequence, scores tend to be similar, but that there are also a substantial number of examples for which memorisation scores differ.

A portion of the variation in memorisation scores over languages might be explained by language similarity: the genetic similarity of languages – quantified using the Uriel database (Littell et al., 2017) – positively correlates with the numbers reported in Figure 3.10, with a Pearson's r of 0.51, 0.68 and 0.78 for TM, GS and CM, respectively, with $p < 0.05$ for GS and CM. This suggests that more similar languages also have more similar changes in target translations.

Source sequences with different positions on the memorisation maps from two language pairs provide insight into how the relation between the source and target affects memorisation. Examples that move from the top right in one language to the bottom left in another show how targets go from easily learnable to not learnable: in Examples (10) and (11) the second target (t_2) seems misaligned. In Example (12) t_2 is contextually relevant, by discussing that something takes too long, but it is not a translation of "It's a long story".

(10) s En: She's not a child anymore.
t_1 Es: Ya no es una niña.
t_2 De: Du hast das Kind verwöhnt, Matthew. (*You spoiled the child, Matthew*)

(11) s En: It is an international obligation.
t_1 Es: Es una obligación internacional.
t_2 Fr: Nianias sur l'opportunité de cet embargo. (*Nianias on the advisability of this embargo*)

(12) s En: It's a long story.
t_1 Nl: Het is een lang verhaal.
t_2 It: Sarebbe troppo lungo spiegarsi. (*It takes too long to explain*)

What about examples that move from the top right to the bottom right, i.e. go from easily learnable to only learnable if they are in the training set? Generally, they seem to deviate from source sequences in more subtle ways – e.g. they are missing a term or phrase in t_2, as is the case in Example (13), where "Kenneth" is missing, Example (14) where "hunting game" translates into "game" and Example (15), where "viable suspect" is simply translated as "suspect".

(13) s EN: Kenneth, what are you doing here?
t_1 ES: Kenneth, ¿qué haces aquí?
t_2 FR: Que fais-tu ici (*What are you doing here?*)

(14) s EN: Is this a hunting game?
t_1 FR: C'est un jeu de chasse?
t_2 NL: Is dit een spelletje? (*Is this a game?*)

(15) s EN: We need a viable suspect.
t_1 ES: Necesitamos un sospechoso viable.
t_2 DE: Wir brauchen einen Verdächtigen. (*We need a suspect*)

3.4 Intermezzo: what about formulaic phrases?

Up to this point, we have examined datapoints and their features across all examples in the training dataset. Given the emphasis of Part II of this thesis on compositionality and non-compositional formulaic expressions, we now shift focus from considering *all* datapoints to a specific subset, analysed through the lens of non-compositionality. How do examples containing formulaic phrases score on our memorisation maps? Formulaic expressions, such as idioms, often convey figurative meanings that do not directly follow from their individual words and may be specific to a language or culture. As such, they require special treatment in translation, particularly when a literal rendering would be nonsensical in the target language (Baker et al., 1992; Bortfeld, 2003). To examine the memorisation scores of sequences containing formulaic expressions, we now outline the process used to select relevant stimuli, analyse how these examples differ for the three metrics under consideration, and conclude with a brief discussion.

3.4.1 Selecting formulaic and control stimuli

Firstly, we select examples that contain **compositional and non-compositional noun compounds** from a resource constructed by Tayyar Madabushi et al. (2021). Tayyar Madabushi et al. requested 12 annotators to collect example sentences of the compounds from web text, and curated an overview of 223 English compounds and their non-literal meanings based on these examples. We separate the compounds that have no literal meaning from those that have at least one non-literal meaning; for instance, "mailing list" is only used literally, whereas "elbow room" is both used literally (as a joint room) and figuratively, to represent space and freedom. Our parallel corpus has 1002 and 4154 matches for the non-compositional compounds and the compositional compounds, respectively. We automatically label instances for non-compositional compounds as figurative or literal using GPT-4o (Achiam et al., 2023),[2] using a straightforward prompt: "Consider the sentence `<sentence>`. Is the compound `<compound>` used figuratively or literally here? Answer using one word." Cognisant of the fact that this labelling might be inaccurate, we include both results for the subset of figurative non-compositional compounds and the full set of non-compositional compounds. 28% of the examples are labelled as figurative.

Secondly, we collect **idioms** from the Oxford Dictionary of English, specifically those that were included in the MAGPIE corpus of Haagsma et al. (2020). Over 600 are an exact match with source sequences from our parallel corpus, appearing in a total of 7078 examples. Similar to the non-compositional compounds, we automatically label the examples as figurative or literal using GPT-4o, using an analogous prompt but with 'compound' replaced with 'idiom'; 68% of the occurrences are labelled as figurative.

Lacking a standard resource for English **proverbs**, we, lastly, collect proverbial phrases from English Wikipedia,[3] postprocessing them to remove authors and potential variations, removing some prefixes such as "do not", "you cannot" and "there is/are" to improve recall. We remove multi-sentence instances, and remove "It is on" because it led to too many false positives for our parallel corpus. For 110 proverbs, we find 293 exact matches in the source sequences of our parallel corpus.

As these formulaic phrases are typically short and embedded in source sequences, we do not necessarily expect them to stand out in absolute terms – one idiom appearing in a longer sentence does not require memorisation for the full sequence but rather memorisation for a phrase contained within that sequence. We thus contrast the memorisation scores of these sequences to memorisation scores of **control stimuli** selected based on the source and target length of the white-space tokenised source and target sequences.

3.4.2 Results: formulaic sequences are memorised less

For our TM, GS and CM metrics, Figure 3.11 illustrates the difference between formulaic and control stimuli, aggregating over data from all five language pairs, including idioms and non-compositional compounds tagged as figurative by GPT-4o separately. Across

[2] Specifically, `gpt-4o-2024-08-06`, accessed on the 20th of February, 2025. Note that this experiment was added following the publication of this chapter, and that GPT-4o was not yet available at the time of publication.

[3] https://en.wikipedia.org/wiki/List_of_proverbial_phrases, retrieved on the 20th of February, 2025.

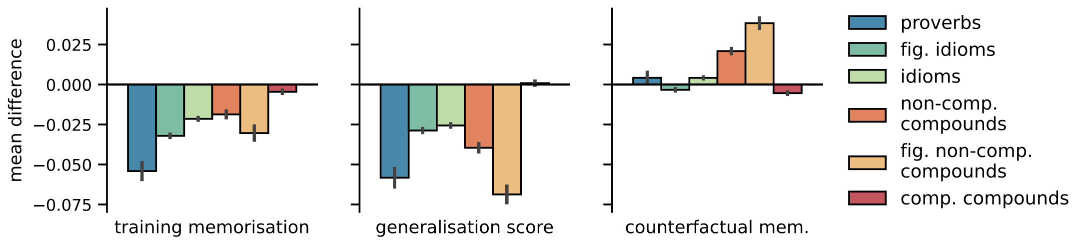
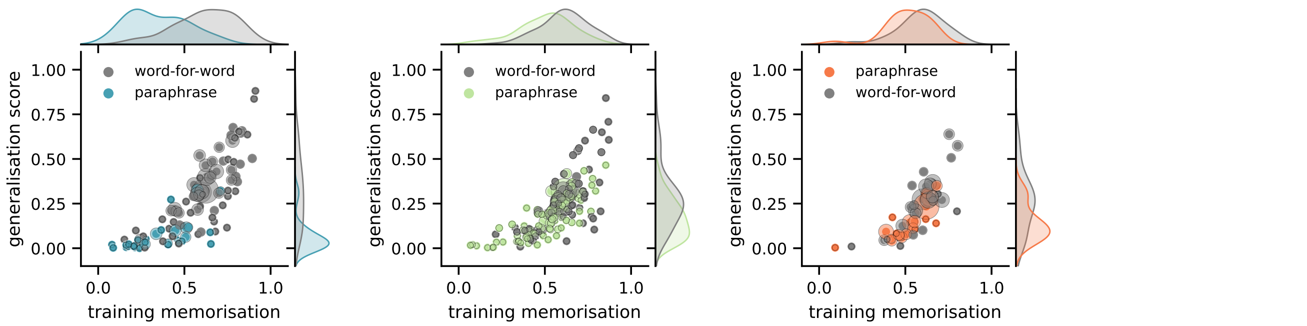


Figure 3.11: Differences in memorisation scores when comparing formulaic stimuli to control stimuli of the same source and target length. Error bars show the standard error.

the board, all formulaic sequences score lower in terms of both TM and GS, and out of all stimuli, the compositional compounds show the smallest difference to their control stimuli. If we compare the scores for our stimuli to their controls using an independent two-sided t-test (with a Bonferroni correction for comparing four types of stimuli to their controls over three metrics and five languages), that difference is nearly always significant for the formulaic stimuli for TM and GS (except for 4 out of 30 comparisons), but for none of the tests for compositional compounds. While selecting control stimuli based on sequence length is a crude approximation of a true control stimulus – ideally, one might correct for other factors, such as token frequency – the smaller difference for the compositional compounds consolidates that the control stimuli provide a meaningful point of comparison. Figurative idioms and figurative non-compositional compounds show a larger difference to the controls compared to just idioms and non-compositional compounds in general. Still, we interpret these results with caution since it is unknown how accurate the labelling is without manually evaluating GPT-4o on this task. The only stimuli that stand out for CM are the (figurative) non-compositional compounds, which, for all languages, have significantly different CM scores compared to control stimuli.[4]

On the effect of paraphrasing non-compositional phrases Why would sequences containing proverbs, idioms, and non-compositional compounds have lower TM and GS scores (or higher CM scores, in case of the non-compositional compounds)? One potential explanation is that due to the non-compositionality of these phrases, target translations contain more paraphrased material than the control stimuli; or, in terms of the numerical features we introduced in §3.3, there is less source-target overlap. We measure the median percentual change of surface-level features for formulaic stimuli

[4] Taking into account the hypothesis that non-compositional stimuli may have more accurate paraphrased translations than compositional examples (see §2.3.3), one might worry that the LL differences compared to controls stem from underestimating the TM and GS scores of formulaic stimuli. LL is based on target tokens' log-probability, but models may spread probability mass across several valid paraphrases. In section A.3, using En-Nl data, we confirm that formulaic stimuli also show lower TM and GS compared to controls when we rely on a neural quality estimator, which is expected to be more semantics-aware. In that scenario, they do show slightly higher CM than controls.



(a) Proverbs (b) Idioms (c) Non-comp. compounds

Figure 3.12: Visualisation of formulaic phrases based on the mean TM and GS scores for examples in which they appear. Dots are scaled based on the phrase's frequency, and the colour indicates whether the phrase of interest is usually translated word for word or is paraphrased.

compared to control stimuli across data from the five language pairs. The top three features showing the largest differences per phrase type are as follows:

- *For proverbs*: unaligned tokens (for both the source and target), and the back-translation Levenshtein ED;
- *For idioms*: unaligned tokens (source), word overlap and the backtranslation Levenshtein ED;
- *For non-compositional compounds*: unaligned tokens (source), the segmentation ratio of the target and the source-target length difference after tokenisation.

Most of these features, indeed, express that decreased source-target overlap we would expect when seeing more paraphrases in the target. The segmentation ratio and length difference could explain why non-compositional compounds are the only examples with increased CM: they appear in sequences with tokens that are more often rare and thus get tokenised into more tokens, which we previously identified as being positively correlated with CM (§3.3).

We can further consolidate the role paraphrasing plays by taking a closer look at our data: for En-Nl, I manually labelled translations for the proverbs, 200 of the idioms, and all non-compositional compounds. The translations were labelled as 'word for word' or 'paraphrase', based on how the majority of up to 5 figurative examples of that phrase in our corpus are translated.[5] Even though proverbs, idioms and non-compositional compounds are generally non-compositional phrases, word-for-word translations can still occur, for instance, if the equivalent phrase exists in the target language (e.g. "biological clock" exists both in English and in Dutch, whereas "sitting duck" requires a

[5] Excluding examples for which less than 66% has a consistent label, examples that primarily appear to be false positives in context, such as the idiom "on the level", or examples that copy the English phrase in the Dutch target, such as "brain drain".

paraphrase in Dutch). We previously identified that formulaic stimuli have lower TM and GS scores than control stimuli; Figure 3.12 now demonstrates that even within the formulaic stimuli, the subgroup that has the lowest TM and GS scores are those labelled as paraphrases.[6] This effect is by far the strongest for the proverbs, which is likely because they tend to be longer phrases than idioms or noun compounds, thus having a more significant effect on the overall memorisation scores of the source-target pairs in which they appear.

It should be noted that, particularly for proverbs, not all word-for-word target translations are correct translations, e.g. in case of (1) "tomorrow is another day" translated as "het morgen is een andere dag", (2) "too little too late" translated as "veel te weinig te laat" and (3) "to the victor go the spoils" translated as "de winnaar krijgt de verwenningen". Not only are they semantically very odd, but also (for 1 and 2) grammatically incorrect. This could result from the fact that OPUS not only contains human translations but also auto-aligned pairs from the web, which is how machine-translated text could have ended up in the training corpus we use.

3.4.3 Discussion

Translating proverbs, idioms, and non-compositional compounds accurately requires memorisation: no human automatically knows that "What goes around comes around" should map to "Wie de bal kaatst kan hem terug verwachten" ("*Who passes the ball can expect it back*") when translated into Dutch. This is information that simply must be committed to memory, for both humans and computational models alike. One would thus hope to have higher memorisation scores for these types of phrases, but, in practice, formulaic stimuli score lower in terms of both TM and GS compared to control stimuli. The experiments conducted here are exploratory and preliminary – to consolidate these findings, the experiments would need to be repeated with multiple external annotators. Nonetheless, given the consistency across the different formulaic stimuli and the comparison to compositional compounds, we can safely state that formulaic stimuli are memorised less and that paraphrasing plays a role in this. This is unsurprising, given that phrases with reduced source-target overlap scored lower in terms of TM and GS across the board, but it does highlight a paradox that we will later return to in chapters 5 and 6: what NMT models actually memorise does not necessarily equal what we think they should memorise.

3.5 Approximating memorisation measures

Following our intermezzo about formulaic phrases, we now return to focusing on all of our datapoints. Having examined correlations between datapoints' features and

[6]Inspect the results per formulaic phrase interactively using our demo.

memorisation values, we now go one step further and treat this as a regression problem: given the characteristics of a datapoint, can we predict the TM, GS and CM metrics? We include the previously mentioned features and additional ones obtained from an NMT system during training. We examine the performance of our regression models and explore how well they generalise across language pairs. The analysis aids in consolidating findings from §3.3 and improves our understanding of how language-independent our findings are. Since computing CM is resource-intensive, the predictors can also serve as memorisation approximators (we circle back to this in §3.6.2).

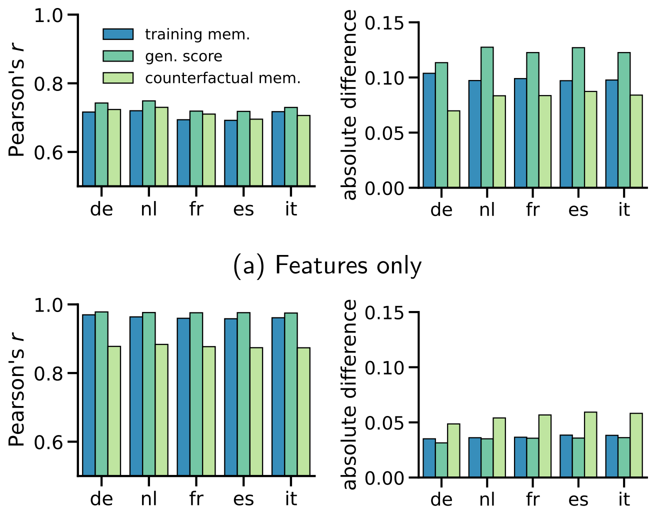
Experimental setup To extract training signals, we train one `transformer-base` model per language pair on the full dataset for 50 epochs, acting as our *diagnostic run* from which we extract the following signals:

- **Confidence** and **variability**: the mean and standard deviation of the target likelihood computed over all epochs (we follow Swayamdipta et al., 2020, in the choice of metric, but with probability replaced with likelihood);
- **Final train likelihood**: the likelihood of the target in the final training epoch;
- **Forgetting**: the sum of all decreases in target likelihood observed for consecutive epochs (adapted metric from Toneva et al., 2019);
- **Hypotheses' likelihood** obtained in the final epoch. Uncertainty can aid in detecting out-of-domain data (D'souza et al., 2021), and hallucinations (Guerreiro et al., 2023);
- We also included **final train likelihood - confidence** since initial experiments suggested those two correlated most strongly with TM and GS, and CM is known to be a combination of those two signals.

Apart from the hypotheses' likelihood, these signals are ones you would naturally obtain while training a model. We train shallow MLPs (with two hidden layers of 100 dimensions, trained as detailed in Appendix A) to predict the TM, GS and CM metrics. We train one MLP on the datapoints' features from §3.3, and one on the features and the training signals, and report their performance using Pearson's correlation and the absolute difference between predictions and memorisation scores. We train the MLP using the En-De memorisation map and apply it to data from the other language pairs.

Results If we first look at the MLP trained on the surface-level features only, the predictions already strongly positively correlate with the memorisation metrics, with Pearson's r around 0.7 and a mean absolute difference around 0.1, see Figure 3.13a. Combining the features and training signals further boosts performance (see Figure 3.13b).

Since we applied the En-De MLPs to the other languages, these figures illustrate that an MLP trained on one language is transferable to models trained with other target languages. However, an important caveat to mention is the fact that the language pairs



(a) Features only

(b) Features and training signals

Figure 3.13: Predicting memorisation using an MLP, based on examples' features and models' training signals. The MLP is trained on En-De and applied to all languages.

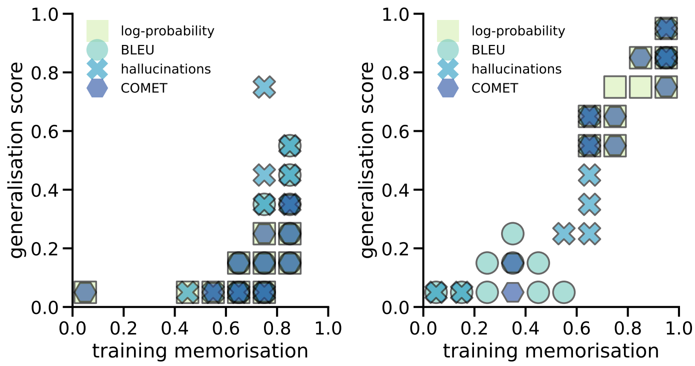
we examined here are all Indo-European and are, therefore, quite strongly related. The transferability of the MLP predictors might thus be limited to related languages. We further comment on the set of languages used in the limitations section, see §3.7.1. Note that this also does not mean that models for the different language pairs behave similarly for the same source sentences, but instead that models trained on different language pairs behave similarly for source-target pairs with the same features. In practice, this means that we can make an educated guess about the amount of memorisation required for a new datapoint using predictors trained on a subset of the data, or using predictors trained on data from a related language pair.

3.6 Memorisation and performance

Finally, we examine the relation that different regions of the memorisation map have to models' performance. In §3.6.1, we focus on the influence of data subsets from our original 1M training corpus, and in §3.6.2, we turn to sampling specialised training corpora from a larger dataset of 30M examples. The previous sections showed that results across language pairs are highly comparable. Given the computational expense of our experiments, we will now focus on En-Nl data only.

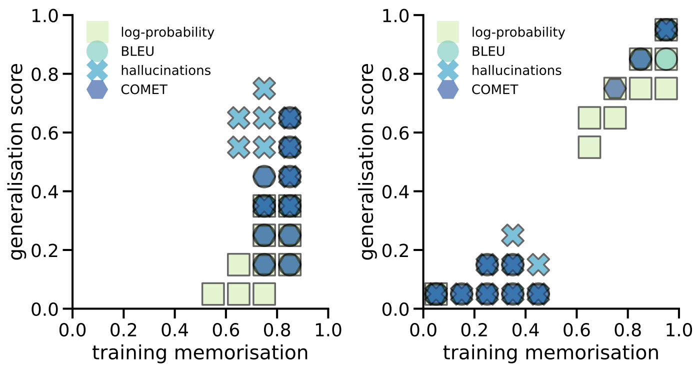
3.6.1 Importance of different regions

How do examples from specific regions of the memorisation map influence NMT models trained on that data? We now investigate this by creating subsets of examples based on their coordinate on the memorisation map, and either withholding them from training,



(a) Most relevant (b) Least relevant

Figure 3.14: The ten most and the ten least relevant regions on the memorisation map per performance metric, based on training models while *withholding* data from these regions. The most relevant regions show the largest change in performance when withheld.



(a) Most relevant (b) Least relevant

Figure 3.15: The ten most and the ten least relevant regions on the memorisation map per performance metric, based on training models *only* on data from these regions. The most relevant regions have the best performance.

or training exclusively on those subsets.

Experimental setup For 55 coordinates (i, j), where $i, j \in \{.1, .2, \ldots, 1\}$, $j \leq i$, we create data subsets including the nearest examples for that coordinate, up to 750k source tokens. Depending on the number of examples surrounding a coordinate, the datapoints can lie closer or further away before reaching the limit. For each subset, we train models with three seeds in two setups: one in which the training set has that subset withheld, and one in which the training set consists exclusively of that subset.

We evaluate models according to four performance metrics: (i) **BLEU** scores for the FLORES-200 development set (Goyal et al., 2022). (ii) the mean **log-probability** of a target, averaged over datapoints from the FLORES-200 data. (iii) a **hallucination tendency** computed using the approach of Lee et al. (2018), which involves the insertion

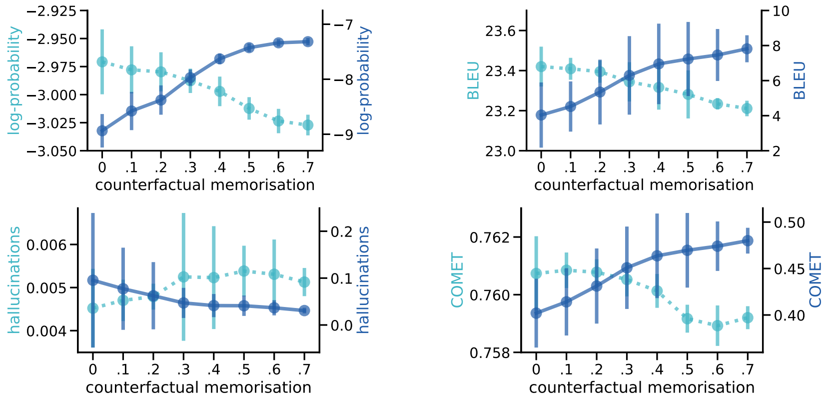


Figure 3.16: Performance change along the CM dimension, when withholding subsets with a certain CM score (light blue dashed lines, left-aligned y-axes) or training on subsets with that CM score (dark blue solid lines, right-aligned y-axes). Error bars indicate standard deviations over regions with the same CM score.

of a token into a source sentence and repeating that for more than 300 tokens (high-frequency, mid-frequency and low-frequency tokens and punctuation marks), and four token positions. A hallucination is recorded if BLEU drops below 1 after an insertion. We apply this to 1000 examples (500 from FLORES, 500 from our parallel OPUS) and, following Lee et al., measure the ratio of source sequences which can be perturbed to at least one hallucination. (iv) **COMET-22**, an ensemble-based neural evaluation metric (Rei et al., 2022), also computed over the FLORES-200 data.

Results To express an example's impact in the withholding setup, we average the performance of all models for which that example *was not* in the training set. The more negatively the performance is affected, the more important an example is. We aggregate over regions of examples and exclude regions that represent <2k datapoints. For the exclusive setup, we do the opposite, aggregating performance for a region using all models that *did* have examples in that region in the training set. We then compute the ten most relevant regions and the ten least relevant ones. For the withholding setup, *most* relevant means that the BLEU score, log-probability or COMET score decreases the most or that the hallucination tendency increases the most if you withhold datapoints from this region. For the exclusive setup, *most* relevant simply means that that region scored the highest (for BLEU, log-probability and COMET) or lowest (for the hallucination tendency).

Figures 3.14 and 3.15 visualise the most and least relevant regions, for the withholding and exclusive setups, respectively. Figure 3.16 directly visualises the relation between the CM scores and the performance metrics. In general, the figures suggest that examples with a higher CM value are more beneficial, and examples closest to the diagonal are the least relevant. This is in accordance with related work from CV (Feldman and

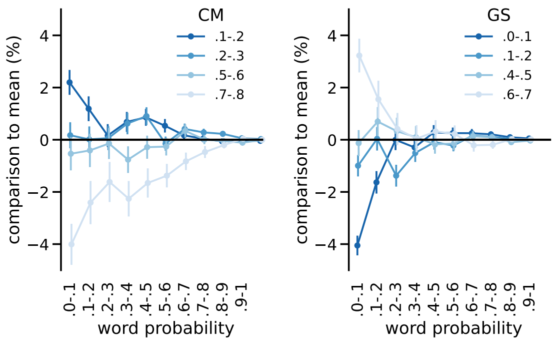


Figure 3.17: Change in word probability (y-axis) when removing coordinates based on CM and GS, per word probability bucket (x-axis). Error bars indicate standard error.

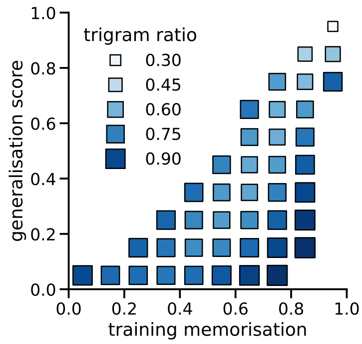


Figure 3.18: Ratio of unique trigrams per coordinate of the memorisation map.

Zhang, 2020) and NLP classification tasks (Zheng and Jiang, 2022), where examples with high CM values had a larger (positive) contribution to the accuracy for test data than random examples. There are, however, substantial differences between the different performance metrics regarding the usefulness of the lower CM regions, and the standard deviations in Figure 3.16 are quite high, particularly for the BLEU and COMET scores in the exclusive setup. This is due to the fact that a CM value of, for instance, 0.1, represents regions with widely varying TM and GS scores, that have very different types of datapoints associated with them (as we established in §3.3). Examples in the bottom left of the memorisation map are the least beneficial when maximising BLEU scores, but examples from the top right are the least beneficial when maximising the log-probability of evaluation data, even though both the bottom left and top right would have a CM score close to zero. This result underscores that memorisation should not be treated as a one-dimensional phenomenon. The results for COMET are somewhat in between those of BLEU and log-probability, but they nonetheless reflect the overall pattern of high-CM examples being more relevant than low-CM examples.

Going further: why high CM might be beneficial Why might examples with high CM be more beneficial for generalisation performance? As laid out in §2.2.2, Feldman and Zhang (2020) observe that training examples with high CM usually are atypical 'long-tail' examples and that they improve performance on visually similar test examples. Analogous processes might be at play for translation. Yet, there may be benefits to examples with high CM values even without similar test examples.

Firstly, we examine the log-probability performance impact of the withholding setup more closely. Why are the bottom rows, and the examples with high CM in particular, most relevant? This metric is computed using the *target* tokens' probabilities, which are easily negatively affected if there are some unexpected target tokens. Coordinates in the bottom rows might be relevant because they include somewhat 'noisy' data, which increases uncertainty in the model during training and thus smooths the output probability distribution. To examine whether our data reflects that, we put tokens from the FLORES sequences in buckets based on the mean token probability that they have in the predictions of all models trained in §3.6.1. We compare these token probabilities to those from models that leave out examples with a certain CM or a certain GS in Figure 3.17. If examples with high CM are removed during model training (e.g. examples from the 0.7-0.8 CM band), the token probabilities for buckets with a relatively low probability show the largest *decrease*. Vice versa, when removing examples with low CM, the token probabilities for buckets with the lowest probability *increase* the most. A similar effect is observed for GS, where removing low GS examples decreases the probabilities of low-probability buckets the most. This suggests that removing examples with high CM (and low GS) makes the output distribution less smooth.

Secondly, we would like to point out that examples with a high CM score generally have less redundancy than examples with a low CM score (in particular, compared to examples with a high TM *and* GS). Our training corpus has 1M *unique* source sentences. Although none of them are repeated, some sentences are more alike than others in terms of n-gram count, explaining that redundancy. To illustrate that, Figure 3.18 conveys the ratio of unique trigrams in the data from a particular coordinate.

Summarising, these preliminary investigations provide two reasons for why high CM examples could be beneficial: there is less redundancy among them (evidenced by more unique n-grams) which could make them more informative training data, and removing examples with a high CM score negatively affects models' predictions for low-probability tokens, in particular; therefore, including them as training material may quite literally reserve probability mass for the long tail of the output distribution.

3.6.2 Specialising NMT systems using memorisation metrics

We have now examined the relation between models' performance and the different regions of the memorisation map, but all within our original 1M EN-NL datapoints. To understand whether our findings extrapolate to a larger dataset, we perform a *proof-of-concept* study to show that we can put the lessons learnt to use with new data: memorisation metrics can be predicted using datapoints' features and distinct regions of the map have different roles. We now use these lessons for targeted model training.

Experimental setup We again train NMT systems in a low-resource setup, yet, different from the previous sections, we now select examples from a larger set of OPUS examples for EN-NL (30M examples) based on their memorisation score as *predicted* using the features-only MLP from §3.5. We first sample 1M random examples, and then sample one dataset with high CM examples. Because for log-probability both the withholding and exclusive setup suggested it is not just high CM, but also low GS examples that are

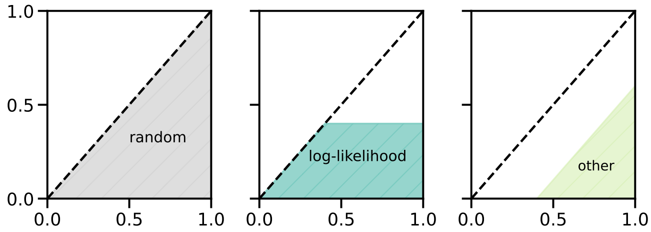


Figure 3.19: Areas from which we select examples when specialising for a certain metric.

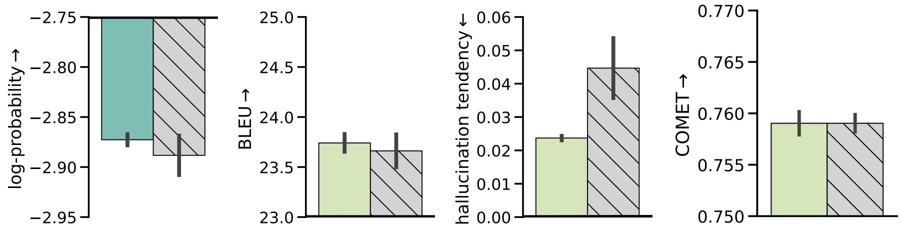


Figure 3.20: Results of comparing specialised models (in colour) to models trained on randomly selected OPUS data (in gray, with hatches). Error bars show the standard error.

beneficial, we sample a separate 1M dataset for log-probability. We mark the regions on the memorisation map in Figure 3.19. Examples are randomly sampled from those areas until they match the random dataset in the number of tokens. For those three datasets, we train three model seeds.

Results We compare the specialised models to a model trained on 1M random examples (Figure 3.20) and observe that the specialised models perform on par or show a slight improvement, with the largest relative improvement observed for the hallucination tendency.

At the same time, Raunak et al. (2021) reported that when trying to elicit hallucinations from the model using its training examples, examples with high CM scores lead to more hallucinations. To determine whether our results also reflect that, Figure 3.21 displays the mean CM scores associated with each hallucination from the current and previous subsection. For the models from the withholding setup of §3.6.1 (trained on these examples), but not §3.6.2 (not trained on these examples), hallucinations are indeed more associated with examples with higher CM scores. Together, these results indicate that at the instance level, examples with higher CM scores are more likely to turn into hallucinations themselves when artificially perturbed, but that they are useful training material at the system level, nonetheless. Potentially, high CM examples are better training material because they show some natural variation, making the model more robust to those artificial perturbations.

All in all, this small-scale experiment provides a *proof-of-concept*: even when using heuristics (i.e. applying the MLP to new datapoints) we can start to use memorisation

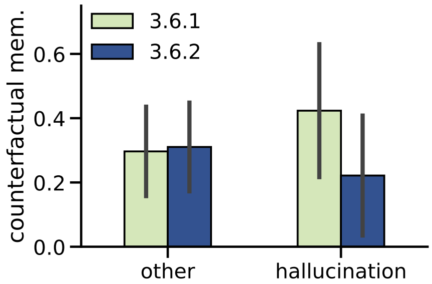


Figure 3.21: Using all hallucinations from §3.6.1 (withholding setup) and §3.6.2, we trace the mean CM score of the unperturbed sequence. Error bars report standard deviation.

metrics in a deliberate way when training NMT systems. However, the hallucination results underscore that the relation between examples with high CM scores and model performance is not straightforward: examples that are most beneficial for systems' quality can introduce vulnerabilities at the same time. The performance changes observed, however, are very minor, likely due to the application of an imperfect MLP predictor, and future work would further have to invest in efficient and effective predictive models of memorisation scores when integrating them as data selectors during training.

3.7 Conclusion and discussion

Learning the input-output mapping that is represented by NMT training data involves so much more than simply learning a function that translates words from one language into another and rearranges words. It requires understanding which words form a phrase and should be translated together, which words from the source should be ignored, which words can be copied from source to target, and in which contexts "eggs in a basket" are no typical eggs and require a paraphrase in the target language. NMT systems *need* memorisation of patterns that are out of the ordinary.

There are, however, many open questions regarding what memorisation is, when it is desirable and how to measure it. In this chapter, we took a step towards answering those by creating a map of the memorisation landscape for 5M datapoints. We used graded metrics based on CM to position each example on the memorisation map. We identified salient features for each of the metrics to characterise what memorised examples are like (§3.3), demonstrated that sequences containing formulaic sequences score lower in terms of TM (§3.4), illustrated that we can approximate memorisation metrics using surface-level features (§3.5) and drew connections between models' performance and regions of the memorisation map (§3.6). We identified that findings from other tasks and domains about CM, as laid out in §2.2.2, transfer to NMT: CM highlights examples that contribute most to models' performance.

Furthermore, our results illustrate that memorisation is not one-dimensional: CM assigns similar scores to paraphrases and slightly inaccurate translations, examples with high CM scores can be beneficial and introduce vulnerabilities at the same time, and there are nuances about which region of the map is most beneficial depending on the performance metric used. We recommend caution when discussing different phenomena under the umbrella term of 'memorisation'. Authors should be very specific about the type of memorisation their results apply to, and ideally explore the many dimensions of memorisation, for instance, by developing memorisation maps across tasks to understand how the memorisation–generalisation continuum changes accordingly.

3.7.1 Limitations

We identify three main limitations of our work. Firstly, the experimental setup used is rather **computationally expensive** due to the repeated model training as explained in §3.2. Because of this, we opted for a much, much smaller dataset than state-of-the-art NMT systems would use (OPUS can contain hundreds of millions of examples for a high-resource language pair), but it still limits the applicability of the methodology to other tasks and for other researchers.

Secondly, we did not investigate the impact of major changes to the **experimental setup**, such as using a different model or model size or using a different or a larger dataset. Even though our findings are expected to extend beyond our specific experimental setup, the precise memorisation scores we obtained are specific to our setup – e.g. a larger system is likely to memorise more, and systems trained for much much longer are likely to see increased memorisation. We did, however, investigate two variations of our experimental setup in §3.3.2 by varying the metric underlying our memorisation map and varying the training data selection criteria.

The final limitation concerns the **language pairs used**. We relied on parallel data in order to rule out that differences between language pairs were due to dataset differences, but this prevented us from including low-resource languages, for which parallel data is unavailable at a large scale. In preliminary experiments, we also experimented with Afrikaans (together with German and Dutch), and many of the qualitative patterns observed were similar. The fact that the languages are from the same language family and geographical region also limits the generality of our results. The five language pairs had a very similar relation between surface-level features and memorisation scores. Yet, this would likely have been different for target languages from other families. For instance, our overlap features are only meaningful in the case of a shared script and a partially shared vocabulary, and the length-based features are likely to have a different relation to memorisation when comparing analytic vs agglutinative languages.

3.7.2 Retrospective and outlook

NLP is an exceptionally fast-paced field, and in the time that passed between the start of the work that led to the publication discussed in this chapter in the summer of 2022 (followed by publication in the autumn of 2023), and the time of writing this retrospective in 2025, the task of NMT has changed massively. NMT was initially not affected by the trend of fine-tuning or prompting LLMs becoming the de facto standard for other NLP tasks, but this changed around the time the paper was published. This becomes evident even by simply considering the findings reported yearly by the conference on machine translation (WMT). Whereas LLMs were not even mentioned by Kocmi et al. (2022), in 2023, the organisers succinctly summarised LLMs' contributions as "LLMs are here but not quite there yet" (Kocmi et al., 2023), followed by "the LLM era is here but MT is not solved yet" (Kocmi et al., 2024). Looking back, a fourth limitation of the work conducted is thus that it **trains models from scratch**, and cannot provide reliable estimations of how memorisation would differ during LLM fine-tuning on NMT data, or whether our findings hold for source-target pairs an LLM may have memorised in its pretraining stages.

Nonetheless, we consider our work to be a valuable contribution to the field. It was only the fourth article to consider CM in the context of NLP (following Zheng and Jiang, 2022; Zhang et al., 2023; Raunak et al., 2021), and the second to examine CM on the scale of millions of textual examples (following Zhang et al., 2023). Of related work that appeared following our publication, work by Prashanth et al. (2024) is most strongly related, who examine verbatim memorised examples of `Pythia` models (Biderman et al., 2023). They propose a taxonomy and report corpus-wide and datum-level features that relate to memorisation, distinguishing between recitation (quotes committed to memory through duplication), reconstruction (passages that can be partially produced by filling in gaps in a more widely used template), and recollection (reproduction of sequences where that reproduction is not explained by duplication or data templates). The recollected sequences are the most similar to the high CM examples we discussed, and Prashanth et al. identify that, similar to our results, the presence of rare tokens is strongly associated with this category.

Other recent related work by Lesci et al. (2024) proposes CM estimators, not at the instance level, but at the level of batches of data to provide so-called memorisation profiles per LLM. For `Pythia` models, they demonstrate that memorisation is stronger and more persistent in larger models, that the data presentation order and learning rate influence what is memorised and what is forgotten, and that memorisation in larger models can be predicted from the profiles of smaller models.

We aimed to provide a nuanced discussion of memorisation, emphasising that it can be beneficial to model performance while introducing vulnerabilities (such as hallucinations for training examples) at the same time and that models do not always memorise what we would want them to memorise (as is the case for sequences containing formulaic phrases). In Dankers and Raunak (2025), my co-author and I further dive into memorisation vulnerabilities in NMT.[7] We show that when training NMT models using knowledge distillation to create compressed models with competitive performance, this not only leads to better smaller models, but also to models that demonstrate more memorisation than models of the same size trained from scratch. We focus on verbatim memorisation and extractive memorisation – where models memorise to output the target after seeing only a prefix of the source – and also identify increased hallucinations as a consequence of knowledge distillation.

Going forward, we hope that our work can serve as a point of comparison when evaluating qualitative patterns of what underlies memorisation in other tasks and experimental setups (akin to the work of Prashanth et al., 2024), and that our per-datum memorisation metrics can be a beneficial resource. CM and GS are simply impossible to compute when working with datasets containing thousands or millions of datapoints due to the leave-one-out principle on which the definitions rely. Given the number of models used in our approximation and the statistics we provided suggesting that our approximations are reliable, future work developing proxy metrics for GS and CM (akin to the work of Lesci et al., 2024) could thus use our data as a resource for benchmarking those metrics before applying them to new datasets or models.

[7] I was the first author of this article, but it is not a part of this thesis.

Chapter 4

Layer-based memorisation localisation

4.1 Introduction

Whereas the previous chapter highlighted that, to some extent, all datapoints are memorised and that data should be viewed along a memorisation–generalisation continuum, we now shift focus. Instead of taking this broad perspective, we examine specific examples that are known to require substantial memorisation from our models to investigate *memorisation localisation* – i.e. identifying which weights, subcomponents, or layers are most strongly associated with storing particular information. Specifically, we explore where memorised information is stored within the many layers of a transformer.

Whether we can localise memorised information and edit models' memories has been widely studied, but there is little consensus in the literature on which layers play a key role in memorisation within deep neural models. Related work on image classification from CV mostly focused on memorisation of perfectly memorised mislabelled examples, positing that lower layers capture generalisable features while deeper layers memorise (Morcos et al., 2018; Cohen et al., 2018; Ansuini et al., 2019; Baldock et al., 2021; Stephenson et al., 2021, i.a.).[1] We dub this the **generalisation-first, memorisation-second hypothesis**. Related NLP studies primarily discuss memorisation of facts: methods that examine so-called 'knowledge neurons' mostly point towards the top layers of transformer-based LMs (Dai et al., 2022; Zhao et al., 2024a; Chen et al., 2024), whereas methods applying causal tracing and model editing primarily point to early and middle layers (De Cao et al., 2021; Meng et al., 2022, 2023). Methods that project hidden representations into the vocabulary space or zero-out internal updates to the hidden representations suggest the lowest layers are most important for fact and idiom recall (Haviv et al., 2023; Geva et al., 2023). A final strand of related work in NLP

[1] Although this has been challenged by Maini et al. (2023).

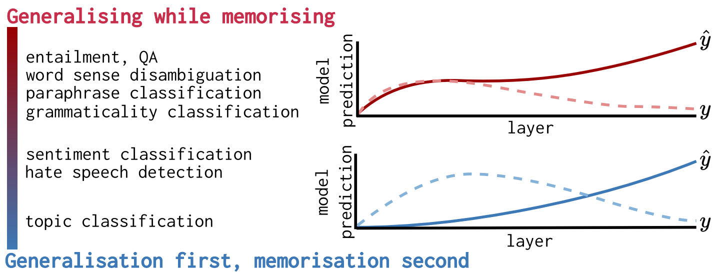


Figure 4.1: If we train transformer to memorise incorrect label $\hat{y}$, the implementation of that memorisation is task-dependent. We demonstrate this for twelve NLP classification tasks. The visualisation is for illustrative purposes.

studies localisation for verbatim memorisation. Stoehr et al. (2024) and Chang et al. (2024) suggest that storing sequences verbatim happens in a distributed manner over many layers, in which lower layers play a more critical role than higher layers.

We extensively reviewed these lines of related work in §2.2.3. Summarising, memorisation localisation studies have drawn a wide range of conclusions. It remains unclear whether the differing conclusions can be attributed to a distinction between the vision and language modalities, to a difference between the types of memorisation investigated, to localisation techniques employed, to the different metrics used to evaluate the localisation or even to the various models under investigation. In this chapter, we contribute a crucial piece of the puzzle in the memorisation localisation landscape, by, akin to work from CV, studying memorisation of mislabelled examples ('noise memorisation'), using models for NLP classification tasks by learning a classification head and fine-tuning the many layers of the models. This allows us to determine whether the 'deeper layers' answer from CV truly contrasts with the 'lower layers' answer from the majority of NLP studies, or whether that was simply unique to noise memorisation. Additionally, it allows us to explore whether the 'lower layers' answer is specific to fact memorisation and verbatim memorisation.

The experiments in this chapter contribute to the thesis's overarching research question RQ2: "*Which model-internal mechanisms enable memorisation?*" In studying layer-based localisation, we answer the following two sub-questions:

1. *Can memorisation of mislabelled examples be localised to individual layers?* We use four memorisation localisation methods (§4.2), and first examine their accuracy in a control setup. Afterwards (§4.3), we address this question, identifying that memorisation is not implemented in individual layers but that multiple layers gradually shift mislabelled examples to their newly assigned class. In §4.4, we introduce the visualisation technique of **centroid analysis** to make this story more interpretable, visualising how the hidden representations change from layer to layer, and how that aligns with the layer-based results obtained.
2. *To what extent is layer-based localisation consistent across LMs and classification*

tasks? We apply our localisation methods to twelve NLP classification tasks and four LMs. The four models mostly yield consistent results. We identify subtle differences between tasks that we relate to models' generalisation performance on unseen data. The better a model generalises to new data for a particular task, the more relevant deeper layers are for memorisation. Figure 4.1 illustrates this. In §4.5, we consider changing the model size or modifying the datasets to further study how this influences the results.

Overall, our findings do not align with the generalisation-first, memorisation-second hypothesis, but support a more nuanced version of the hypothesis. We elaborate on this in our discussion (§4.6), in which we also reflect on what our findings mean for localisation and model editing going forward, what the limitations of our work are and how our work aligns with literature that appeared after the publication of the chapter.

4.2 Methods and experimental setup

To gain a good understanding of how memorisation is task- and model-dependent, and to what extent our results are specific to a localisation method, we use twelve datasets, four LMs and four localisation methods. Here, we detail the tasks, datasets and models used, and elaborate on the localisation techniques and their accuracy when evaluated in a control setup, before moving on to the full localisation results in the next section.

4.2.1 Tasks and datasets

We combine datasets from the classification benchmarks GLUE (Wang et al., 2019b) and SuperGLUE (Wang et al., 2019a), which mostly contain binary classification tasks, with datasets from more diverse domains and with larger label set sizes. Table 4.1 provides an overview of the task, domain, training set size and number of labels per dataset.[2] For each dataset, we perturb the labels of 15% of the training examples ('noisy' examples, $x \in \mathcal{X}_n$, $y \in \mathcal{Y}_n$), with the new label randomly drawn from all labels but the original one. The remaining 85% is unperturbed ('clean' examples, $x \in \mathcal{X}_c$, $y \in \mathcal{Y}_c$). The tasks generally fall into four categories. The first five datasets capture varying aspects of generic **natural language understanding**,[3] and were previously included in either the GLUE or the SuperGLUE benchmarks:

1. `CoLA`: this dataset includes sentences from books and journals on linguistic theory, and requires indicating whether a sentence is grammatical (Warstadt et al., 2019);

[2] Appendix B provides URLs for the various datasets and elaborates on the licenses associated with the datasets.

[3] Note that the subdivision into the four groups of tasks is a way to organise our experimental results rather than a prescriptive categorisation: we are not necessarily stating that sentiment, hate speech or topic classification are not NLU tasks.

Category	Dataset	Task	Domain	Size	Labels
NLU	WiC●	word sense disambiguation	Word/Verbnet, Wiktionary	5.4k	2
	RTE✖	textual entailment	news, Wikipedia	2.5k	2
	MRPC■	paraphrase classification	news	3.7k	2
	CoLA✚	labelling grammaticality	linguistic theory books	8.5k	2
	BoolQ◇	question answering	Google queries, Wikipedia	9.4k	2
Sentiment	SST-2◆	sentiment classification	movie reviews	6.9k	2
	SST-5△	sentiment classification	movie reviews	8.5k	5
	Emotion⌑	emotion classification	tweets	16k	6
Hate speech	ImplicitHate▽	hate speech classification	tweets	5.1k	7
	Stormf.✱	hate speech classification	social media	8.6k	2
Topic Classification	Reuters●	topic classification	news	5k	8
	TREC★	topic classification	news, misc.	5.5k	6

Table 4.1: Datasets with their domain, label set size and training set size. Going forward, datasets are marked consistently in figures using the same colours and symbols.

2. MRPC: this dataset requires indicating whether two sentences are each other's paraphrase for examples from the Microsoft Research Paraphrase Corpus (Dolan and Brockett, 2005);
3. RTE: this is a small-scale dataset for the task of recognising textual entailment, including an 'entailment' and a 'non-entailment' class, containing data from various annual textual entailment challenges gathered by Wang et al. (2019b);
4. BoolQ: this binary classification task from Clark et al. (2019a) requires, given a Wikipedia passage and a Google query, a yes or no answer to the question;
5. WiC: this dataset contains binary labels indicating whether the same sense of a word is used in two different sentences (Pilehvar and Camacho-Collados, 2019).

Next, we include two datasets for the task of **hate speech classification**:

6. ImplicitHate: this dataset includes tweets that are assigned one of seven labels indicating the type of hate speech present (ElSherief et al., 2021). The different classes include examples of grievance, incitement, inferiority, irony, stereotypes and threats. All remaining examples are labelled as 'other';
7. Stormfront: this dataset includes snippets from a white supremacist forum, with labels indicating whether or not they contain hate speech (de Gibert et al., 2018).

We also include three datasets for **sentiment-related tasks**:

8. SST-2: sentiment classification using two classes (positive and negative) for the *Stanford Sentiment Treebank* (SST) (Socher et al., 2013). Instead of using the GLUE-based version of SST – that includes full sentences, but also sub-sentences – we only include full sentences, to make SST-2 more comparable to SST-5;
9. SST-5: sentiment classification using five classes (very negative, negative, neutral,

positive and very positive) from SST (Socher et al., 2013);

10. `Emotion`: tweets that are labelled using one of the emotions of anger, fear, joy, love, sadness, and surprise. We use a publicly available 20k split of the dataset, which is a subset of the data originally published by Saravia et al. (2018).

Finally, we use two datasets for the task of **topic classification**:

11. `Reuters`: we use the ModApte split of the Reuters text classification dataset (Apte et al., 1994), and restrict the examples to those that are assigned one topic, using labels that appear at least 100 times ('earn', 'acq', 'crude', 'trade', 'money-fx', 'interest', 'money-supply' and 'ship'). By sticking to one label, we can reliably perturb the label as with the other datasets. This text classification task involves predicting the topic of snippets taken from the Reuters financial news service;
12. `TREC`: this question classification dataset includes questions that are predominantly from the news domain (Li and Roth, 2002), and labels questions as abbreviations, entities, descriptions, humans, locations or numerical values.

4.2.2 Language models

We analyse four pretrained LMs: `BERT-base` (Devlin et al., 2019), `OPT-125m` (Zhang et al., 2022b), `Pythia-160m` (Biderman et al., 2023) and `GPT-Neo-125m` (Black et al., 2021; Gao et al., 2020). The models are all transformer-based but have different types of transformer layers, and vary in their pretraining data and pretraining procedure as previously laid out in §2.1.2. The variants used for the majority of the experiments have twelve layers each, but we also explore how the results change when we move from `OPT-125m` to `OPT-1.3B` in §4.5.1. We fine-tune each model separately for the twelve tasks; Appendix B describes the hyperparameters and technical setup used for training. Fine-tuning is performed by introducing a randomly initialised classification head, and the input to that head is the `[CLS]` token for `BERT` and the final token in the sequence for the other models. We freeze the input embeddings to ensure that memorisation is limited to the layers, and fine-tune the layers along with the classification head. This type of fine-tuning was introduced by Devlin et al. (2019) (as discussed in §2.1.2) and was the de facto standard for modifying LMs for downstream tasks for years afterwards, although recently supervised fine-tuning has been popularised that fine-tunes lexical heads for downstream tasks rather than learning a new classification head (e.g. as is the case for Phi-3 and Llama-3 models, two SOTA systems at the time of writing the thesis; Grattafiori et al., 2024; Abdin et al., 2024). The pretrained models (θ_P) are fine-tuned for 50 epochs, and checkpoints are stored when the training accuracy is near-ceiling (θ_{M_1}), and at the end of training (θ_{M_2}). We also train models using the original labels (θ_O), using the same random seeds as θ_{M_1} and θ_{M_2}. Results reported in §4.2.4 and §4.5 are based on one fine-tuning seed, and the remainder of the chapter reports results computed using three seeds. Seeds control the data presentation order and the initialisation of the classification heads.

4.2.3 Localisation techniques

We apply four localisation methods, which we detail in this subsection.

Layer retraining and layer swapping First, we perform layer retraining, similar to Maini et al. (2023). We reset layers of interest using weights from θ_P, freeze the remaining layers using weights from θ_{M_2}, and retrain using clean examples for five epochs. If the resulting model maintains its performance on noisy data, the retrained layers are redundant in terms of memorisation. If the performance on noisy data decreases, that does not guarantee that memorisation can be localised to the retrained layers since the retraining objective may have multiple minima, of which only some maintain the memorisation performance. We retrain consecutive layers of window sizes ranging from one to twelve.

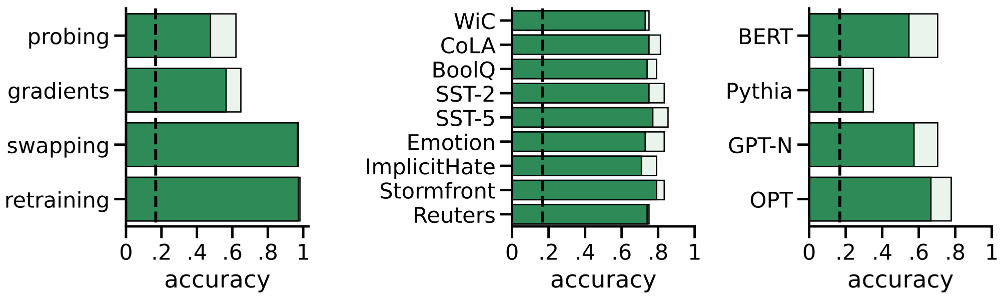
Alternatively, we swap layers between θ_{M_2} and θ_O, using the same window sizes. If swapping layers leads to a drop in performance on noisy examples while maintaining performance on clean ones, it becomes more likely that the layers were vital for memorisation (although this is again not guaranteed). We indicate layer relevance using the **memorisation error**: the ratio of incorrect predictions for noisy examples. The lower the error rate for noisy examples when retraining or swapping a layer, the less likely it is that this layer was crucial for memorisation.

Retraining or swapping all twelve layers means modifying the full model, and provides a baseline for the maximum error we can expect for the noisy data. In the results section, we will use this to normalise the results, such that the memorisation error is 1.0 when modifying all twelve layers.

Forgetting gradients We also inspect gradients, computed by back-propagating $-\mathcal{L}(\mathcal{X}_n, \mathcal{Y}_n, \theta_{M_1})$ and computing the L_1-norm per layer. We use θ_{M_1} due to gradient saturation in θ_{M_2}.[4] The assumption is that memorisation is localised in the layers requiring the largest updates when 'forgetting' noisy labels. Because gradient magnitudes do not reliably pinpoint layers, we used two tasks to decide on the norm to use and whether or not to normalise gradients using gradients for clean examples and gradients for a frozen model – we estimate these hyperparameters for gradient preprocessing using two of the twelve tasks as detailed in Appendix B.2.

Probing Lastly, we train behavioural probing classifiers (Alain and Bengio, 2017; Conneau et al., 2018; Hupkes et al., 2018), introduced in §2.1.4, to predict whether,

[4]See Akyürek et al. (2022) for a discussion of issues with gradient-based methods when tracing knowledge in a model.



(a) Per localisation technique (b) Per dataset (c) Per model

Figure 4.2: Control setup accuracy@1 (light) and accuracy@2 (dark) per localisation method, dataset or model (computed using probing and gradients), and a random baseline (dashed).

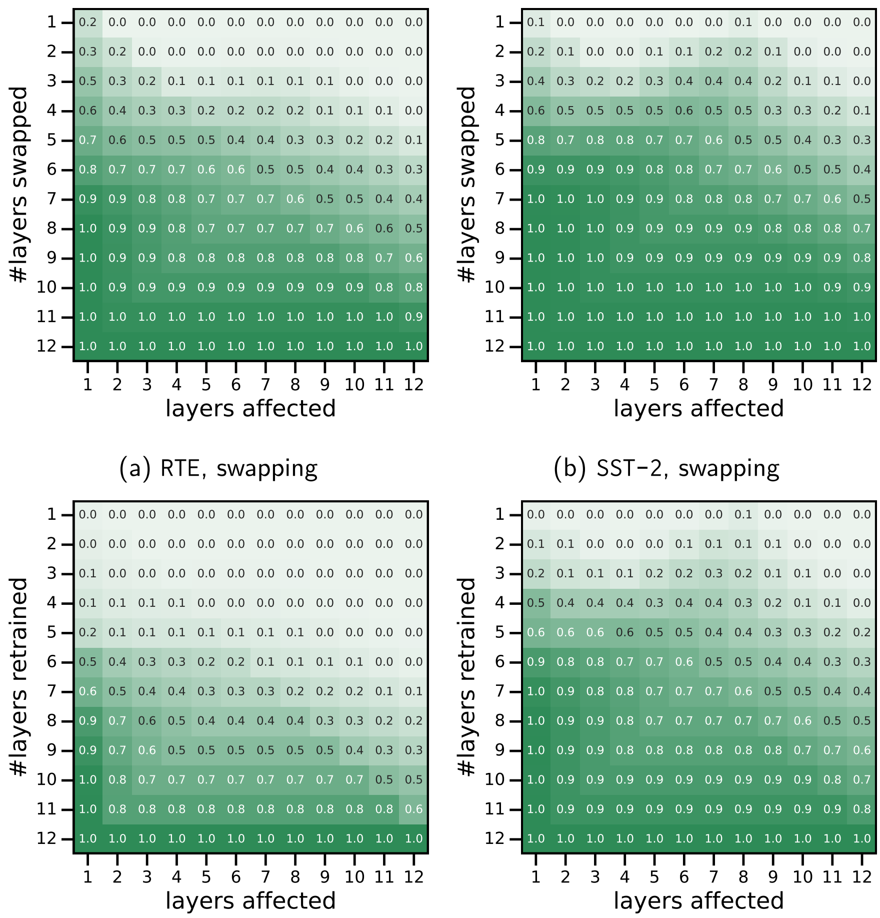
for a hidden state encoding x in layer l ($\mathbf{h}_l^{(x)}$), $x \in \mathcal{X}_n$ or $x \in \mathcal{X}_c$. The classifier is an MLP with one hidden layer, trained for 100 epochs maximum with a learning rate of $2e{-}4$. The hidden states come from *training* examples that are redistributed into a training, validation and test set for the probe. The classifiers are trained separately per layer, using five random seeds per layer. We extract the F_1-score on the test partitions and use the increase from $l{-}1$ to l as an indication of l's involvement in memorisation (except for layer 1, which we compare to the F_1-score from a probe trained on θ_P).

4.2.4 Control setup: does localisation succeed?

We now evaluate the localisation techniques by enforcing memorisation in pre-specified layers and examining whether the techniques pinpoint those layers (i.e. whether localisation succeeds).

Experimental setup We approach this as a multitask learning setup, to ensure all layers are fine-tuned, but only two are modified by the task with noisy labels: the entire model is fine-tuned using `RTE`, while the remaining task can only modify two layers at a time (layers one and two, six and seven or eleven and twelve). We train the model separately for the remaining eleven tasks and these three different choices of layer combinations. Afterwards, we first use `MRPC` and `TREC` to validate the postprocessing steps for the forgetting gradients (see Appendix B.2), after which all localisation techniques were applied to the remaining nine tasks. We evaluate the techniques using **accuracy@k**, indicating the percentage of the k highest-scoring layers that were among the correct ones for that setup, computed for $k \in \{1, 2\}$.

Results Figure 4.2a summarises the accuracies per localisation technique. Swapping and retraining are very accurate, but gradients and probing are less reliable, with accuracy@1 just over 60%. Note that the near-perfect accuracy for retraining and swapping here does not guarantee perfect accuracy in the uncontrolled setup; the per-



(a) RTE, swapping

(b) SST-2, swapping

(c) RTE, retraining

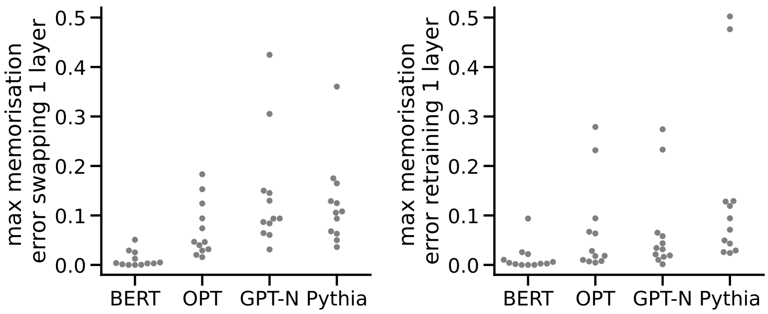
(d) SST-2, retraining

Figure 4.3: Memorisation error for layer swapping and retraining for two datasets, for the OPT model.

layer freezing is just very well-aligned with the per-layer approach of those techniques. The accuracy per dataset (Figure 4.2b) only shows slight variations. For the two lowest-scoring localisation techniques (probing, gradients), Figure 4.2c details the accuracies per model. `Pythia` scores particularly badly for the gradient analysis, for which the accuracies barely exceed the baseline. Postprocessing (Appendix B.2) did not help, which underscores gradients' unreliability.

4.3 Results for memorisation localisation

We now apply the localisation techniques to models for which all layers have been fine-tuned for one task at a time. The results indicate how important each layer is for memorisation, per dataset, per model. We cannot simply aggregate over all results (twelve layers × twelve datasets × four localisation techniques × four models), because the absolute scores returned by different techniques are not directly comparable. We discuss the results per localisation technique.



(a) Layer swapping (b) Layer retraining

Figure 4.4: Maximum memorisation error over twelve layers when modifying one layer; dots represent datasets. Jitter along the x-axis was added to improve visibility.

4.3.1 Layer swapping and retraining

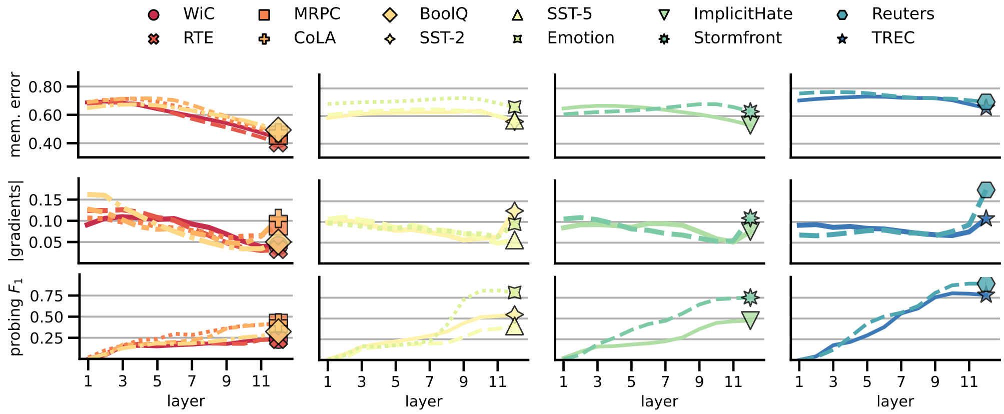
When swapping or retraining layers, we gradually modify more and more layers in θ_{M_2}, either using weights from θ_O, or by retraining layers using clean examples.[5] We modify one to twelve layers at a time, and measure the effect on the memorisation error.

Case study: `RTE` vs `SST-2` Before discussing trends across all datasets, we inspect two specific sets of results to gain a deeper understanding of the data. Figure 4.3 details memorisation error rates for `RTE` and `SST-2` (for `OPT`): in these matrices, value z in row x, column y, indicates that for all layer combinations of x consecutive layers including y, z was the mean error rate. We show the results separately for swapping and retraining.

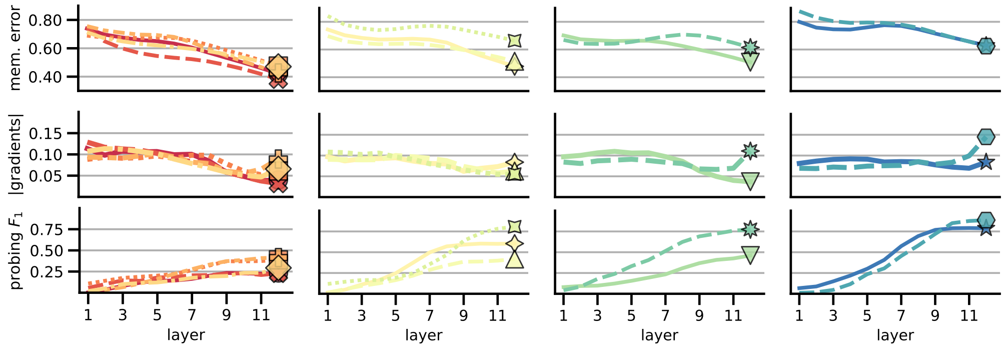
Which commonalities and differences do we observe? For both datasets, modifying a few layers only yields low error rates (see the top few light green rows), and fully reverting memorisation requires modifying seven to ten layers. Memorisation is thus not limited to a few layers, but, instead, dispersed over the model. Despite these similarities, the datasets differ in which layers appear the most crucial for memorisation: for `RTE`, modifying early layers leads to the largest increase in memorisation error, whereas for `SST-2`, both the very first layers and layers in the middle appear most relevant.

Aggregating results The findings for these two tasks are echoed in the overall swapping and retraining results. Firstly, **memorisation is not confined to individual layers**: modifying individual layers barely affects the memorisation error. This is shown in Figure 4.4, which provides the memorisation error when modifying one layer only, taking

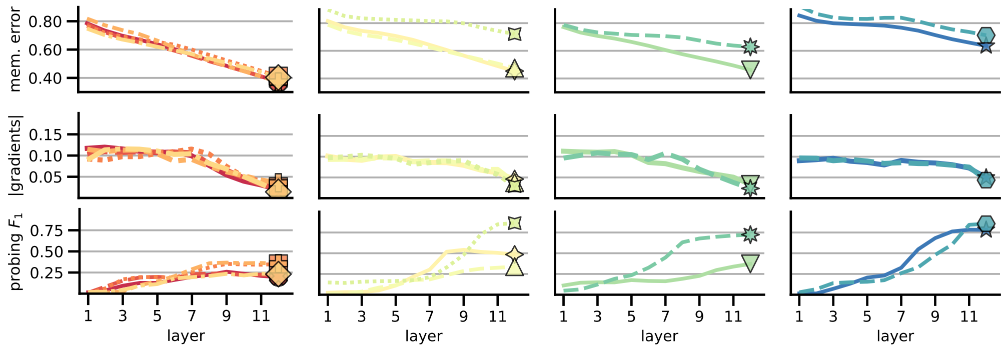
[5]When modifying parameters internally, one should try to ensure that changes to the model that go beyond the main capability focused on, are minimal. To that end, when layer swapping, we monitor errors on clean examples to ensure that the mixture of models θ_O and θ_{M_2} differs only in terms of predictions for noisy examples; the mean error for clean examples over all windows was 0.3%. When layer retraining, we similarly monitor errors on clean examples: for examples that were included in the retraining training set, this error is 0.2%, whereas for examples that were in the original training set but retraining validation set, this error is 7.6%.



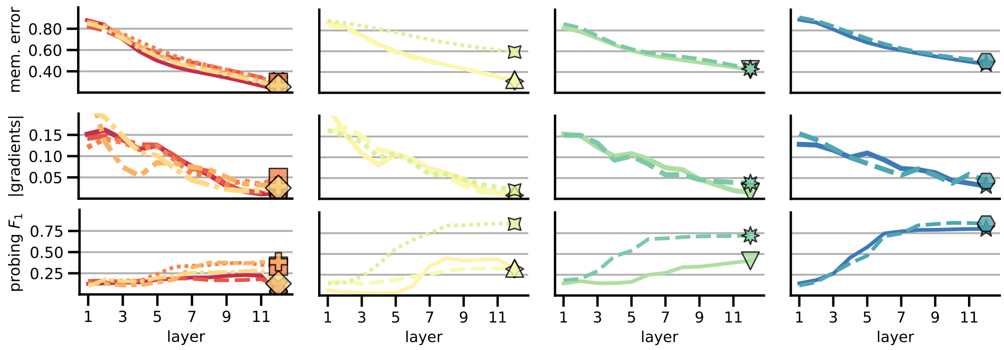
(a) BERT



(b) OPT



(c) GPT-N



(d) Pythia

Figure 4.5: Layer swapping results (top), gradients' norms' (middle), probing scores (bottom row, the *increase* between layers indicates layer relevance). From left to right, columns represent NLU tasks, sentiment tasks, hate speech classification and topic classification.

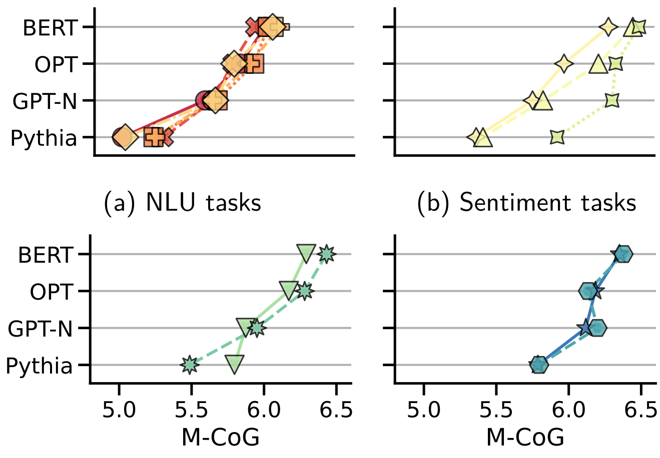


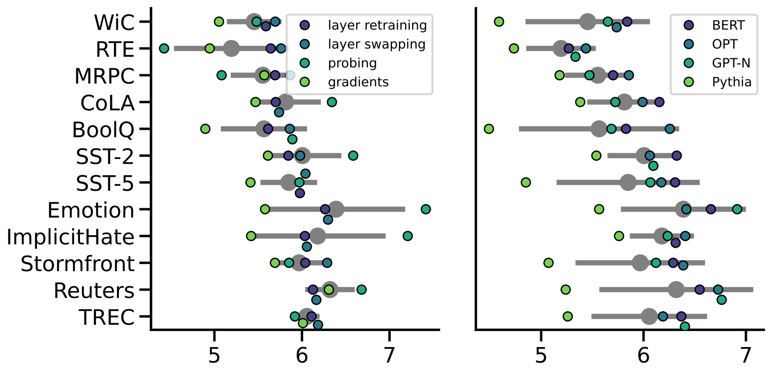
Figure 4.6: M-CoG coefficients for layer retraining, that give a coarse indication of whether lower or higher layers matter more for memorisation.

the *maximum* over layers (i.e. highlighting the largest error increase), showing datasets as dots. For most model-dataset combinations, the memorisation error rate is below 15% when modifying one layer. This agrees with findings from Maini et al. (2023), who similarly employed layer retraining to identify that memorisation in image classification is not confined to individual layers.

Secondly, **the importance of layers does appear task-dependent**. To investigate this more systematically, we express layer relevance using the mean memorisation error, computed by averaging over rows in the result matrices of layer swapping and layer retraining. Figure 4.5 details this error per model and per dataset, where the top row of each subfigure shows the layer swapping results for one model.[6] Across the board, **early layers matter more for memorisation** – as can be seen by the presence of predominantly negative slopes in these figures – but that effect is more prominent for the group of NLU tasks than for the other groups – as can be seen by the slope steepness. For `BERT` and `OPT`, in particular, there appear to be tasks that have nearly uniform memorisation errors for the different layers, emphasising that memorisation appears more distributed in these models, and less distributed in `GPT-N` and `Pythia`.

We can summarise the per-layer weights by computing a *Memorisation Centre-of-Gravity* (M-CoG), which is a weighted sum of all layers with weights summing to 1: $\sum_{i=1}^{12} \alpha_i \cdot i$. For layer swapping and retraining, α_i is the normalised memorisation error for layer i, as depicted in Figure 4.5. Figure 4.6 displays the M-CoG coefficients for layer retraining, per model, and Figure 4.7 provides M-CoG coefficients per dataset by averaging over models and over localisation techniques. The results show **strong agreement between models** in terms of the relative ordering of tasks, which is

[6] We omit layer retraining from this figure because of the high correlation between the results of layer retraining and layer swapping.



(a) Per localisation technique (b) Per model

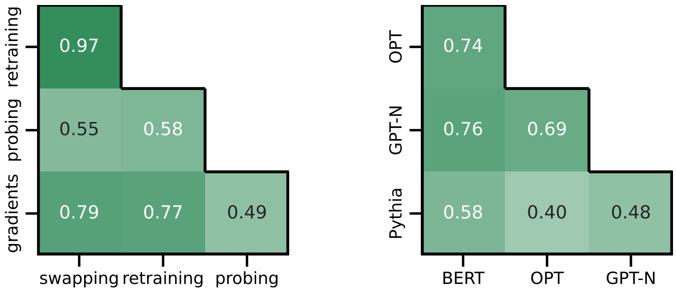
Figure 4.7: M-CoG coefficients shown per dataset and per localisation technique (left, thus averaged over models) or per model (right, thus averaged over localisation techniques). Error bars show standard deviations.

supported by the average pairwise correlation of the data from Figure 4.6 being 0.85 (Spearman's ρ). We also note that there is a very high correlation between the results from layer swapping and layer retraining: this correlation is $\rho = 0.97$ for the M-CoG coefficients (Figure 4.8a), and $\rho = 0.91$ for the raw layer weights (Figure 4.9a).

4.3.2 Probing

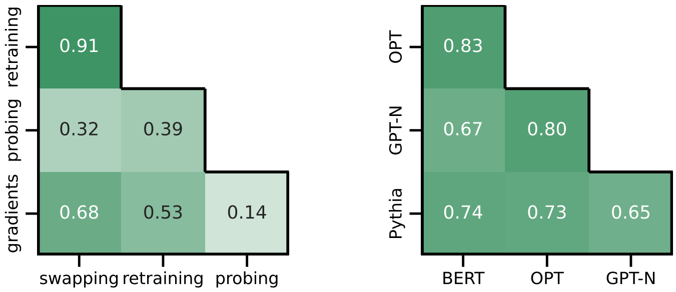
In Figure 4.5, the subfigures' bottom rows display the probing performance for the four models. The *increase* from layer to layer indicates the layers' relevance. We first observe that the performance typically does not decrease for deeper layers – i.e. representations do not 'lose' information about the fact that some examples are noisy. Secondly, the performance is quite low for NLU tasks, especially, which could mean that clean and noisy examples are more alike for these tasks than for the remaining tasks. Lastly, in accordance with the previous results, the probing performance does not change suddenly – i.e. **memorisation is not local to individual layers** – and **tasks differ in how the probing performance changes over layers**: performance flattens early for some tasks, such as **RTE**, but improves over all layers for others, such as **Emotion** and **Reuters**. For Pythia, probing performance peaks earlier than for the other models, indicating that the lower layers are extra important for this model.

To draw more generic conclusions, we compute the M-CoG coefficients by using the per-layer increase in probing performance as weights. Figure 4.7a includes the M-CoG averaged over models and demonstrates that, across the board, probing puts a larger emphasis on deeper layers compared to layer swapping and retraining. The M-CoG of probing have a moderately positive correlation to the swapping and retraining coefficients (see Figure 4.8a), and raw weights per layer have a weakly positive correlation to swapping and retraining (see Figure 4.9a).



(a) Comparing localisation techniques (b) Comparing models

Figure 4.8: Spearman's ρ, comparing M-CoG coefficients from different localisation techniques and models.



(a) Comparing localisation techniques (b) Comparing models

Figure 4.9: Spearman's ρ, comparing raw layer-wise scores from different localisation techniques and models. When comparing models, we collect weights from four techniques. Those are not directly comparable, so we apply min-max normalisation per technique.

4.3.3 Gradient analysis

Finally, we inspect the gradient norms, post-processed as described in Appendix B.2. In Figure 4.5, the middle rows of each subfigure visualise gradient norms per model. Visually, the results of the different models show some qualitative differences; most notable is that for `BERT` and `OPT`, there is a slight increase in the norm for the final layers that is absent for `GPT-N` and `Pythia`, and that for `Pythia` the results across datasets are quite similar. In spite of these differences between models, the ordering of layer relevance for the different tasks is not too dissimilar from the previous localisation methods: the M-CoG coefficients obtained using gradient norms correlate strongly with layer swapping and retraining, and moderately with probing (Figure 4.8a), and the raw layer scores correlate moderately to strongly with layer swapping and retraining (but weakly with probing) (Figure 4.9a). We should interpret these results with caution since when running the control setup, the gradients failed to pinpoint the correct layers for `Pythia` completely (§4.2.4). *That* gradients agree with swapping/retraining supports our overall findings, but we recommend against relying solely on gradients to pinpoint memorisation localisation.

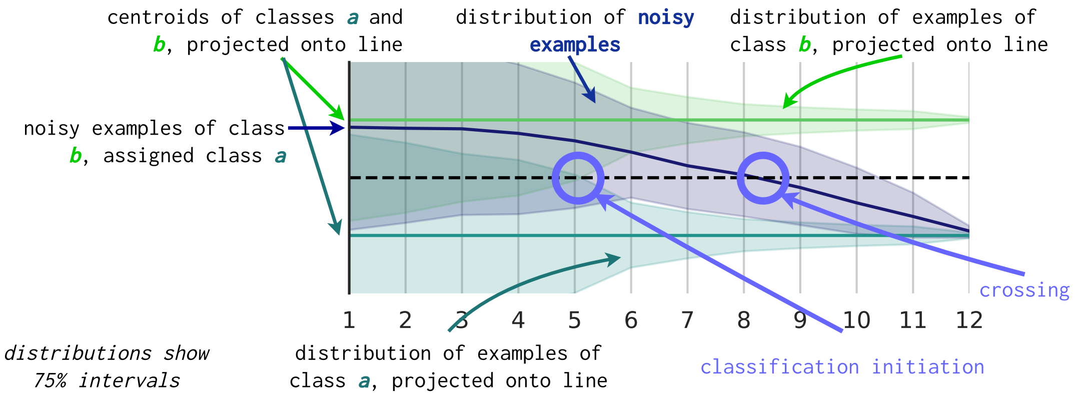


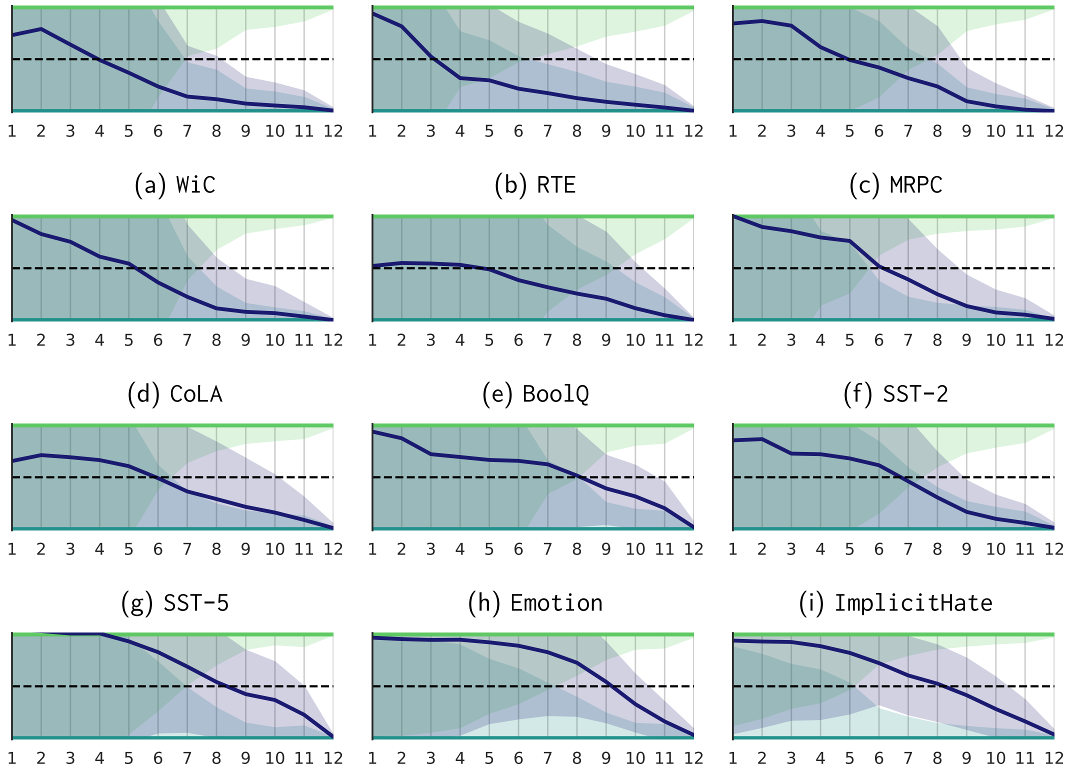
Figure 4.10: Visual explanation of the centroid analysis, using data from TREC.

4.3.4 Intermediate conclusion

In this section, we have taken a closer look at the layer-based memorisation localisation results for mislabelled ('noisy') examples, and inspected aggregated results for all techniques and models via M-CoG coefficients. Because memorisation is not strictly localised to individual layers, these coefficients lie close to the middle layer, but they do generally skew towards earlier layers and provide us with an ordering of tasks. The most notable pattern in that ordering is that the earlier layers are the most important for the NLU tasks, in particular. This is somewhat surprising since the NLP community would typically consider an NLU task, such as natural language inference, to be more complex than something like topic classification, and assumes higher-level tasks to be processed in higher layers.[7] If that is the case, it seems natural for memorisation to also happen in higher layers, but this appears to be contradicted by our experiments. At the same time, we know that lower layers affect higher layers, not only because the hidden representation of layer l feeds into layer $l+1$, but also because of the residual stream. The higher relevance for lower layers may not necessarily mean that they encode more of the memorised noisy labels. Instead, they could be more relevant due to their *steering* function, indirectly influencing higher layers too. We will return to this discussion in the next section. Independent of why the lowest layers appear most relevant, the differences between groups of tasks observed remain.

Although this section has concentrated primarily on the comparison of localisation methods, we finally note that when computing correlations between models (Figures 4.8b, 4.9b), these are strongly positive, except for `Pythia`, yielding more moderate correlations. That suggests that our results are not specific to one training setup, but somewhat generic to twelve-layer transformer-based pretrained LMs.

[7]E.g. Müller-Eberstein et al. (2023) show that for topic classification in `BERT-base` (using unperturbed datasets), the centre-of-gravity as defined by Tenney et al. (2019) lies around layer 4/5 for topic classification, whereas for natural language inference it is layer 11.



(a) WiC (b) RTE (c) MRPC

(d) CoLA (e) BoolQ (f) SST-2

(g) SST-5 (h) Emotion (i) ImplicitHate

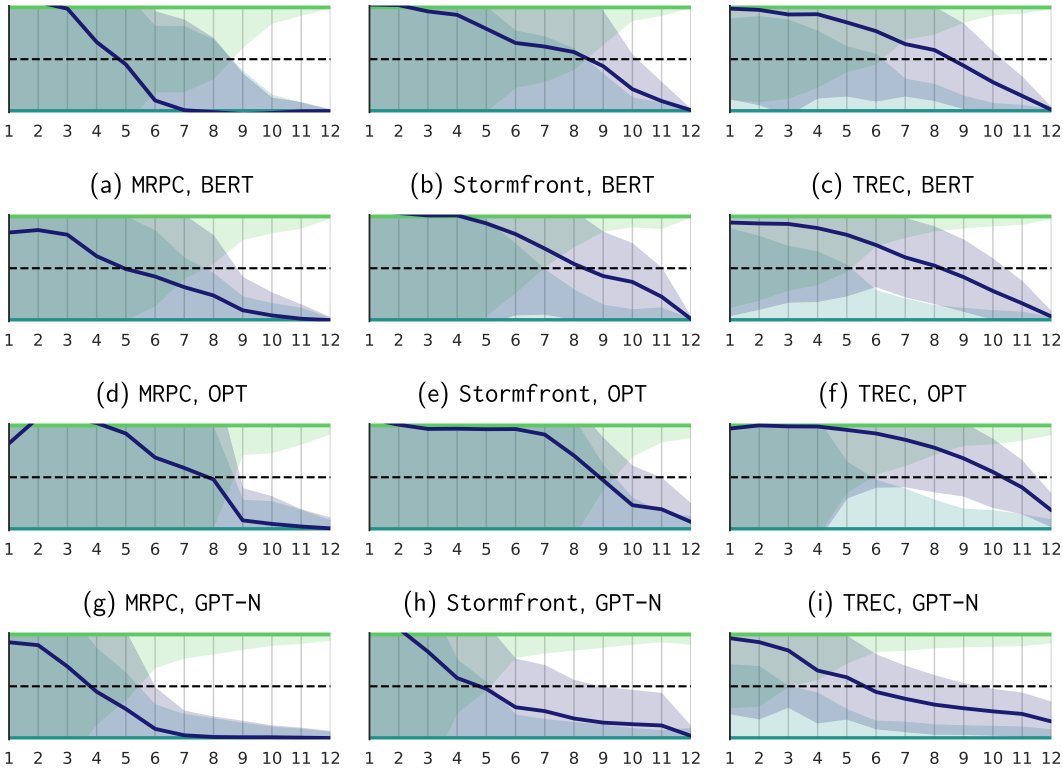
(j) Stormfront (k) Reuters (l) TREC

Figure 4.11: Centroid analysis visualisations, showing graphs for OPT for all twelve datasets.

4.4 Making memorisation interpretable via centroid analysis

The results from §4.3 suggested that earlier layers are the most relevant for memorisation. To better understand why, we make models' internal processing of memorised examples more interpretable through a **centroid analysis**. We examine pairs of classes, monitoring examples with original class y_b and noisy class y_a, for all pairs of a and b, for models θ_{M_2}. We compute the centroids of the hidden representations from the two classes, and determine the line that goes through the two centroids. Then, we compute the projections of all remaining points onto that line, measuring the distance between the line's anchor point (centroid a) and centroid b, normalised by the distance between the two centroids. This is performed separately per layer. In layer one, points belonging to y_a and y_b largely overlap. Towards layer twelve, the two classes are fully separated, and in between, the memorised examples move away from centroid b and move towards centroid a. Figure 4.10 explains this via annotations for TREC.

Figure 4.11 provides centroid analysis visualisations for all datasets using OPT, and Figure 4.12 demonstrates for three datasets how the results vary between the four models. Consistent across models and datasets is that noisy examples moving from b to a happens gradually over the course of many layers, and typically already starts in the lowest layers. This agrees with the results from the previous section, where



(a) MRPC, BERT (b) Stormfront, BERT (c) TREC, BERT

(d) MRPC, OPT (e) Stormfront, OPT (f) TREC, OPT

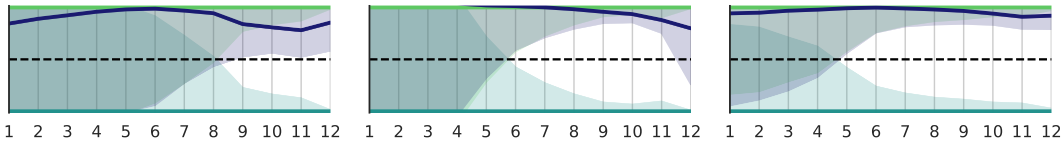
(g) MRPC, GPT-N (h) Stormfront, GPT-N (i) TREC, GPT-N

(j) MRPC, Pythia (k) Stormfront, Pythia (l) TREC, Pythia

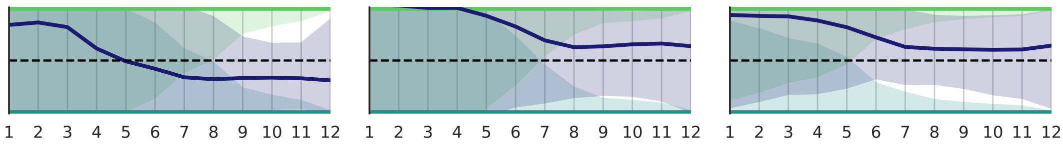
Figure 4.12: Centroid analysis visualisations for four datasets, for all four models.

we found that memorisation is not confined to individual layers, but is a cooperative process of many different layers, where lower layers are more important than deeper ones. With regards to those lower layers, the centroid analysis appears to suggest that their relevance is (as considered in §4.3.4) at least partially due to steering higher layers. After all, the representations of noisy examples do not necessarily consistently show *more* change in the first few layers, even if the first few layers generally received the highest relevance scores in the previous section.

We can also use the centroid analysis to visualise the effect of layer swapping, and further examine the roles of lower and higher layers. We construct models where the bottom six or top six layers come from θ_O, and the remaining components come from θ_{M_2}. For all datasets, we can prevent the noisy examples from moving closer to centroid a than b (on average) by replacing the bottom six layers, but for only a minority does replacing the top six have a similar effect. For instance, for OPT, when replacing the top six layers, the noisy examples remain closer to their original class only for four out of the twelve datasets (TREC, Reuters, Emotion and Stormfront). Figure 4.13 visualises the effect that swapping layers has on OPT through the centroid analysis, for three datasets: in the first row, the bottom six layers come from θ_O, whereas in the second row, the top six layers come from θ_O. Replacing the bottom six is clearly more effective at keeping the noisy examples close to their original class y_b than replacing the top six; yet, replacing



(a) MRPC, θ_O-θ_{M_2} (b) Stormfront, θ_O-θ_{M_2} (c) TREC, θ_O-θ_{M_2}

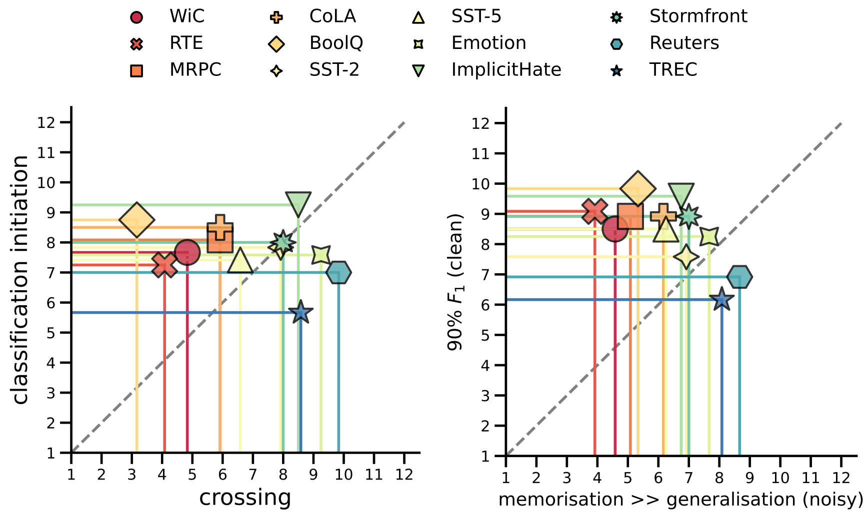


(d) MRPC, θ_{M_2}-θ_O (e) Stormfront, θ_{M_2}-θ_O (f) TREC, θ_{M_2}-θ_O

Figure 4.13: Illustration of the effect replacing 6 layers has, for MRPC, Stormfront and TREC data, for OPT. The bottom six layers come from one model, the second six layers come from another, as indicated in the subcaptions.

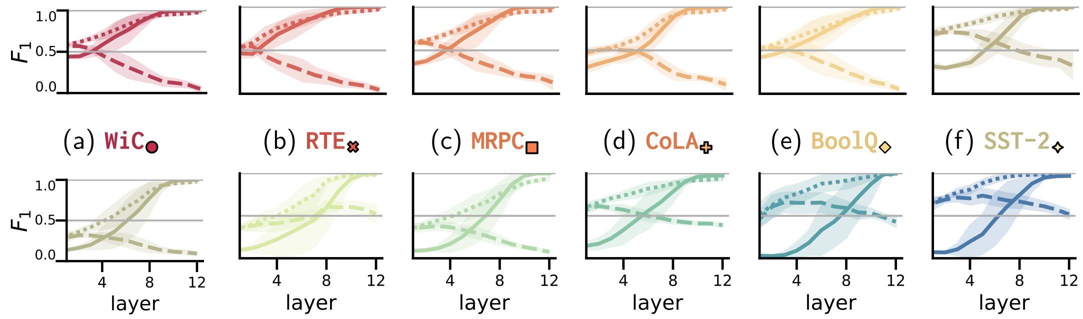
the top six is more effective for TREC and Stormfront than MRPC. This underscores that even if memorisation is gradual, it starts early. Because of that early start, intervening earlier rather than later is more successful; once the hidden representations have moved closer to a, applying the deeper layers from θ_O is insufficient to revert memorisation. At the same time, when replacing the top six, we do observe that noisy examples' hidden states do not necessarily move closer to their new class than they already were after layer six – i.e. their distance to centroids a and b stabilises. This nuances the steering function of lower layers: higher layers cannot be steered without being trained to do so; memorising noisy labels truly is a cooperative process between layers.

Task/dataset differences The demonstration of how layer swapping affects the model internally underscores that there are differences between the various tasks and datasets. Visual inspection of Figure 4.11 also indicates that there can be substantial differences between datasets, primarily in terms of how and when the noisy examples move from b to a, and what the overlap of the underlying distributions for y_a and y_b looks like. To summarise these differences, we compute two statistics: the **crossing** – the first layer in which the noisy mean is closer to a than to b – and **classification initiation** – the first layer without overlapping distributions for the two classes. Figure 4.10 provides a visual explanation of what these two statistics represent. We aggregate these statistics over models in Figure 4.14a. Many NLU tasks have a crossing that is 'earlier' than their classification initiation (e.g. **RTE** and **WiC**). The two events are closer together for hate speech and sentiment tasks, and topic classification tasks (**TREC** and **Reuters**) have crossings that are relatively 'late' compared to their classification initiations. This underscores the findings from §4.3: sentiment and hate speech tasks, and topic classification, in particular, rely more heavily on deeper layers for memorisation.



(a) Centroid analysis (b) Probing

Figure 4.14: Summary of the memorisation and classification onset for all datasets, averaged over models, computed using the centroid analysis or via probing.



(a) WiC (b) RTE (c) MRPC (d) CoLA (e) BoolQ (f) SST-2

(g) SST-5 (h) Emotion (i) ImpHate (j) Stormf. (k) Reuters (l) TREC

Figure 4.15: We train probes to predict the noisy label (solid line, shown for noisy examples) or the original label (dashed line for noisy examples, dotted line for clean examples).

Consolidation via probing The centroid analysis relies on low-dimensional projections of the hidden representations. To consolidate that we reach similar conclusions about the depth of the crossing and classification initiation using different methods, we train behavioural probing classifiers to predict an example's class from the hidden state, using (i) original or (ii) noisy labels. Figure 4.15 shows a) test F_1-scores of the noisy examples for the original label (dashed line), b) test F_1-scores of the noisy examples for the perturbed label (solid line), and c) the performance on clean examples when training with the original labels (dotted line). Tasks vary widely in terms of when the F_1-score for noisy labels exceeds that of the original labels. This happens early on for WiC and RTE, but for other tasks (e.g. SST-2, TREC and Reuters), the probe is better at predicting the original label before it can predict the noisy one.

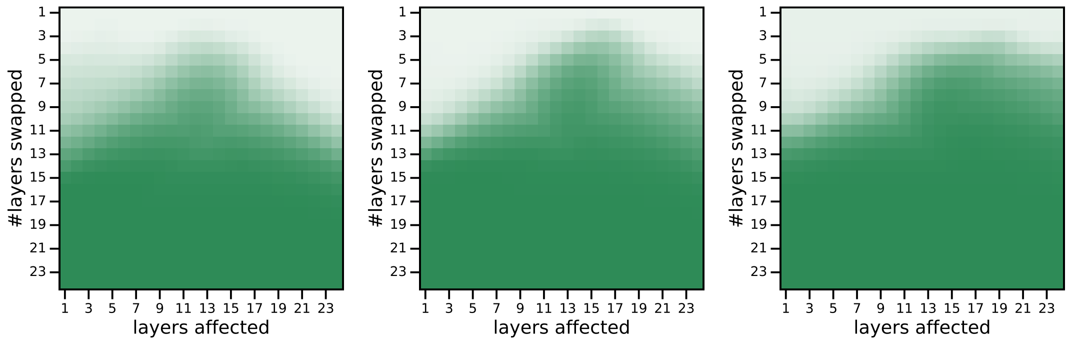
We apply the probes to noisy examples and compute a statistic similar to the

Correlates	all	BERT	OPT	GPT-N	Pythia
- *Generalisation score (thresholding applied)*					
crossing	0.75	0.88	0.94	0.94	0.72
memorisation»generalisation	0.63	0.86	0.88	0.94	0.69
M-CoG	0.56	0.78	0.69	0.92	0.69
- *Validation score (normalisation applied)*					
crossing	0.70	0.90	0.84	0.83	0.70
memorisation»generalisation	0.61	0.90	0.77	0.76	0.77
M-CoG	0.54	0.80	0.52*	0.72	0.69

Table 4.2: Spearman's ρ relating memorisation for the 12 tasks to models' generalisation performances. *: p>0.05

crossing: the layer at which the F_1 of probe ii exceeds the F_1 of probe i with ten percentage points, referred to as 'memorisation»generalisation' in Figure 4.14b. The timing of this event very strongly correlates with the crossings (Spearman's $\rho = 0.84$). We apply the probes to clean examples to compute a statistic similar to the classification initiation: the layer at which the probes' F_1 for clean examples (normalised by random guessing performance) reaches 90%. The depth of this event strongly correlates with classification initiation ($\rho = 0.73$). Together, these two events thus tell a story similar to that of the centroid analysis (Figure 4.14b), but with starker differences between the topic classification datasets and the remaining ones.

Memorisation's connection to generalisation When inspecting model internals, we have seen that the depth of memorisation (quantified as M-CoG coefficients, the 'crossing' and 'memorisation»generalisation') appears anti-correlated with the difficulty of a task. However, we have yet to have a proper way of quantifying that difficulty. We firstly take θ_{M_2}'s accuracy on **validation data**, measuring the percentual increase compared to random guessing performance to make the numbers more comparable across tasks and datasets. Secondly, we compute a metric akin to the **generalisation score** introduced in chapter 3, by training on a randomly selected 50% of the data, and evaluating on the held-out 50%, repeated with 30 random seeds to obtain a robust estimate of the generalisation score for all datapoints. Instead of taking the generalisation score directly, we apply thresholding by computing the percentage of examples for which the generalisation score exceeds random guessing, to account for the different label set sizes. As indicated in Table 4.2, these two metrics correlate moderately with the memorisation depth when combining data from all models (Spearman's $\rho > 0.54$), with most correlations being stronger when examining results per model. All in all, this suggests that the better a model generalises a task to new data, the more deeper layers are involved in memorisation.



(a) RTE (b) SST-2 (c) TREC

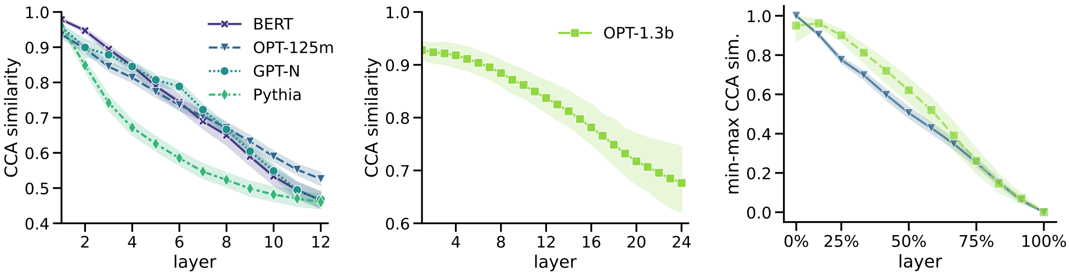
Figure 4.16: Layer swapping results for three datasets, for OPT-1.3B containing 24 layers. The graphs show the error rate for noisy examples, that goes from 0% when swapping only 1 layer to 100% when swapping all layers.

4.5 Going further: varying tasks and model size

In the previous sections, the results relied on four models of the same size (using twelve layers each) and the datasets as detailed in §4.2.1, which are quite varied in terms of their label set sizes. Here, we explore how the results would have differed had we used a larger model or a more uniform setup for the different datasets.

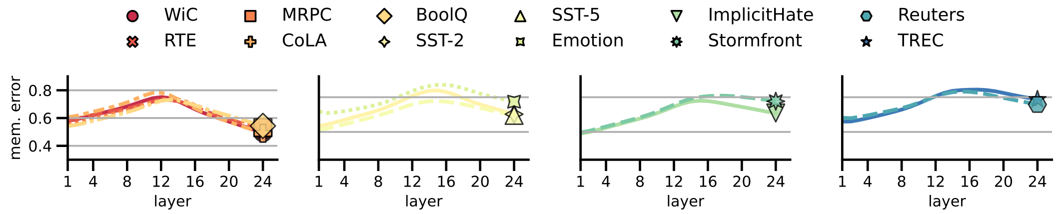
4.5.1 Increasing model size

Firstly, to examine to what extent the results observed were specific to 12-layer architectures, we apply layer swapping to the 1.3B variant of OPT, containing 24 layers and ten times the number of parameters of the other models we considered. Figure 4.16 firstly provides three example matrices, similar to the ones discussed in §4.3.1. For all three datasets shown, swapping the middle layers most effectively reverts memorisation when considering the smaller window sizes. Still, there are clear distinctions between the three datasets, too: for RTE the middle layers appear most relevant, whereas for SST-2 and TREC the upper layers are more relevant than for RTE. Figure 4.18 averages the rows from the matrices to summarise results across the twelve datasets, displaying a pattern similar to what we observed before, with NLU tasks relying more heavily on (relatively speaking) lower layers than the remaining tasks. The agreement is also reflected in Spearman's ρ comparing the layer swapping M-CoG coefficients for OPT-1.3B to the coefficients of the smaller models: $\rho = 0.73$ for Pythia, $\rho = 0.84$ for GPT-N, $\rho = 0.75$ for BERT and $\rho = 0.87$ for OPT (small). When we execute the centroid analysis and summarise the results using the crossing and classification initiation events (Figure 4.19a), we similarly observe that the crossings correlate very strongly with the crossings from the four models ($\rho = 0.94$), although the classification initiations correlate very weakly with OPT-1.3B ($\rho = 0.15$, but $p > 0.05$).



(a) 12-layer models (b) 24-layer model (c) Comparison of OPT models

Figure 4.17: CCA similarities of hidden states from θ_P and θ_{M_2}, averaged over datasets. Error bands capture 95% confidence intervals. (c) compares OPT models, with min-max normalisation applied to the similarities and the layers shown relative to the model size.

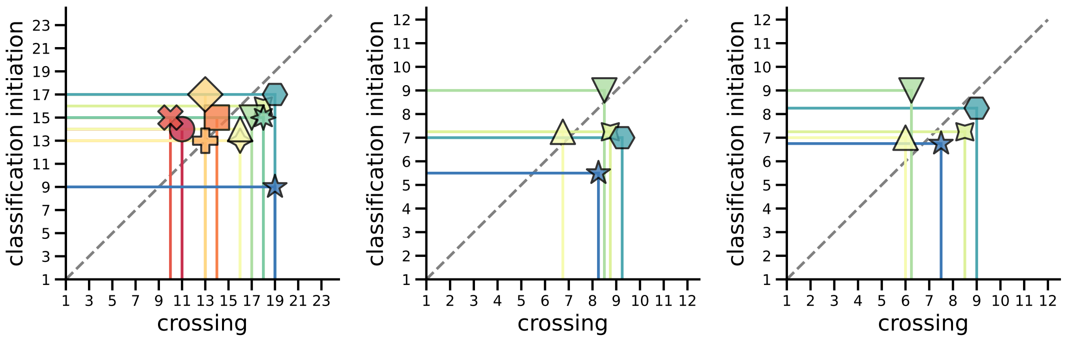


(a) NLU tasks (b) Sentiment tasks (c) Hate speech tasks (d) Topic classification

Figure 4.18: Per-layer memorisation error rate, averaged over all window sizes during layer swapping for `OPT-1.3B`. A higher error rate suggests higher relevance for memorisation.

In spite of the agreement in terms of the ordering of tasks, there is a difference compared to the results of the twelve-layer models since the lowest layers (in absolute terms) appear much less relevant. One potential cause could be the interaction of memorisation with general fine-tuning training dynamics: various sources identify that lower layers are less affected by fine-tuning procedures compared to deeper layers (Merchant et al., 2020; Mosbach et al., 2020, 2021), and this might be exacerbated in deeper models. To investigate this for our own models, we gather hidden representations for the training data using θ_{M_2} (the same representations used for probing) and compare them to the corresponding hidden representations from θ_P. We compare them using CCA, a technique described in §2.1.4 that finds linear transformations of two sets of representations such as to maximise the correlation between them. Figure 4.17 presents those CCA similarities for the twelve-layer models and for `OPT-1.3B`. Because `OPT-1.3B`'s representations have higher similarities across the board, we compare the two `OPT` models in Figure 4.17c by min-max scaling, which demonstrates that, relative to the smaller model, earlier layers are less affected by fine-tuning in `OPT-1.3B`.

Interestingly, with the results in Figure 4.17a, we have stumbled upon another finding, which is unrelated to the model sizes: the earliest layers are by far the most affected in `Pythia`. Previously, in Figure 4.7b, we already noticed that this model has



(a) OPT-1.3B (b) Multi-class tasks (c) Binarised tasks

Figure 4.19: Summary of the memorisation and classification onsets for (a) OPT-1.3B and for the four twelve-layer models for the (b) multi-class tasks and (c) their binarised versions.

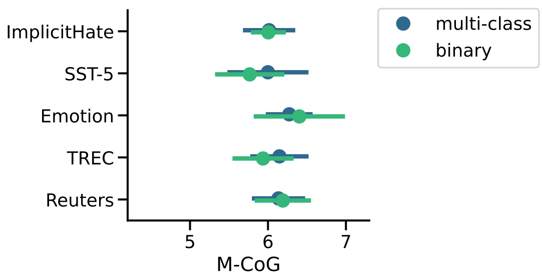


Figure 4.20: M-CoG coefficients for layer swapping, comparing multi-class to binarised tasks. Error bars show standard deviations over models.

M-CoG coefficients that are quite low, which fits with the more generic observation that earlier layers are more relevant, but that 'early' might be relative to the effects of fine-tuning. What could drive these out-of-the-ordinary fine-tuning effects on Pythia? If we reconsider the different architectures laid out in §2.1.2, the most prominent difference to the other three models is the parallelised layout of the transformer layer. Previously, pre-normalised transformer layers were proposed to improve upon post-normalised layers suffering from vanishing gradients (e.g. for discussions on the behaviour of pre-normalised transformer, see Xiong et al., 2020; Takase et al., 2023). We hypothesise that the more shallow computational graph of the parallelised layer dampens the effect of vanishing gradients even more, because of the reduced number of consecutively applied LayerNorm operations. Investigating this further lies outside the scope of this chapter.

4.5.2 Binarisation of tasks

Finally, having identified that our tasks differ in terms of the layers that matter most for memorisation, we should also note that the tasks and datasets with the largest M-CoG coefficients in §4.3 and the deepest crossings in §4.4 also happen to be the tasks that do not have a binary label set – e.g. consider Figure 4.14a, where among the six deepest crossings, there are five from multi-class tasks. To ensure that the effect observed is not specific to tasks with a large label set size, we now change the multi-class tasks (`SST-5`, `Emotion`, `ImplicitHate`, `TREC`, `Reuters`) into binary classification and repeat layer swapping and the centroid analysis. We do this by taking the most frequent two classes for a task, and training models again with 15% of the labels perturbed, using one model seed only. We now compare these models to the same model seed trained on the multi-class variant of the same tasks.

For layer swapping, the M-CoG of the multi-class and binary setups correlate with Spearman's $\rho = 0.84$, combining datapoints from all four models (see Figure 4.20); those same coefficients have a mean difference of -0.05 and a mean absolute difference of 0.16, meaning that overall, the coefficients differ only slightly.

When we repeat the centroid analysis and compute the crossing and classification initiation events, those similarly correlate strongly before and after binarisation ($\rho = 0.90$ with $p < 0.05$ for the crossing and $\rho = 0.67$ with $p > 0.05$ for the classification initiation). Figure 4.19c shows the events when averaged over models. And when we look at the absolute numbers obtained for these two events, the crossing is an average of 0.85 layers earlier, and the initiation is an average of 0.45 layers later, meaning that although the binarised tasks yield slightly different results, they still starkly differ from the results obtained for the group of NLU tasks.

4.6 Conclusion and discussion

In this chapter, we set out to contribute a crucial piece of the puzzle in the memorisation localisation landscape, by employing an experimental setup that is the NLP equivalent of seminal work from CV. Whether earlier layers or deeper layers are more responsible for memorisation has been widely discussed, and our contribution helped us determine whether the 'deeper layers' answer from CV truly contrasts with the 'lower layers' answer from the majority of NLP studies, or whether that was simply unique to noise memorisation. We performed memorisation localisation using classification tasks by perturbing a subset of the labels and tracing those 'noisy' examples over layers. Applying four localisation techniques to four models crystallised that memorisation is not local to specific layers but a cooperative process of weights from many layers.

Nonetheless, not all layers appear equally important. Overall, early layers are more important than later ones, which is supported by both the results from §4.3 and the centroid analysis in §4.4: the model's manipulation of memorised examples *starts* in lower layers, and to prevent memorisation, early intervention was thus more successful than late intervention. Lower layers do not necessarily fully implement memorisation, but might also steer higher layers. At the same time, the visualisation of layer swapping in §4.4 demonstrated that memorisation truly is a cooperative process between layers, since higher layers cannot be steered unless they are trained to do so. We mainly discussed results for 12-layer models, and demonstrated, for a subset of the results, how they change when moving to a 24-layer model. This consolidated that memorisation is a gradual process, and taught us that it is still the earlier layers that are more important, but that 'early' might be relative to the model's size and is affected by the fine-tuning procedure. This is not in accordance with the generalisation-first, memorisation-second hypothesis but does agree with the most recent work on image classification in CV by Maini et al. (2023) – who found noise memorisation to be dispersed over layers. It also aligns with the related work from NLP that primarily points to lower layers as being most responsible for fact memorisation and verbatim memorisation, while also describing cooperative roles for earlier and deeper layers (e.g. Geva et al., 2023; Chang et al., 2024; Haviv et al., 2023).

Can we, due to the importance of early layers, conclude that our results falsify the generalisation-first, memorisation-second hypothesis? The centroid analysis results suggest that this question requires a nuanced answer due to the variation observed among tasks. The depth of memorisation is positively correlated with a model's generalisation capabilities – i.e. we do observe a generalisation-first, memorisation-second tendency, but only at the level of the different tasks and not within every individual task. In addition to observing variation across tasks, we also identified some variation across the four LMs. Although the per-layer weights of the localisation techniques and the M-CoG coefficients generally correlated strongly positively across the different models, one model stood out: `Pythia` has the lowest agreement with other models, and displays the highest relevance of earlier layers. We discussed how this could be related to architectural differences between models, although further investigations would be needed to consolidate this.

Finally, we would like to mention that the dispersed nature of memorisation implies that editing model weights locally does not necessarily erase memorised information, even if a flipped label suggests this at the level of the output layer. To give a concrete example of this, let us take another look at Figures 4.13a to 4.13c, where we replaced the bottom six layers, which successfully kept the hidden states of noisy examples from moving towards their newly assigned label. And yet, if we inspect the figures closely, we still observe some changes to the hidden states in the higher layers, even if that does not move them all the way to the noisy centroid, meaning that this change might not be observable in terms of the model's final prediction. The higher layers thus still capture some information about the atypicality of these noisy examples. This might be harmless when purposefully reverting memorisation of mislabelled examples, or when editing facts about named entities like cities. For fact editing, Hase et al. (2023) demonstrated that editing success does not correlate well with fact localisation results – i.e. you can change the behaviour of the model by changing layers that did not initially seem to encode that information. This underscores that there is a difference between behavioural changes in the model and true *erasure*, where the latter is much more important when considering memorisation of PII (e.g. Carlini et al., 2021). If PII is similarly encoded in a distributed way, editing information locally in individual layers might keep the model from emitting it at the output level, but might not actually remove the information from the entire model. This might be one of the reasons why safety measures can easily be reversed in pretrained LMs modified to reduce harmful outputs (e.g. Zhan et al., 2024). How to reliably remove information encoded in a distributed manner is an important avenue for future work.

4.6.1 Limitations

We identify five main limitations of our work, of which the first is that our data reflects a strongly **simplified variant of memorisation**. To trace memorised examples over transformer's many layers, we resorted to label flipping to create 'noisy' examples. This situation is somewhat unnatural when considering real-world examples that require memorisation from LMs. For example, in the case of sentiment analysis, that might be a sarcastic phrase whose sentiment is the opposite of what is expected based on a literal interpretation. We cannot guarantee that our noisy examples behave in the same way as real-world examples would. Similarly, memorisation of noisy examples need not affect models in the same way as the memorisation of facts or sequences, and memorisation during fine-tuning might behave different from memorisation during pretraining. As laid out in the introduction of this chapter (§4.1), we opted for this type of data manipulation to create an experimental setup that more closely resembles that of related work from CV.

Secondly, our analyses were focused at localisation at **the level of the layers**. Yet, a layer may not be the right granularity – even when examining whether memories are stored within an individual layer – since a layer will contain many more parameters than those memorising the data of interest. As a result, when applying layer retraining or swapping, we are potentially modifying more than needed. Because of the focus that related work has had on the layer level, too, we consider our approach to be a valuable contribution, nonetheless.

Thirdly, we arrived at our conclusions by applying four localisation techniques, but it should be noted that the **localisation techniques themselves are imperfect**. In our control setup (§4.2.4) where only two layers were modified during fine-tuning, probing and gradient analyses could not accurately pinpoint those two layers, and the techniques that could pinpoint them (layer swapping and layer retraining) are more reliable at determining which layers are *not* crucial for memorisation than at determining which ones are. Because of the general agreement between the techniques and the results from §4.3-4.4 we do think our conclusions are robust, but the absolute numbers of layer relevance should be taken with a grain of salt.

Related to the imperfection of localisation techniques is the potential **unreliability of the centroid analysis**, which we introduced as a way of visualising what is happening to examples over the different layers. This visualisation is a one-dimensional projection of hidden representations and thus an extreme simplification of the intricate process of memorisation. We do not mean to use it as a localisation technique per se, but as a way to explain the outcomes of other experiments. At the same time, we did observe that the 'crossing', the depth at which the hidden representations are closer to the centroid of the mislabelled class than their original class, yields similar correlations to models' generalisation performance as the comparison using M-CoG coefficients. This, at the very least, suggests the results from §4.3 and §4.4 do align with one another.

Lastly, we analysed the group of noisy examples as a whole and concluded that many layers work together to gradually shift examples from their original class to the newly assigned class. However, we **have not examined individual examples**; it could still be the case that for individual examples, memorisation is more localised to specific layers. We only have preliminary results suggesting that individual examples, too, are memorised over multiple layers, which is the fact that in §4.3, swapping and retraining individual layers was mostly unsuccessful in increasing the memorisation error rate.

4.6.2 Retrospective and outlook

In this chapter, we reviewed fine-tuned LMs. Fine-tuning has been the predominant strategy to turn LMs into narrow experts for a given task ever since the release of BERT (Devlin et al., 2019). However, the community has turned away from applying the standardised modelling pipeline of pretraining and fine-tuning. Since the early years of this decade, prompting and in-context learning (consider Liu et al., 2023b; Dong et al., 2024, for extensive reviews of work in these respective directions) have been widely explored as data-efficient manners to tune models for tasks on the go. This moved the focus away from what models acquire during fine-tuning and instead put the focus on what models may have already memorised during the pretraining stage (as is the case for fact memorisation and verbatim memorisation). At the time of writing this thesis, however, fine-tuning is making a 'comeback' in a different form: after the pretraining stage, many models are further trained in a supervised fine-tuning stage, and a stage that relies on reinforcement learning using human feedback (e.g. for Phi-3 and Llama-3 models; Grattafiori et al., 2024; Abdin et al., 2024). Supervised fine-tuning differs from traditional fine-tuning in that it prepends a task description to an input example and fine-tunes lexical heads for downstream tasks instead of learning a new classification head per task. It is also typically applied in a multitask setup. Otherwise, the setups are not dissimilar from one another, which is why we expect our main findings concerning the importance of lower layers to generalise to supervised fine-tuning. The multitask setup will likely influence the relative ordering of task-specific memorisation in the network, but further experimentation would be needed to consolidate this. Our work remains a key piece in the memorisation landscape, linking insights from CV and NLP, in spite of the shift towards supervised fine-tuning, but we do recognise that the focus on **traditional fine-tuning** is a limitation.

We conclude by noting that the debate on the relevance of earlier vs deeper layers remains open, as studies that appeared after publication of this chapter offer, again, conflicting evidence. Menta et al. (2025) bypass the attention mechanisms of `GPT-N` and `Pythia` in various layers, establishing that verbatim memorisation is best prevented by performing that intervention in the later layers. In contrast, Huang et al. (2024) show that intervening in hidden representations of the lower layers in `Pythia` is more effective at preventing verbatim memorisation. They also emphasise that memorisation is distributed across layers and input tokens, with certain words acting as 'triggers', such as named entities. While some words in memorised sequences were recalled directly, others were inferred using general language modelling capabilities. These contrasting findings are puzzling, yet unsurprising, given the varied answers to the memorisation localisation question we previously reviewed in §2.2.3. We encourage future work to run controlled comparisons, either by testing multiple memorisation localisation methods within a unified setup or by analysing different memorisation types together (e.g. PII, facts, idioms, noise, verbatim memorisation). Finally, Huang et al.'s insight into the overlap between memorisation and general language modelling aligns with our finding that clean and noisy examples are not perfectly separated across layers: there might simply not be a stark distinction between parameters that store memorised examples and parameters that perform the main task of interest.

Part II

(Non-)compositionality: a memorisation–generalisation case study

In the previous two chapters, I discussed memorisation of data primarily as something that can apply to a range of different examples in our dataset. Our models train on a dataset, and some examples are memorised, whereas others are not. In this context, generalisation was discussed as the models' performance on standardised or IID evaluation data. I now turn to a more narrow case study of memorisation and generalisation, within the context of (non-)compositionality. As laid out in §2.3.1, natural language itself is assumed to be compositional, enabling humans to use language productively. It allows them to understand and produce sentences they have never heard before. Whether neural networks are similarly compositional language learners has been a topic of ongoing debate, and dedicated datasets for studying this have been reviewed in §2.3.2. Here, generalisation is thus a specific type of OOD generalisation, where evaluation sets assess whether models have a specific capability related to the compositionality of language.

At the same time, natural language is not fully compositional, due to, among other things, formulaic sequences whose meaning deviates from a compositional interpretation. Memorisation of the correct meaning is crucial if computational models are to understand such sequences. In the chapters that follow, I approach memorisation through "the archetypal formulaic sequence" (Wray, 2002): the idiomatic expression. There is thus a bigger emphasis on what models *should* memorise instead of what they actually memorise (which was the focus of the previous chapters).

The second part of the thesis includes two chapters. In chapter 5, based on Dankers et al. (2022a), I evaluate compositional generalisation in NMT via 'systematicity' and 'substitutivity' tests, along with studying idiom acquisition during model training (requiring non-compositional processing, evaluated using the 'overgeneralisation' test). The results underscore that, paradoxically, transformer is both not compositional enough and too compositional at once. Afterwards, chapter 6, based on Dankers et al. (2022b), elaborates on internal mechanisms that NMT systems have developed for coping with idiomatic expressions in translation, digging into how, for example, "out of the blue" on the source side morphs into "uit het niets" ("*out of nothing*") on the target side. The analyses focus on the extent to which idioms are captured as one unit and on the interaction between idioms and their surrounding context. We inspect self-attention and cross-attention patterns, study how hidden representations change over layers, and adopt (amnesic) probing to identify whether figurative translations can be predicted using hidden representations. Methodologically, the two chapters thus rely on behavioural experiments and interpretability studies, respectively.

Chapter 5

The paradox of (non-)compositional generalisation

5.1 Introduction

Compositionality is assumed to play an essential role in how humans understand language, but whether neural networks also exhibit this property has long been a topic of vivid debate (e.g. Fodor and Pylyshyn, 1988; Smolensky, 1990; Marcus, 2003; Nefdt, 2020). Studies about the compositional abilities of NLP models have mostly been focused on synthetically generated datasets, with simplified languages, in which compositionality can be ensured and isolated (e.g. Lake and Baroni, 2018; Keysers et al., 2019; Hupkes et al., 2020; Kim and Linzen, 2020), as we previously reviewed in §2.3.2. In such tests, the interpretation of expressions is computed according to the *strong* or *local* definition of compositionality (see §2.3.1 for a discussion of the many definitions of compositionality) without acknowledging that natural language is riddled with exceptions to strong compositionality. Idioms are one such example, since the meaning of most idioms cannot fully be derived from their parts. Still, other examples of local compositionality violations come to mind as well. Sometimes expressions' meanings depend on their parts in a compositional way, but arriving at this meaning requires a more *global* approach because disambiguation is needed – for example, consider homonyms ("these dates are perfect for our dish/wedding") or scope ambiguities ("every human likes a cat").

The tension between local and global forms of compositionality not only inspired many debates on the most adequate characterisation of natural language, but it also affects the evaluation of compositionality in NLP models. On the one hand, we want models to be robust and reliable, for which local compositionality is assumed to be of help. At the same time, natural languages are rife with formulaic language and other exceptions to compositionality, thus requiring NLP systems to balance compositional

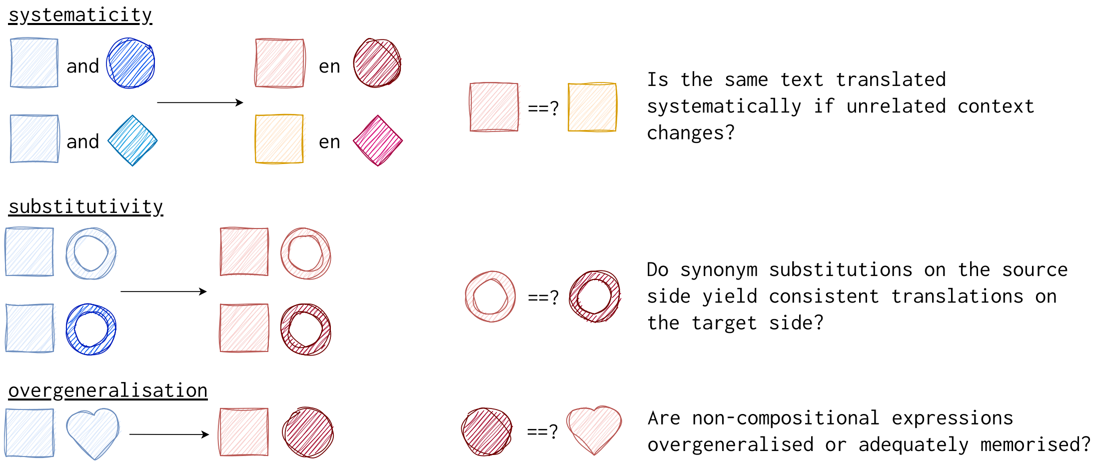


Figure 5.1: A visual summary of the three tests we run: *systematicity* measures translations' consistency as the context material changes, *substitutivity* evaluates translations' consistency under meaning-preserving synonym substitutions (e.g. replacing "donut" with "doughnut"), and *overgeneralisation* evaluates whether translations of idioms (such as learning something "by heart") are compositional ("door het hart") or non-compositional ("uit het hoofd").

and non-compositional processing adequately.

In this chapter, we discuss this paradox through the analysis of NMT systems' outputs, contrasting compositional generalisation capabilities to non-compositional processing and memorisation. Prior to publishing this chapter, there were no existing evaluation datasets for compositional generalisation in MT for models trained on natural language. We present new data to address this gap in the literature, where the data is used to reformulate three theoretically grounded tests from Hupkes et al. (2020)[1]: *systematicity*, *substitutivity* (both evaluating compositional processing) and *overgeneralisation* (evaluating non-compositional processing of idioms). Figure 5.1 provides a visual summary of the three tests. Afterwards, we evaluate transformer NMT systems (Vaswani et al., 2017) trained on English-Dutch data from the OPUS corpora collection (Tiedemann and Thottingal, 2020). This chapter thus contributes to the overarching research question RQ3, with a specific focus on (non-)compositionality: "*To what extent are memorisation and generalisation at odds with one another?*" In the process, we answer the following sub-questions:

1. *How can we reformulate theoretically-grounded compositionality tests outside of toy task scenarios for NMT?* We provide a reformulation of the systematicity, substitutivity and overgeneralisation tests originally proposed by Hupkes et al., reimagined for English-Dutch data in an NMT setup, using unaltered natural language training corpora. We provide our experimental setup and preliminaries on the data used in §5.2, after which §5.3 lays out how the individual tests are

[1] I was a core contributor of this article, but it is not a part of this thesis.

performed.

2. *How compositional are NMT systems, and is the source of the errors natural language variation or model behaviour?* We analyse transformer's compositional abilities via the systematicity and substitutivity tests (§5.3.1, §5.3.2). In §5.4 we manually analyse errors made by the models to study the source of those errors.

3. *How do NMT systems acquire non-compositional translations of idioms, and how does this align with generalisation performance?* We analyse transformer's non-compositional abilities via the overgeneralisation test (§5.3.3), while elaborating on how compositional generalisation and overgeneralisation change during training.

Our results firstly indicate that models often do not behave compositionally under the strict, local interpretation. Some inconsistencies in the compositionality tests reflect natural variation in language, whereas others are actual mistakes. Secondly, we find that models acquire idiomatic translations in two phases: early on during training, the models learn to overgeneralise word-for-word translations, and later on, they start to memorise paraphrased translations. Models' convergence based on memorisation does not appear to align with the other evaluation metrics – i.e. BLEU scores on unseen data and compositional generalisation. Following the description of our tests and results, we end this chapter in §5.5 with a discussion of our findings, the limitations of our approach and an overview of work that appeared following the publication of this chapter.

With our study, we contribute to ongoing questions about the compositional abilities of neural networks, and we provide nuance to the nature of this question when natural language is concerned: how local should the compositionality of models for natural language actually be? Apart from an empirical study on the compositionality paradox and idiom acquisition, our work also presents a call to action to the community: we should rethink the evaluation of compositionality in neural networks and develop benchmarks using *real* data to evaluate compositionality on natural language, where composing meaning is not straightforward since compositional and non-compositional examples co-occur.

5.2 Experimental setup

We train transformer models with English as the source language and Dutch as the target language. In this section, we first elaborate on the training setup used, followed by an introduction to the data used for evaluation of these models.

5.2.1 Training NMT systems

We train `transformer-base` models (Vaswani et al., 2017) using the `fairseq` toolkit (Ott et al., 2019). Our training data consists of a collection of MT corpora bundled in OPUS

Training set size	BLEU dev	BLEU devtest
small	20.79±0.19	20.54±0.39
medium	24.47±0.36	24.27±0.27
full	25.95±0.19	25.72±0.08

Table 5.1: BLEU scores for the 'dev' and 'devtest' subsets of the FLORES datasets, for models trained on corpora of three sizes, with standard deviations computed over five seeds per training set size.

#	Template
1	The N_{people} $V_{transitive}$ the N^{sl}_{people} .
	E.g. The poet criticises the king .
2	The N_{people} Adv $V_{transitive}$ the N^{sl}_{people} .
	E.g. The victim carefully observes the queen .
3	The N_{people} P the $N^{sl}_{vehicle}$ $V_{transitive}$ the N^{sl}_{people} .
	E.g. The athlete near the bike observes the leader .
4	The N_{people} and the N_{people} $V^{pl}_{transitive}$ the N^{sl}_{people} .
	E.g. The poet and the child understand the mayor .
5	The $N^{sl}_{quantity}$ of N^{pl}_{people} P the $N^{sl}_{vehicle}$ $V^{sl}_{transitive}$ the N^{sl}_{people} .
	E.g. The group of friends beside the bike forgets the queen .
6	The N_{people} $V_{transitive}$ that the N^{pl}_{people} $V^{pl}_{intransitive}$.
	E.g. The farmer sees that the lawyers cry .
7	The N_{people} Adv $V_{transitive}$ that the N^{pl}_{people} $V^{pl}_{intransitive}$.
	E.g. The mother probably thinks that the fathers scream .
8	The N_{people} $V_{transitive}$ that the N^{pl}_{people} $V^{pl}_{intransitive}$ Adv .
	E.g. The mother thinks that the fathers scream carefully .
9	The N_{people} that $V_{intransitive}$ $V_{transitive}$ the N^{sl}_{people} .
	E.g. The poets that sleep understand the queen .
10	The N_{people} that $V_{transitive}$ Pro $V^{sl}_{transitive}$ the N^{sl}_{people} .
	E.g. The mother that criticises him recognises the queen .

Table 5.2: The synthetic sentence templates, modelled after synthetic data from Lakretz et al. (2019).

(Tiedemann and Thottingal, 2020), of which we use the English-Dutch subset provided by Tiedemann (2020), which contains 69M sentence pairs. To examine the impact of the amount of training data – a dimension that is relevant because compositionality is hypothesised to be more important when resources are scarcer – we train one setup using the **full** dataset, one using $\frac{1}{8}$ of the data (**medium**), and one using one million source-target pairs in the **small** setup. For each setup, we train models with five seeds and average the results.[2]

To evaluate our trained models, we adopt FLORES-101 (Goyal et al., 2022), which contains 3001 sentences from Wikinews, Wikijunior and WikiVoyage, translated by

[2] All training details are listed in Appendix C.2.

#	Template
1	The N_{people} (VP TO (VP VB (NP NP (PP IN (NP NP (PP IN NP)))))) *E.g. The woman wants to use the Internet as a means of communication .*
2	The N_{people} (VP VBP (VP VBG (S (VP TO (VP VB (S (VP TO VP)))))))) *E.g. The men are gon na have to move off-camera .*
3	The N_{people} (VP VB (NP NP (PP IN NP)) (PP IN (NP NP (PP IN NP)))) *E.g. The doctors retain 10 % of these amounts by way of collection costs .*
4	The N_{people} reads an article about (NP NP (PP IN (NP NP (PP IN (NP NP (PP IN NP)))))) *E.g. The friend reads an article about the development of ascites in rats with liver cirrhosis .*
5	The N_{people} reads an article about (NP (NP DT NN) (PP IN (NP NP (SBAR (S (WHNP WDT) VP))))) . *E.g. The teachers read an article about the degree of progress that can be achieved by the industry .*
6	An article about (NP NP (PP IN (NP NP (PP IN (NP NP (PP IN NP)))))) is read by the N_{people} . *E.g. An article about the inland transport of dangerous goods from a variety of Member States is read by the lawyer .*
7	An article about (NP NP (PP IN (NP NP (, ,) (SBAR (S (WHNP WDT) VP))))) , is read by the N_{people} . *E.g. An article about the criterion on price stability , which was 27 % , is read by the child .*
8	Did the N_{people} hear about (NP NP (PP IN (NP NP (PP IN (NP NP (PP IN NP)))))) . *E.g. Did the friend hear about an inhospitable fringe of land on the shores of the Dead Sea ?*
9	Did the N_{people} hear about (NP (NP DT NN) (PP IN (NP NP (SBAR (S (WHNP WDT) VP))))) ? *E.g. Did the teacher hear about the march on Employment which happened here on Sunday ?*
10	Did the N_{people} hear about (NP NP (SBAR (S (VP TO (VP VB (NP NP (PP IN NP))))))) ? *E.g. Did the lawyers hear about a qualification procedure to examine the suitability of the applicants ?*

Table 5.3: The semi-natural sentence templates used. The syntactic structures for NPs and VPs in purple are instantiated with data from the OPUS collection, and nouns marked in blue are instantiated using the same vocabulary used for the synthetic data. The predefined tokens in black remain the same.

professional translators, split across three subsets. We train the models until convergence on the development set, with a patience of 10 epochs. The BLEU scores for development and test sets are shown in Table 5.1.

5.2.2 Preliminaries on the data

While all our models are trained on 'natural' data from OPUS, we use different types of data for evaluation: synthetic, semi-natural, and natural data. In the sections that follow, we use this data for the compositionality tests, but here, we first explain how the data was generated.

For our **synthetic** evaluation data, we consider the data generated by Lakretz et al. (2019), previously used to probe for hierarchical structure in neural language models. The data consists of sentences with a fixed syntactic structure and diverse lexical material. We extend the vocabulary and the templates used to generate the data and generate 3000 sentences for each of the resulting 10 templates (see Table 5.2).

In the synthetic data, we have full control over the sentence structure and lexical items, but the sentences are shorter (with an average of 9 source tokens vs 16 in OPUS) and simpler than typical in NMT data. To obtain more complex yet plausible test sentences, we employ a data-driven approach to generate **semi-natural** data. Using the tree substitution grammar Double DOP (Van Cranenburgh et al., 2016), we obtain noun and verb phrases (NP, VP) whose structures frequently occur in OPUS, as is detailed in Appendix C.1. We then embed these NPs and VPs in ten synthetic templates with 3000 samples each. In Table 5.3, we provide examples for each of the ten templates used, along with the internal structure of the complex NP or VP that is varied in the template. Instantiations of the NPs and VPs are drawn from the OPUS corpus, the synthetic words marked in blue come from the vocabulary used for the synthetic data, and the words marked in black remain unchanged.

Lastly, we extract **natural** data directly from OPUS, as will be further detailed in the sub-sections of the individual tests.

5.3 (Non-)compositional generalisation tests

We redefine three tests from Hupkes et al. (2018) for a setup in which models are trained on a regular, natural language NMT corpus. In this section, we go over the three tests, elaborating on the test design, evaluation metrics and results one by one.

5.3.1 Systematicity

One of the most commonly tested properties of compositional generalisation is **systematicity** – the ability to understand novel combinations made up from known components. In natural data, the number of potential recombinations to consider is infinite. We choose to focus on recombinations in two sentence-level context-free rules: `S → NP VP` and `S → S CONJ S`. Rather than quantifying systematicity as combinatorial generalisation using known components (which is highly dependent on the training set and the individual model evaluated, and is not as straightforward to measure in NMT), we focus on the *consistency* of translation under recombination. This is further elaborated on below.

Test design The first setup, `S → NP VP`, concerns recombinations of noun and verb phrases. We extract translations for input sentences from our generated synthetic and semi-natural data, as well as versions of them with the (1) noun (`NP → NP'`) or (2) verb phrase (`VP → VP'`) adapted. In (1), a noun from the `NP` in the subject position is replaced with a different noun while preserving number agreement with the `VP`. In (2), a noun in the `VP` is replaced. `NP → NP'` is applied to both synthetic and semi-natural data; `VP → VP'` only to synthetic data. We use 500 samples per template per condition per data type.

The second setup, `S → S CONJ S`, involves phrases concatenated using "and", and tests whether the translation of the second sentence is dependent on the first sentence.

S→S CONJ S

The girl sees that the men cry , and the poet criticises the king
S_1→S_1'
The girl sees that **the aunts** cry , and the poet criticises the king
S_1→S_3
The painter avoids the mayor , and the poet criticises the king

S→NP VP

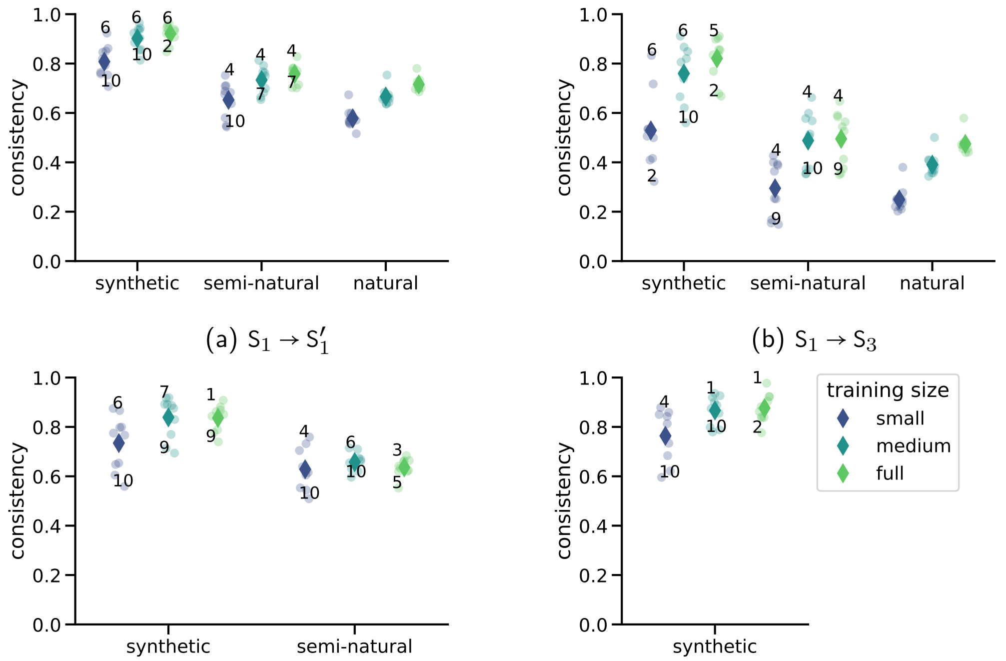
NP→NP'	*VP→VP'*
The girl sees that the men cry	The girl sees that the men cry
The baker sees that the men cry	The girl sees that **the aunts** cry

Figure 5.2: Illustration of the systematicity experiments S → S CONJ S and S → NP VP. Each experiment involves extracting translations before and after the replacement of the blue part, and then comparing the translation of the underlined words.

We combine two sentences (S_1 and S_2) from different templates, and we consider again two different conditions. First, in condition $S_1 \rightarrow S_1'$, we make a minimal change to S_1 yielding S_1' by changing the noun in its VP. In $S_1 \rightarrow S_3$, instead, we replace S_1 with a sentence S_3 that is sampled from a template different from S_1. We compare the translation of S_2 in all conditions. For consistency, the first conjunct is always sampled from the synthetic data templates. The second conjunct is sampled from synthetic data, semi-natural data, or from natural sentences sampled from OPUS with similar lengths and word frequencies as the semi-natural inputs. We use 500 samples per template per condition per data type. Figure 5.2 illustrates the different setups experimented with. The underlined words remain unchanged, and the data manipulation is marked in blue.

Evaluation In related work using synthetically generated data, systematicity is evaluated by leaving out combinations of 'known components' from the training data and using them for testing purposes (see §2.3.2). The necessary familiarity of the components (the fact that they are 'known') is ensured by high training accuracies, and systematicity is quantified by measuring the test set accuracy. If the training data is a natural corpus and the model is evaluated with a measure like BLEU in MT, this strategy is not available. We observe that being systematic requires being consistent in the interpretation assigned to a (sub)expression across contexts, both in artificial and natural domains. We, therefore, focus on **consistency** rather than accuracy, allowing us to employ a model-driven approach that evaluates the model's systematicity as the consistency of the translations when presenting words or phrases in multiple contexts.

We measure consistency as the equality of two translations after accounting for anticipated changes. For instance, in the S → NP VP setup, two translations are consistent if they differ in one word only, after accounting for determiner changes in Dutch ("de" vs "het"). In the evaluation of S → S CONJ S, we measure the consistency of the translations of the second conjunct. For the examples listed in Figure 5.2, this amounts to measuring the consistency of the translation of the underlined words.



(a) $S_1 \rightarrow S_1'$

(b) $S_1 \rightarrow S_3$

(c) $NP \rightarrow NP'$

(d) $VP \rightarrow VP'$

Figure 5.3: Systematicity results for setup S → S CONJ S (a and b) and S → NP VP (c and d). Consistency scores are shown per evaluation data type (x-axis) and training dataset size (colours). Datapoints represent templates (∘) and means over templates (⋄).

Results Figure 5.3 shows the results for the S → NP VP and S → S CONJ S setups. The average performance for the natural data closely resembles the performance on *semi-*natural data, suggesting that the increased degree of control did not severely impact the results obtained using this generated data. In general, the consistency scores are low, illustrating that models are prone to changing their translation of a (sub)sentence after small (unrelated) adaptations to the input. It hardly matters whether that change occurs in the sentence itself (S → NP VP), or in the other conjunct (S → S CONJ S), suggesting that the processing of the models is not local as assumed in strong compositionality. We will further elaborate on the types of inconsistencies observed in §5.4. Models trained on more data seem more locally compositional, a somewhat contradictory solution to achieving compositionality, which, after all, is assumed to underlie the ability to generalise usage from *few* examples (Lake et al., 2019). This trend is also at odds with the hypothesis that inconsistencies are a consequence of the natural variation of language, which models trained on *more* data are expected to better capture.

5.3.2 Substitutivity

Under the strict interpretation of the principle of compositionality, synonym substitutions should be meaning-preserving: substituting a constituent in a complex expression with a synonym should not alter the complex expression's meaning, or, in the case of MT, its

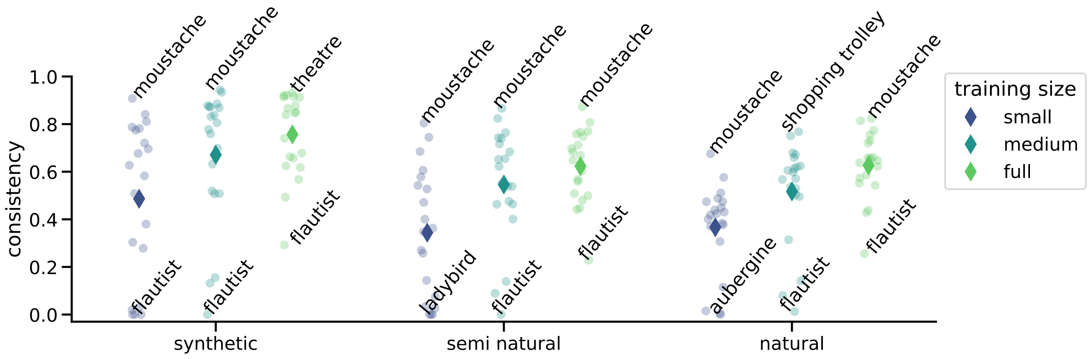
Synonym pair				**Subordinate clause**
British	*Freq.*	*American*	*Freq.*	
aeroplane	6728	airplane	5403	that travels by ...
aluminium	17982	aluminum	5700	that sells ...
doughnut	2014	donut	1889	that eats the ...
foetus	1943	fetus	1878	that researches the ...
flautist	112	flutist	101	that knows the ...
moustache	1132	mustache	1639	that has a ...
tumour	7338	tumor	6348	that has a ...
pyjamas	808	pajamas	1106	that wears ...
sulphate	3776	sulfate	1143	that sells ...
yoghurt	1467	yogurt	2070	that eats the ...
aubergine	765	eggplant	762	that eats the ...
shopping trolley	217	shopping cart	13366	that uses a ...
veterinary surgeon	941	veterinarian	6995	that knows the ...
sailing boat	5097	sailboat	1977	that owns a ...
football	33125	soccer	6841	that plays ...
holiday	125430	vacation	23532	that enjoys the ...
ladybird	235	ladybug	303	that caught a ...
theatre	19451	theater	13508	that loves ...
postcode	479	zip code	1392	with the same ...
whisky	3604	whiskey	4313	that drinks ...

Table 5.4: Synonyms for the substitutivity test, along with their OPUS frequency, Dutch translation, and the subordinate clause used to insert them in the data.

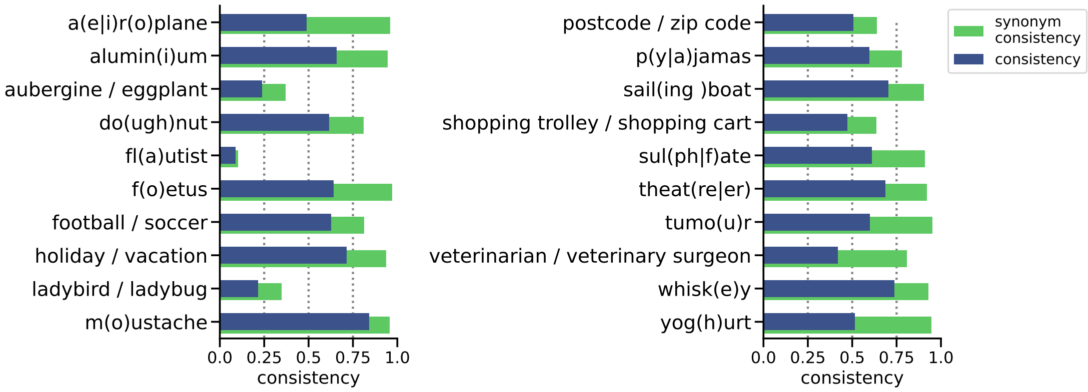
translation. Here, we test to what extent models' translations abide by this principle by performing the **substitutivity** test from Hupkes et al. (2020), which measures whether the outputs remain consistent after synonym substitution.

To find synonyms – source terms that translate into the same target terms – we exploit the fact that OPUS contains texts both in British and American English. Therefore, it contains synonymous terms that are spelt differently, such as "doughnut" and "donut", and synonymous terms with a very different form, such as "aubergine" and "eggplant". We use 20 synonym pairs in total (see Table 5.4).

Test design For each synonym pair, we select natural data from OPUS in which the terms appear and perform synonym substitutions. Thus, each sample has two sentences, one using the British English term and one using the American English term. We also insert the synonyms into the synthetic and semi-natural data using 500 samples per synonym pair per template, through subordinate clauses that modify a noun – e.g. "the king *that eats the doughnut*". Table 5.4 includes all clauses used.



(a) All consistency scores



(b) Consistency scores per synonym

Figure 5.4: (a) Consistency scores of synonyms (averaged ⋄, and per synonym ∘) for substitutivity per evaluation data type, for three training set sizes. (b) Consistency per synonym, measured using full sentences (in dark blue) or the synonym's translation only (in green), averaged over training dataset sizes and data types.

Evaluation Like systematicity, we evaluate substitutivity using the consistency score, expressing whether the model translations for a sample are identical. We report both the full sentence consistency and the consistency of the synonyms' translations only, excluding the context. Cases in which the model omits the synonym from both translations are labelled as consistent if the rest of the translation is the same for both input sequences.

Results In Figure 5.4a, we summarise all substitutivity consistency scores. We observe trends similar to the systematicity results: models trained on larger training sets perform better, and synthetic data yields more consistent translations compared to (semi-)natural data. We further observe large variations across synonyms, for which we further detail the performance aggregated across experimental setups in Figure 5.4b. The three lowest scoring synonyms – "flautist", "aubergine" and "ladybug" – are among the least frequent synonyms, which stresses the importance of frequency for the model to pick up on synonymy.

In Figure 5.4b, we show both the regular consistency and the consistency of the synonyms' translations, illustrating that a substantial part of the inconsistencies are due to varying translations of the context rather than the synonym itself, underscoring the models' non-local processing. We will further elaborate on the types of inconsistencies observed for both synonyms' translations and the translation of the contexts in §5.4.

5.3.3 Overgeneralisation

In our final test, we focus on exceptions to compositional rules. In natural language, typical exceptions that constitute a challenge for local compositionality are *idioms*. For instance, the idiom "raining cats and dogs" should be treated non-locally to arrive at its meaning of heavy rainfall. A locally compositional approach would yield an overly literal but nonsensical translation ("het regent katten en honden"). When a model's translation is too local, we follow Hupkes et al. (2020) in saying that it **overgeneralises**, or, in other words, it applies a general rule to an expression that is an exception to this rule. Overgeneralisation indicates that a language learner has internalised the general rule (e.g. Penke, 2012).

We select 20 English idioms for which an accurate Dutch translation differs from the literal translation from the English MAGPIE corpus (Haagsma et al., 2020). Because acquisition of idioms is dependent on their frequency in the corpus, we use idioms with at least 200 occurrences in OPUS based on exact matches, for which over 80% of the target translations do not contain a literal translation.

Test design Per idiom, we collect data from three sources: **natural** (sentences from OPUS that represent an exact match with the idiom's surface form as contained in MAGPIE), **semi-natural** and **synthetic**. For the latter two categories, we insert the idiom in 500 samples per idiom per template, by attaching a relative clause to nouns representing a human – e.g. "the king *that said 'I knew the formula by heart'*". The clauses themselves are drawn from source sentences in OPUS, and can be found in Table 5.5. In the third column of Table 5.5, we show idiom translations elicited from the model by embedding the idiom in a string of ten random nouns. Since idioms only receive their figurative meaning in a supportive context, we expect a literal, local translation in this scenario. Nearly all idioms are indeed locally translated, which indicates that the idiom is not memorised as one lexical unit per se, but that it is only translated non-locally in specific contexts.

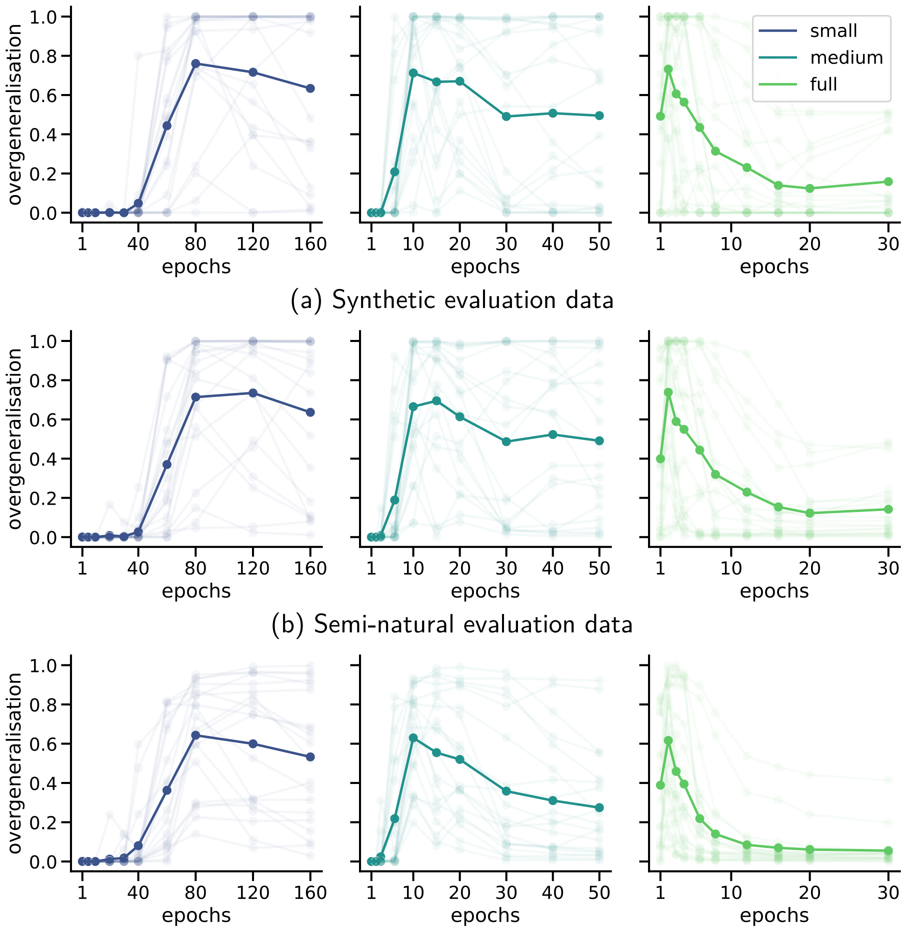
Evaluation Per idiom, we assess how often a model overgeneralises and how often it translates the idiom differently. To do so, we identify keywords that indicate that a translation is overgeneralised. If the keyword's literal translation is present, the whole translation is labelled as an overgeneralised translation. For instance, for "by heart",

Idiom	**Relative clause**	**Local translation**
once in a while	that said “ I will play it once in a while ”	eens in een tijdje
do the right thing	that said “ Just do the right thing ”	doen het juiste ding
out of your mind	that said “ Have you gone out of your mind ”	uit je hoofd
state of the art	that said “ This is a state of the art, official facility ”	stand van de kunst
from scratch	that said “ We are cooking from scratch every day ”	van kras
take stock	that said “ Take stock of the lessons to be drawn ”	nemen voorraad
across the board	that said “ I got red lights all across the board ”	aan boord
in the final analysis	that said “ In the final analysis, this is what matters ”	in de laatste analyse
out of the blue	that said “ It just came out of the blue ”	uit het blauwe
in tandem	that said “ We will work with them in tandem ”	in tandem
by heart	that said “ I knew the formula by heart ”	door hart
come to terms with	that said “ I have come to terms with my evil past ”	komen overeen met*
by the same token	that said “ By the same token I will oppose what is evil ”	bij dezelfde *token*
at your fingertips	that said “ The answer is right at your fingertips ”	binnen handbereik*
look the other way	that said “ We cannot look the other way either ”	kijken de andere manier
follow suit	that said “ And many others follow suit ”	volgen pak
keep tabs on	that said “ I keep tabs on you ”	houden *tabs*
in the short run	that said “ In the short run it clearly must be ”	in de korte lopen
by dint of	that said “ We are part of it by dint of our commitment ”	door de *int*
set eyes on	that said “ I wish I had never set eyes on him ”	*set* ogen op

Table 5.5: Idioms used in the overgeneralisation test, with the underlined words being indicative of a local translation. The relative clauses are used to insert idioms into synthetic and semi-natural templates. The local translation indicated is the translation given by the model when the idiom is embedded in a string of ten random words. * marks examples translated non-locally, and words that appear (partially) untranslated are italicised.

the presence of “hart” (“*heart*”) suggests a literal translation. An adequate paraphrase would say “uit het hoofd” (“*from the head*”). In Table 5.5, the keywords selected are underlined. We evaluate the overgeneralisation tendency as the fraction of inputs for which the translation contains the literal translation of the keyword, for ten checkpoints between the start of training and one of the last checkpoints the five seeds had in common before converging based on the development test’s BLEU scores (epoch 160, 50 and 30 for the small, medium and full training set sizes, respectively).

Results In Figure 5.5, we report the results. For all evaluation data types and all training set sizes, phases can be identified. Initially, the translations do not contain the idiom’s keyword, not because the idiom’s meaning is paraphrased in the translation, but because the translations consist of high-frequency words in the target language only. Very early on, for instance, during epoch one for the small and medium training set sizes, these are just strings of repeated words like “de” (“*the*”). Slightly later during training, the models will start to produce simple sentences that wrongly translate the



(a) Synthetic evaluation data

(b) Semi-natural evaluation data

(c) Natural evaluation data

Figure 5.5: Visualisation of overgeneralisation for idioms throughout training, with a line per idiom and the overall mean. Overgeneralisation occurs early on in training and precedes memorisation of idioms' translations. The colours indicate different training dataset sizes.

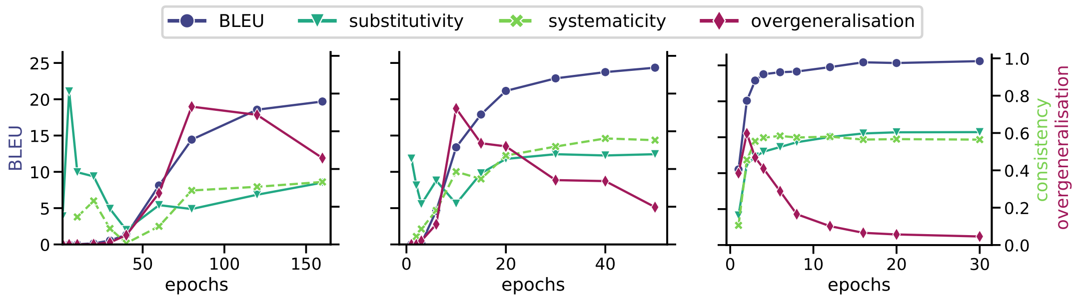
idiom or omit it – e.g. during epoch six when training on the medium training set size "Once in a while, a student surprises you" is translated as "Als je een student bent" ("*If you are a student*"), which is completely inaccurate and omits the idiom, but does contain some words from the source sentence. Afterwards, overgeneralisation peaks, when the models emit very literal, word-for-word translations of the idioms – e.g. when trained on the full corpus, the model translates "In the short run, it clearly must be" in epoch two as "In de korte loop moet het duidelijk zijn" ("*In the short run it should be clear*"), translating "run" literally as "walk". Finally, the model starts to memorise the idioms' atypical translations, as signalled by the absence of the literal translations of the keywords we marked. This is in accordance with overgeneralisation results on synthetic data (Korrel et al., 2019; Hupkes et al., 2020), and earlier results presented in a debate regarding human acquisition of irregularities of the past tense of verbs (Rumelhart and McClelland, 1986, i.a.). It also agrees with analyses of the different training stages of NMT systems presented by Voita et al. (2021), who identify that models first learn target-side language modelling (leading models to hallucinate frequent tokens), then move on to producing word-for-word translations and end with a final stage in which words in the translation are reordered, yielding more fluent translations.

If we now inspect the results for the different evaluation data types, we observe more overgeneralisation for synthetic and semi-natural data compared to natural data, suggesting the context in which an idiom is embedded somewhat influences how it is translated. At the same time, there are many idioms for which the natural contexts barely appear to matter, since for only 6, 10 and 8 idioms (for the full, medium and small training corpus, respectively) it holds that the semi-natural data is overgeneralised at least 10 percentage points more compared to the natural data at the end of training. Note that we embedded the idiom in the synthetic and semi-natural data using templates, constantly repeating the exact same idiomatic clause. Taking that into account, the similarity in the results between the natural data and the other two evaluation data types could signal inadequate disambiguation skills in addition to a lack of memorisation of non-local translations.

When analysing the results for the different training set sizes, overgeneralisation is more prominent in converged models trained on smaller datasets than in models trained on the full corpus. For the natural evaluation data, more than 5%, 25% and 50% of the examples are overgeneralised at the final checkpoint evaluated for the full, medium and small training set sizes, respectively, suggesting that the models trained on the medium and small data, in particular, are *too compositional* close to convergence. This provides an interesting contrast to the results from our substitutivity and systematicity results, where we observed processing that was *not as locally compositional* as expected. In the smaller training sets, the absolute frequency of the idioms changed. Still, the relative frequency was preserved, which could indicate that either absolute frequency or diversity of contexts in which idioms are observed influences models' tendency to memorise. Future work would have to investigate the training dynamics of idioms further to consolidate which factors are the main drivers of idiom memorisation here.

Going further: training dynamics across tests Finally, we would like to know to what extent idiom memorisation (mis)aligns with other performance indicators we measured. Figure 5.6 displays BLEU scores, consistency scores for systematicity (`S → S CONJ S`) and substitutivity, and overgeneralisation scores, using the natural data for one model seed.[3] Substitutivity consistency scores show increases prior to dropping as soon as the BLEU scores increase, due to highly consistent (but inaccurate) translations during the early epochs, when synonyms are simply omitted from the translation and the

[3]These models are reproduced following the publication of this chapter, due to a lack of access to all intermediate checkpoints of the main results we discussed. We provide the technical setup used for the reproduction in Appendix C.2.



(a) Small (b) Medium (c) Full

Figure 5.6: Demonstration of how overgeneralisation aligns with BLEU scores, and systematicity and substitutivity results, per training corpus size.

model produces input-agnostic translations.[4] As soon as the BLEU scores increase, the other metrics do too, with overgeneralisation peaking while the other metrics are still improving.

Relatively speaking, overgeneralisation changes the most when the other metrics are starting to converge, making it likely that this metric would benefit the most from continued training. The model trained on the full corpus is the only one that has fully stopped overgeneralising the translations of idioms. However, it should be taken into account that we purposefully selected the idioms to be frequent (appearing at least 200 times in the full corpus), and that memorisation may not have been as successful for less frequent idioms. In chapter 6 we will return to NMT systems' tendency to translate idioms overly literally, which is very prominent when examining a much wider range of idioms. Yet, also for the 'full' model, the overgeneralisation tendency still rapidly changes between, e.g., epochs 6 and 20, while much smaller changes occur for the BLEU scores. That BLEU scores do not always align well with other performance metrics has previously been reported – e.g. by Voita et al. (2019b) for contextual adequacy of translations in context-aware NMT, by Savoldi et al. (2022) for feminine grammatical gender markings, and by Stadler et al. (2021) for a wide range of linguistic phenomena.

We finally note that for the 'full' training set, the systematicity consistency peaks very early (epoch 6) and then slightly, but steadily, decreases. This score reflects the model's behaviour in the `S → S CONJ S` test and thus evaluates the consistency of a conjunct's translation following a modification of the *other* conjunct, reflecting local changes in the translation following a global modification to the input. Potentially, the abilities that underlie adequately translating global phenomena (such as idioms, or source-side phrases that require a lot of reordering in the target translation) are at odds with locally translating such conjuncts, demonstrating, again, a paradox between local and global processing. Local processing is desirable from a robustness point of view,

[4] Systematicity does not necessarily show the same pattern because the conjuncts cannot be reliably separated for `S → S CONJ S` during the early epochs.

but global processing can be needed for fluent, natural translations. We will return to this paradox in the discussion (§5.5).

5.4 Manual analysis

Our systematicity and substitutivity results demonstrate that models are not behaving compositional according to a strict definition of compositionality. However, we ourselves have argued that strict compositionality is not always appropriate to handle natural language. A reasonable question to ask is thus: are the inconsistencies we marked as non-compositional actually incorrect?

Annotation setup To address this question, we perform a manual analysis. We annotate 1800 inconsistent translation pairs from the systematicity and substitutivity tests to establish whether the inconsistencies are benign or concerning. For systematicity, we use 50 examples per model-data type combination for the $\mathtt{S}_1'$ and $\mathtt{S}_3$ conditions of the `S → S CONJ S` test. For substitutivity, we use 100 examples per model-data type combination, with equal representation of synonyms. We consider four types of inconsistencies:

1. cases of *rephrasing*, where both translations are equally (in)correct;
2. changes reflecting different interpretations of *source ambiguities*;
3. cases in which one of the two translations contains an *error*;
4. *formatting* (mostly punctuation) changes.

For substitutivity samples, we also annotate whether the changes are related to the translation of the synonym, where we distinguish cases where

i. one of the synonym translations is incorrect;
ii. both are incorrect but in a different manner;
iii. both are correct but translated differently;
iv. one synonym remains untranslated.

We annotate all changes observed per pair and report the relative frequency per class.

Results In the systematicity test, 40% of the inconsistencies reflect errors, whereas 38% contain examples of rephrasing, 16% reflect ambiguities in the source sentences and 6% are caused by formatting differences. The distribution of these types differs strongly per training corpus size; for models trained on fewer datapoints, inconsistencies are more likely to represent errors, whereas models trained on more data rephrase more often. For a breakdown per training corpus size and evaluation data type, and a more elaborate discussion of the results of the analysis, we refer to Appendix C.3.

Some target errors can be traced to individual words (e.g. because words are missing, wrongly translated or untranslated), while others reflect broader misinterpretations, such

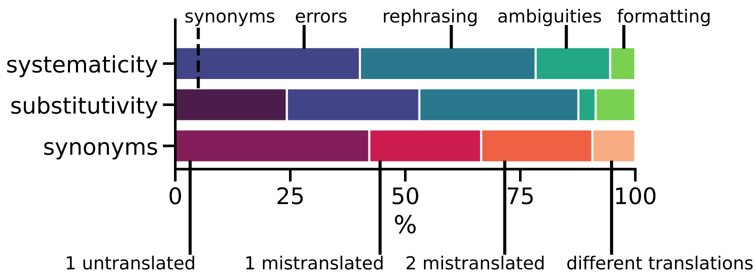


Figure 5.7: Relative frequencies of manually labelled inconsistencies in translations, averaged over data types and training set sizes. The 'synonyms' distribution further details the category 'synonyms' from row two.

as a change in subject-verb agreement displayed in Example (1). Here, "understands" ("*begrijpt*") should agree with "painter" ("*schilder*") but instead agrees with the word "doctors", much earlier in the sentence. A more locally compositional approach to translating would have yielded the correct translation.

(1) s EN: The doctors that laugh admire the {president, baker}, and the painter that admires her understands the king.
t_1 NL: (...) de schilder die haar bewondert, begrijpen de koning.
t_2 NL: (...) de schilder die haar bewondert begrijpt de koning.

Among the inconsistent translations due to ambiguities were also cases where our sentence conjunction was unintentionally ambiguous, if the verb of the first conjunct could take scope over the second conjunct – e.g. in "The friend wishes that the {lawyers, directors} scream, and the victims (...)". In Dutch, the unintended reading triggers SOV word order in the second conjunct, yielding inconsistent translations if the model switches its interpretation following our input perturbation. Such scope shifts often result in questionable interpretations, and models sometimes change the word order in the second conjunct even when a scope change is impossible.

In addition to pointing out errors, it should be noted that many inconsistencies appear benign, particularly for the 'rephrasing' category. A common type of rephrasing is a change in word ordering that does not affect the grammaticality or meaning of the Dutch sentence – e.g. for sentences with adverbs that can appear in various positions. We could not link these reorderings to the input perturbations we made in the systematicity test. See, for instance, Example (2), where reordering occurs after changing "king" to "father".[5] Yet, even benign rephrasings might be undesirable from a robustness and reliability perspective.

(2) s EN: The aunts criticise the {king, father}, and the man definitely observes the

[5] Even though these translations both contain an error ("neemt ... in de gaten"), this is not marked as an inconsistency, because it is shared between the translations.

mayor.

t_1 NL: (...) en de man neemt zeker de burgemeester in de gaten.

t_2 NL: (...) en de man neemt de burgemeester zeker in de gaten.

For substitutivity, most inconsistencies are similar to the ones observed in systematicity: only 24% of the inconsistencies involve the synonyms' translations, where one of the synonyms being untranslated was the most frequent. The inconsistencies can be quite peculiar, e.g. in case of "donut" being translated as "ezel" ("*donkey*"), or the global change in the following example:

(3) s EN: - Yeah, a barbecue sauce {moustache, mustache} contest.

t_1 NL: - Ja, een barbecue met snor. (*missing 'sauce' and 'contest'*)

t_2 NL: - Ja, een barbeceu saus snor wedstrijd.

How frequently each of the four subtypes of synonym inconsistencies occurs depends on the synonym. Where some synonyms are more prone to being untranslated (like "ladybird" and "flautist"), some simply received many different correct translations (like "shopping trolley"), and others received errors very specific to the synonym (like "eggplant" being translated as "egg"+"plant", an interesting case because it reflects processing that is too local). It should be noted that for all synonyms[6] we have observed correct translations, indicating that the models did in fact acquire their meaning. The majority of the substitutivity inconsistencies observed did not appear to be related to the synonyms themselves, and the types of rephrasings also did not reflect the writing style of the sentence. Considering that the synonym changes were related to British and American spelling, and may have changed the tone of the sentence (e.g. "aeroplane" could be considered more archaic compared to "airplane"), one could anticipate changes in word choice in Dutch reflecting this change of style. However, the substitutivity inconsistencies were virtually indistinguishable from those annotated for systematicity.

5.5 Conclusion and discussion

Whether neural networks can generalise compositionally is often studied using artificial tasks that assume strictly *local* interpretations of compositionality. We argued that such interpretations exclude large parts of language and that to move towards human-like productive usage of language, tests are needed that assess how compositional models trained on *natural data* are.[7] We laid out reformulations of three compositional generalisation tests – systematicity, substitutivity and overgeneralisation – for NMT models trained on natural corpora, and assessed models trained on different amounts

[6] Apart from the model with the small training dataset that cannot translate "flautist" and "ladybug".

[7] Dupoux (2018) makes a similar point for models of language acquisition, providing several concrete examples where using less than fully complex data proved problematic.

of data. Our work provides an empirical contribution but also highlights vital hurdles to overcome when considering what it means for models of natural language to be compositional. Below, we reflect on these hurdles and our results.

The proxy-to-meaning problem Compositionality is a property of the mapping between the form and meaning of an expression. Since translation is a *meaning-preserving* mapping from form in one language to form in another, it is an attractive task to evaluate compositionality (as we previously reviewed in §2.3.1), since the translation of a sentence can be seen as a proxy to its meaning. However, while expressions are assumed to have only one meaning, translation is a *many-to-many* mapping. The same sentence can have multiple correct translations, and the translation is *only* a proxy to the meaning, not the meaning itself. This not only complicates evaluation – MT systems are typically evaluated with BLEU because accuracy is not a suitable option – it also raises questions about how compositional the desired behaviour of an MT model should be. On the one hand, one could argue that for optimal generalisation, robustness, and accountability, we like models to behave systematically and consistently; it would make them more robust with regards to, for example, adversarial attacks based on input perturbations (e.g. Cheng et al., 2019; Zhang et al., 2021). We expect the translations of expressions to be independent of unrelated contextual changes that do not affect their meaning (e.g. swapping out a synonym). Additionally, model performance could be improved if small changes do not introduce errors in unrelated parts of the translation. On the other hand, non-compositional behaviour is not always incorrect – it is one of the main arguments in our plea to test compositionality 'in the wild' – and we observe that indeed, not all non-compositional changes alter the correctness of the resulting translations. Changing a translation from "atleet" ("*athlete*") to "sporter" ("*sportsman*") based on an unrelated word somewhat far away may not be (locally) compositional, but is it a problem? And how do we separate harmful mistakes from helpful or benign ones?

The locality problem Inextricably linked to the proxy-to-meaning problem is the locality problem. In our tests, we see that *small, local source changes* elicit *global changes in translations*. For instance, in our systematicity tests, changing one noun in a sentence elicited changes in the translation of a sentence that it was conjoined with. In our substitutivity test, even synonyms that merely differed in spelling (e.g. "doughnut" and "donut") elicited changes to the remainder of the sentence. This counters the idea of compositionality as a means of productively reusing language: if a phrase's translation depends on (unrelated) context that is not in its direct vicinity, this suggests that more evidence is required to acquire the translation of this phrase. At the same, our overgeneralisation experiments using idioms demonstrated a use case in which more global changes following a local modification are actually *desirable*: when comparing a phrase like "state of the world" to "state of the art", a single token difference implies that multiple tokens should change in the translation. Moreover, when working with natural data, not all occurrences of idioms are assumed to be idiomatic; occasionally, "state of the art" *is* used literally (e.g. in "The teacher judged the state of the art her students had created during the pottery workshop") in which case more evidence than just the phrase itself should be used to arrive at the right translation. This underscores the fact that, depending on the context, either local or global behaviour may be preferable.

Tests involving synthetic data present the models with sentences in which maximally local behaviour is possible, and we argue that it is, therefore, also desirable. Our experiments show that even in such setups, models do not translate in a local fashion: with varying degrees of correctness, they frequently change their translation when we slightly adapt the input. On the one hand, this well-known *volatility* (see also Fadaee and Monz, 2020) might be essential for coping with ambiguities for which meanings are context-dependent. On the other hand, our manual analysis shows that the observed non-compositional behaviour does not reflect the incorporation of necessary contextual information and that oftentimes it is even altering the correctness of the translations. Furthermore, this erratic behaviour highlights a lack of default reasoning, which can, in some cases, be problematic or even harmful, especially if faithfulness (Parthasarathi et al., 2021) or consistency is important.

In linguistics, it has been discussed how to extend the syntax and semantics such that 'problem cases' can be a part of a compositional language (e.g. Westerståhl, 2002; Pagin and Westerståhl, 2010b). In such formalisations, global information is used to disambiguate the problem cases, while other parts of the language are still treated locally. In our models, global behaviour appears in situations where a local treatment would be perfectly suitable and where there is no clear evidence for ambiguity. At the same time, local behaviour occurs at the phrase-level when idioms are overgeneralised, or even at the word-level in the case of "eggplant" being translated as "egg plant". We follow Baggio (2021) in suggesting that we should learn from strategies employed by humans, who can assign compositional interpretations to expressions but can, for some inputs, also derive non-compositional meanings. For *human-like* linguistic generalisation, it is vital to investigate how models can represent both these types of processing, providing a locally compositional treatment when possible and deviating from that when needed.

Conclusion In conclusion, with this work, we contribute to the question of how compositional models trained on *natural* data are, and we argue that MT is a suitable and relevant testing ground to ask this question. Focusing on the balance between *local* and *global* forms of compositionality, we formulate three different tests and discuss the issues and considerations that come up when studying compositionality in the context of natural data. Our tests indicate that models show both local and global processing, but not necessarily for the right samples, and that local and global processing emerge at different rates during training. Furthermore, our tests underscore the difficulty of separating helpful and harmful types of non-compositionality, stressing the need to rethink the evaluation of compositionality using natural language, where composing meaning is not as straightforward as in synthetic datasets that treat language like arithmetic.

5.5.1 Limitations

We identify four main limitations of our work. Firstly, while the majority of the work we introduced in §2.3.2 focuses on evaluating models with an explicit distribution shift between the training and test data – systematically introducing new token combinations, syntactic structures or sentence lengths in the test set – constructing **an explicit distribution shift is not the main focus of our work**. Although we partially evaluate with data the models have not seen before, we also conduct evaluations using phrases and sentences from the training corpus (in case of the semi-natural and natural evaluation subsets). We do combine those phrases and sentences with new material (e.g. through a conjunction operation for systematicity, or changing a synonym for substitutivity), which one could argue makes the evaluations OOD.

A second limitation is that, although we urge others to rethink the evaluation of compositionality for natural language, we do **adopt partially synthetic datasets** in our evaluations, because they are easier to manipulate in a controlled manner. Ideally, we would move towards scenarios where both the training and the evaluation data are fully natural. Models trained with natural data might behave differently when evaluated with synthetic data than with natural data. Moreover, our synthetic data yields constructions that are, on occasion, semantically or grammatically odd, which could, again, lead to non-standard behaviour from the models.

Thirdly, our experimental setup is inherently limited by focusing on **one architecture** (`transformer-base`, Vaswani et al., 2017), **one language pair**, **three types of tests** and **one task**. Compositionality is a multi-faceted phenomenon, and how to evaluate whether models truly generalise compositionally is not easily captured in a few tests. Other evaluations to consider could study generalisation of primitives to larger contexts (e.g. Lake and Baroni, 2018), generalisation to new syntactic constructions (e.g. Kim and Linzen, 2020) or generalisation based on corpus-level distribution shift metrics (e.g. Keysers et al., 2019). Similarly, the non-compositional aspects of natural language extend beyond the idioms assessed in our overgeneralisation test, and further evaluations could have been performed, e.g. using proverbs or non-compositional compounds, as we did in §3.4. We chose `transformer-base` as it has been the predominant architecture used in open-source pretrained translation systems since transformer's introduction in 2017, and adopted one language pair due to the intricacies involved with both producing the data and accurately evaluating the results. In the evaluation, we had to account for language-specific properties such as not penalising determiner changes following noun modifications on the source side, and carefully selecting keywords in the overgeneralisation test, for which the knowledge of native speakers is crucial.[8] Our analyses focused exclusively on the task of NMT, but because human translations can demonstrate wide variability – making MT a many-to-many problem – one could argue that robustness and consistency are even more important in other tasks, e.g. for hate speech detection or code generation. We chose to evaluate compositional generalisation within NMT, because compositionality is traditionally well-studied and -motivated for MT, and because MT corpora represent the richness of natural language when it comes to the compositionality continuum. We do not necessarily suggest that MT should become the de facto standard for compositional generalisation evaluation, and encourage explorations similar to ours for other tasks, but would like to underscore that *natural* language should receive more attention in the discourse around compositional generalisation.

Lastly, while we pointed out that transformer can be too compositional (when overgeneralising idioms) and not compositional enough at the same time (for the substitutivity and systematicity tests), we **did not offer solutions** to mitigate this. However, we can point out modifications to explore based on our results. Firstly, since corpus size has a clear impact on consistency scores, data augmentation techniques will likely provide additional improvements in score. Secondly, we noticed that perturbing an input token in one conjunct can affect the translation of a token in the other conjunct. This could be due to the fact that vanilla transformers' self-attention blocks include a softmax over *all* input tokens. Architectural changes, such as sparse attention (e.g. Correia et al., 2019), could partially alleviate the over-reliance on irrelevant context. Thirdly, that models are too compositional and not compositional enough all at once might be because we require all inputs to be processed by the same model weights. More experimental approaches to balancing compositional generalisation and non-compositional memorisation could tailor architectures for this, e.g. by introducing Mixture-of-Experts layers (Shazeer et al., 2017) and biasing routing for sentences for which we know a non-compositional phrase is present. In the next subsection, we will elaborate on others who built upon our findings to mitigate some of the models' issues we pointed out, and in chapter 6 we will point out methods that have been proposed to improve models' translations for idioms.

[8] Two of the authors of Dankers et al. (2022a) are native speakers of Dutch.

5.5.2 Retrospective and outlook

Following the publication of this chapter, many new modelling techniques to improve compositional generalisation have been published. Two of these techniques are particularly related to the issues we pointed out regarding the inconsistency of translations in our substitutivity and systematicity tests:

- **On consistency regularisation**: Yin et al. (2023) introduce two regularisation objectives applied during the training of sequence-to-sequence models. The first promotes consistency of token representations across contexts via a contrastive loss based on cosine similarity. The second enforces similarity between output distributions when the same input is passed through the model under different dropout perturbations, using a Jensen-Shannon divergence loss. These auxiliary loss functions not only improve consistency in our substitutivity and systematicity tests, but also yield improvements in test set BLEU scores and enhance performance on a separate translation dataset (Li et al., 2021a) as well as two compositional generalisation semantic parsing datasets (Kim and Linzen, 2020; Keysers et al., 2019).
- **On regularising through joint dropout**: Niculae and Monz (2023) propose the joint dropout technique that replaces aligned phrase pairs in the source and target sentences with variables, while encouraging models to keep the translation of the remaining text the same. Joint dropout improves the consistency on our systematicity test, as well as BLEU scores, robustness scores and cross-domain generalisation on other datasets, particularly for low-resource languages.

These findings underscore that the inconsistencies we observed are not specific to our setup but more broadly undermine models' compositional abilities, and that reducing models' volatility improves robustness and generalisation. It should also be mentioned, however, that the same volatility and over-reliance on context can serve beneficial purposes in natural language translations, as pointed out by Sharma et al. (2022), who introduce context to reduce gender bias in NMT systems. This apparent paradox of volatility and over-reliance on context being both harmful and beneficial makes one question what it is we desire most for machine-translated text, particularly in an era in which more and more LLMs are used for NMT, and translations are more often sampled rather than being computed with beam search. Perhaps the inconsistencies we penalised models for in our work are completely unavoidable and unproblematic in such scenarios, and simply demonstrate a human-like creative ability to paraphrase? Perhaps, but only *if* it concerns benign rephrasings. If inconsistencies introduce errors, being consistent seems more desirable than being creative. The fact that inconsistencies were less often benign for our smaller training sets, and the fact that Niculae and Monz's regularisation also improves cross-domain generalisation, demonstrate the relevance of consistency when the training data is not fully representative of the data distribution we are interested in generalising towards. As such, it seems impossible to definitively answer whether consistency is always an optimal translation strategy for LLMs. This is simply dependent on the circumstances and application involved.

In addition to new modelling techniques being proposed following this chapter's publication, new datasets have appeared. We redefined three tests proposed by Hupkes et al. (2020) for the scenario of NMT systems trained on natural language, and Liu (2022) directly adopted our systematicity tests and constructed evaluation data accordingly for English-Tamil, English-Gujarati and English-German translation. Two other related articles by Li et al. (2024a) and Liao et al. (2023) adopt our tests but redefine them for the task of visual question-answering using natural language. Li et al. improve substitutivity capabilities of vision models by performing synonym replacements of both words and objects in the images in training examples. Liao et al. create evaluation sets for the notions of systematicity (by combining concepts previously not seen together in test images), substitutivity (by evaluating that an object remains the same when combined with different attributes such as different colours) and non-compositional testing (by combining objects in images that are usually not composed, such as 'door' and 'shirt').

Other MT datasets for compositional generalisation that go beyond the tests we examined were presented by Kumon et al. (2024) and Moisio et al. (2023). Kumon et al. noted that most evaluations focused on lexical generalisation (i.e. presenting words in new contexts in the evaluation data) and, therefore, present an English-Japanese dataset for *structural* generalisation (i.e. presenting new syntactic constructions in the evaluation data). Instead of singling out specific tokens or syntactic constructions that are novel during evaluation, Moisio et al. take a distribution-based approach by creating data splits for four language pairs by maximising the compound divergence as defined by Keysers et al. (2019).

Our work did not merely mean to point out NMT systems' inadequacies when it comes to compositional generalisation or idiom memorisation, but also presented a call to action: as a community, we should rethink how we evaluate compositional generalisation, and we cannot keep removing natural language variation to make evaluation more convenient. This call has been heard and is often explicitly mentioned in articles building upon our work (e.g. Zheng and Lapata, 2023; Sun et al., 2023; Chia, 2024; Chia et al., 2024; Fodor et al., 2025). Not only has there been increased attention for natural language's variable nature in compositional generalisation literature, graded notions of compositionality have also been studied more widely for LMs. Liu and Neubig (2022) measured to what extent LMs' hidden representations of a sequence are compositions of representations of subphrases. Although the mapping between subphrases and parent phrases could be accurately described with a compositional function, the compositionality scores of representations did not align with human compositionality judgements. This could suggest that the scores inadequately capture non-compositional meaning combinations. In Dankers and Titov (2022) and Dankers and Lucas (2023)[9] my co-authors and I quantified compositionality in the context of sentiment analysis using graded metrics for models' hidden representations and human compositionality judgements, respectively. We demonstrated that the more non-compositional an example is, the more challenging it is for LMs to predict the sentiment accurately. Although we acknowledged that compositionality exists along a continuum in natural language (§2.3), compositional generalisation evaluation at present fails to address that full, graded continuum. I encourage future work to expand the scope of (non-)compositional generalisation tests, instead of studying compositionality or non-compositionality in isolation.

[9]I was the first author of these articles, but they are not a part of this thesis.

Chapter 6

The mechanisms behind idiom processing

6.1 Introduction

Now that we have been introduced to models' struggles to balance compositional generalisation with non-compositional processing, and have learnt idiom acquisition to be a two-step process in NMT, we zoom in on idiomatic expressions. They have been a pain in the neck of NLP researchers for as long as the field has existed (e.g. Sag et al., 2002; Rayson et al., 2010; Shwartz and Dagan, 2019) but have been problematic for NMT systems, in particular (e.g. Barreiro et al., 2013; Isabelle et al., 2017; Constant et al., 2017; Avramidis et al., 2019). Idioms such as "kick the bucket" occur much less frequently than their parts (i.e. "kick", "the", and "bucket"), and they require disambiguation before translation. After all, not all *potentially idiomatic expressions* (PIEs) are figurative – e.g. consider "When I kicked the bucket, it fell over" – so whether PIEs should receive a figurative or literal translation depends on the context.[1] Not only do NMT systems need to acquire those disambiguation skills, they also need to memorise adequate translations for idioms, and to translate an individual idiom correctly, NMT systems need to be exposed to a sufficient number of example translations. For widely used NMT corpora, it is unknown to what extent they include sufficient training material to learn idioms, and NMT corpora dedicated to idiom learning are rare (Fadaee et al., 2018).

Although it is known that idioms pose a challenge to NMT systems, little is known about the neural mechanisms enabling idiomatic translations and methods for improving them. Related work on neural mechanisms for idiom processing that appeared prior to the work this chapter is based on mainly studied how idioms are represented by

[1] Up until now, we have simply spoken of 'idioms', but since this chapter discusses both literal and figurative occurrences, we will primarily refer to them as PIEs.

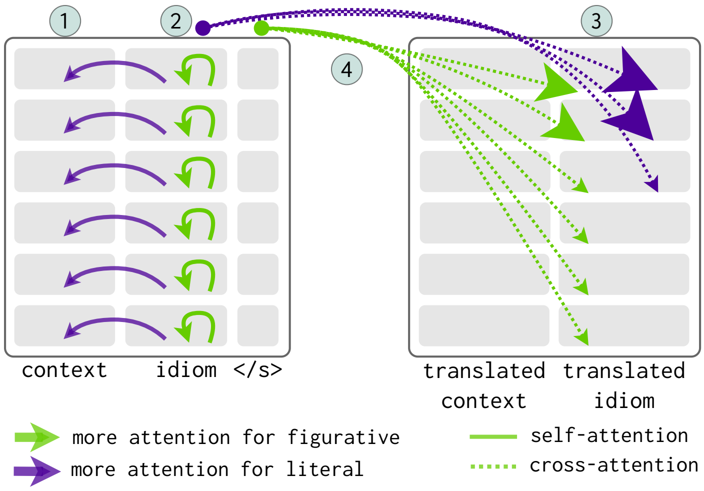


Figure 6.1: How do attention patterns of figurative PIEs that are paraphrased by the model compare to attention patterns of literal PIEs that are translated word for word? We find (1) decreased interaction between the PIE and its context, (2) increased attention within the PIE, (3) decreased cross-attention between the PIE and its paraphrase, (4) increased cross-attention from the paraphrase to the EOS token (</s>).

transformer-based LMs (e.g. García et al., 2021a,b), but LMs are not required to output a discrete representation of the idiom's meaning, which is a complicating factor for NMT models. We previously reviewed related work in this direction in §2.3.3, along with background information on the non-compositional nature of idioms and what is known about how humans process idioms (§2.3.1).

In this chapter, we analyse idiom processing for pretrained NMT transformer models (Vaswani et al., 2017) for seven Indo-European language pairs by comparing literal and figurative occurrences of PIEs. We focus on figurative PIEs that the model paraphrases, in particular, because those signal that the models have memorised to produce non-compositional translations. By doing so, we contribute to answering RQ2: "*Which model-internal mechanisms enable memorisation?*" Throughout this chapter, we address the following sub-questions:

1. *How can we perform analyses of NMT idiom processing at scale?* Large-scale analyses of idiom translations suffer from a lack of parallel corpora (Fadaee et al., 2018). We, therefore, use a monolingual corpus, heuristically label transformer's translations, and verify the heuristic works as intended through human evaluation, as described in §6.2.
2. *How does idiomaticity and the paraphrasing of non-compositional idioms affect attention patterns and hidden representations?* To understand how idioms are represented in transformer, we apply interpretability techniques to contrast the effect of literal and figurative PIEs on the encoder's self-attention and the decoder's cross-attention (§6.3), and the encoder's hidden representations (§6.4 and §6.5).

3. *How do encoder-internal interventions affect non-compositional translations?* Finally, in §6.5, we intervene in the encoding of the English PIEs to show that one can change non-compositional translations into compositional ones, and we comment on how this affects the attention patterns.

After elaborating on our experimental results, we end the chapter with a discussion of our findings (§6.6), commenting on the limitations of our approach and relevant work that appeared after the publication of this chapter. Overall, we find that transformer NMT systems typically translate idioms in a manner that is too compositional, providing word-for-word translations, signalling a lack of memorisation. The analyses of the attention patterns and hidden representations point to the encoder as the mechanism that groups words within figurative PIEs. The grouping manifests through increased attention within the PIE and reduced attention to the context. When translating figurative PIEs, the decoder relies less on the encoder's output than for literal PIEs, directing more attention to the EOS token. These patterns are stronger for figurative PIEs that the model paraphrases than for word-for-word PIE translations and hold across the seven language pairs. Figure 6.1 visually summarises these findings.

6.2 Experimental setup and heuristic annotation method

We use pretrained transformer NMT models (Vaswani et al., 2017) with English as the source language and one of seven languages as the target language (Dutch, German, Swedish, Danish, French, Italian, Spanish).[2] The models are `transformer-base` models, containing encoders and decoders with six layers each. The models are pretrained by Tiedemann and Thottingal (2020) with the Marian-MT framework (Junczys-Dowmunt et al., 2018) on a collection of corpora (OPUS) (Tiedemann and Thottingal, 2020). We extract hidden states and attention patterns for sentences with PIEs. The analyses presented are detailed for Dutch, after which we explain how the results for the other languages compare to Dutch.[3]

Parallel PIE corpora are rare, exist for a handful of languages only, and are limited in size (Fadaee et al., 2018). Rather than rely on a small parallel corpus, we use the largest corpus of English PIEs at the time of the publication of this chapter and annotate the translations heuristically. This section provides corpus statistics and discusses the heuristic annotation method.

MAGPIE corpus The MAGPIE corpus presented by Haagsma et al. (2020) contains 1756 English idioms from the Oxford Dictionary of English with 57k examples of PIEs

[2]Consistent with chapter 3, our figures refer to these languages using their ISO 639-1 codes, that are `nl`, `de`, `sv`, `da`, `fr`, `it` and `es`, respectively.

[3]We provide more details on the models, the data and the technical setup in Appendix D.1.

occurring in different contexts, including both figurative and literal contexts. Consider, for instance, the following four examples:

- "(...) go home, you bum, go home,' sang another to the tune of Auld Lang Syne."
- "(...) Froggy was richer than we all imagined: to the tune of five thousand pounds."
- "(...) their samples under the microscope, stained to pick up particles of iron (...)"
- "Indeed, one issue that has come under the microscope is Thatcher's reforms (...)"

MAGPIE contains identical PIE matches and morphological and syntactic variants, through the inclusion of common modifications of PIEs, such as passivisation ("the beans are spilled") and determiner changes ("spill some beans"). We use 37k samples annotated as fully **figurative** or **literal**, for 1482 idioms that contain nouns, numerals or adjectives that are colours (which we refer to as **keywords**). Because idioms show syntactic and morphological variability, we will mostly rely on noun translations to heuristically label models' translations (as detailed below) and to analyse models' behaviour. Verbs and their translations are harder to identify due to the variability. Moreover, idiom indexes are also typically organised based on the nominal constituents (e.g. Piirainen, 2012). Although MAGPIE includes examples of PIEs within context windows of five sentences, we will only present the PIE and its sentential context to the models. We distinguish between PIEs and their context using the corpus's word-level annotations.

Heuristic annotation method The MAGPIE sentences are translated by the models with beam search and a beam size of five. The translations are labelled heuristically. In the presence of a literal translation of at least one of the idiom's keywords, the entire translation is labelled as a **word-for-word** translation, where the literal translations of keywords are extracted from the model and Google Translate, yielding a set of possible translations. When a literally translated keyword is not present, it is considered a **paraphrase**.[4] We used a similar annotation method in chapter 5 for 20 English idioms, and Shao et al. (2018) previously analysed NMT translations of 50 Chinese idioms using a similar method, using manually curated lists of literal translations of idioms' words to detect literal translation errors.

Table 6.1 summarises the distribution of these categories for all languages, for the subsets of figurative and literal examples from MAGPIE. Generally, paraphrased translations of figurative PIEs are more appropriate than word-for-word translations, whereas literal PIEs can be translated word for word (Baker et al., 1992). The vast majority of literal PIEs indeed result in word-for-word translations. The subset of figurative samples results in more paraphrases, but word-for-word translations dominate with $\geq 76\%$. Although the statistics are similar across languages, there are differences in

[4] The annotation does not evaluate whether paraphrases are correct, which requires expert idiom knowledge in both languages. A paraphrase being provided is a first step to adequately translating idioms and, at present, the only way to detect how the model approaches the task for large datasets.

Category	nl	de	sv	da	fr	it	es
Figurative, paraphrase	20	20	24	18	19	20	24
Figurative, word for word	80	80	76	82	81	80	76
Literal, paraphrase	5	6	8	5	7	9	7
Literal, word for word	95	94	92	95	93	91	93

Table 6.1: Distribution of the heuristically assigned labels for translations of MAGPIE sentences in percentages, expressed within the categories of figurative and literal.

Category	#	nl	de	sv	da	fr	it	es
Figurative, paraphrase	116	88	84	75	81	78	78	87
Figurative, word for word	103	95	92	95	74	96	97	82
Literal, paraphrase	28	54	71	43	82	43	32	50
Literal, word for word	103	98	89	97	89	98	100	94

Table 6.2: Survey statistics: the number of sentence pairs used (#), and the percentage of labels for which the annotator and the algorithm agreed per language.

which examples are paraphrased. Figure 6.2 illustrates the agreement by computing the F_1-score when using the predictions for figurative instances of one language as the target, and comparing them to predictions from another language. The agreement positively correlates with genetic similarity as computed using the Uriel database (Littell et al., 2017) (Pearson's r=0.61, $p < 0.005$).

To assess the quality of the heuristic method, one (near) native speaker per target language annotated 350 samples, where they were instructed to focus on one PIE keyword in the English sentence. Annotators were asked whether (1) the English word was present in the translation (initially referred to as 'copy'), (2) whether there was a literal translation for the word, or (3) whether neither of those options was suited, referred to as the 'paraphrase'.[5] Due to the presence of cognates in the 'copy' category, that category was merged with the 'word for word' category after the annotation. Table 6.2 summarises the accuracies obtained; the heuristic and human annotations have a Cohen's κ of 0.703±0.05. Of particular interest are examples that are figurative and paraphrased, since they represent non-compositional translations, and examples that are literal and translated word for word, since they represent the compositional translations for non-idiomatic PIEs. These categories have annotation accuracies of

[5]Annotators were not involved in the research. Except for Swedish, annotators were native in the target language only. Note that the data annotation study is aimed at assessing the accuracy of the heuristic, and not at fully evaluating the idiomatic translations, for which annotators would have needed native knowledge of English. We, therefore, ask the annotators to perform the same task as the heuristic annotation method by focusing on the keyword. For ethical considerations and more details, see Appendix D.2.

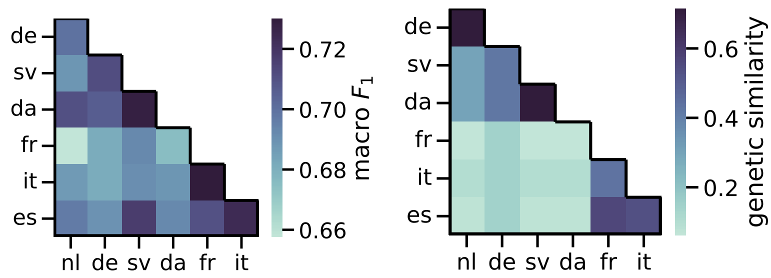


Figure 6.2: The macro-averaged F_1-score of translation labels (paraphrase vs word for word) for figurative PIEs and languages' genetic similarity visualised.

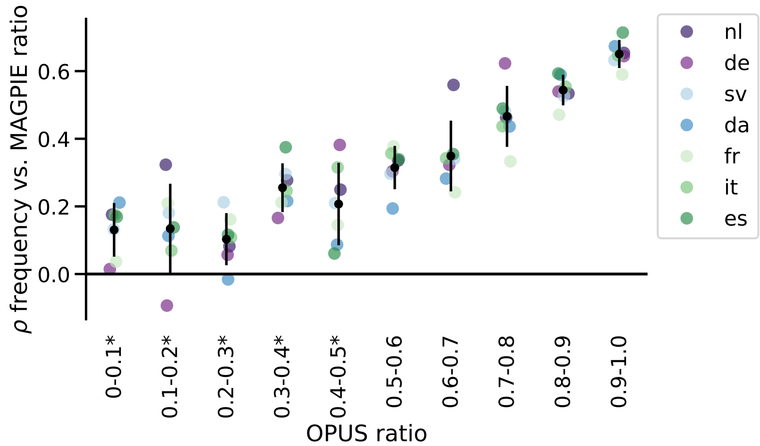


Figure 6.3: Spearman's ρ for the frequency of PIEs' paraphrased translations in OPUS and the ratio of paraphrased examples for figurative PIEs from MAGPIE (y-axis), per the ratio as observed in OPUS (x-axis). The means and standard deviations over language pairs are shown in black. * marks ratio bands for which $p > 0.05$ for at least three language pairs.

$\geq 75\%$ and $\geq 89\%$, respectively. During preliminary analyses, an annotation study was conducted for Dutch by annotators from the crowd-sourcing platform Prolific. The annotators and the heuristic method agreed in 83% of the annotated examples, and for 77% of the samples, an average of 4 annotators agreed on the label unanimously (see Appendix D.2 for more details). Together, the two annotation studies suggest that while the heuristic annotation method is imperfect, it is accurate enough for us to analyse group-level behaviour for the different categories (with the exception of the literal-paraphrase subset, representing only a small portion of the overall dataset).

Frequency effects Whereas a word-for-word translation of an idiom can be learnt without ever observing the idiom in the training corpus, non-compositional paraphrased translations are presumably only possible if the model has seen examples and memorised them. The frequency of the PIEs and paraphrased translations in the NMT training corpus is thus assumed to influence how these pretrained NMT models translate PIEs. We do not have access to the exact version of OPUS used by Tiedemann and Thottingal

OPUS		**Predicted Translations**	
		Paraphrase	*Word for word*
- Target translation type frequency (%)			
Paraphrase	54	49	51
Word for word	46	7	93
- BLEU scores			
Paraphrase		27.2	19.9
Word for word		25.6	38.2
- COMET scores			
Paraphrase		75.1	66.3
Word for word		73.5	80.1

Table 6.3: Distribution of translation labels for PIE occurrences in OPUS for En-Nl, along with the corresponding BLEU and COMET scores.

(2020) since OPUS is a continually growing online corpus. However, we can approximate the frequencies of PIEs by using an OPUS subset released around the same time by Tiedemann (2020). We collect source-target pairs for our seven language pairs containing exact matches of the MAGPIE PIEs and apply the heuristic annotation method to this data, restricting the data to PIEs with at least five matches in OPUS and five figurative examples in MAGPIE. For figurative MAGPIE examples, we measure the ratios of examples paraphrased per PIE, per language pair. If we first measure the correlation between those ratios and the frequency of paraphrased translations in OPUS, this yields a Spearman's ρ of 0.44 ($\pm$0.03 across the different language pairs, with p extremely close to 0), indicating a moderate frequency effect. Secondly, we subdivide PIEs based on the ratio of examples paraphrased in OPUS, since it is likely that occurrences of both paraphrased examples and word-for-word counterexamples influence how models translate PIEs for the unseen MAGPIE data. Figure 6.3 shows that as a PIE's ratio of paraphrased translations in OPUS increases, so does the correlation between the MAGPIE ratio and paraphrased translation frequency. Our analysis is an imperfect representation of the frequency of PIEs in the training set since we lack access to the actual training set, and only rely on exact matches. In spite of this, these results already show the effect of frequency and demonstrate that this effect is modulated by the extent to which the training material reflects the non-compositional nature of idioms.

Translation quality Sentences containing idioms typically yield lower BLEU scores (Fadaee et al., 2018). MAGPIE is a monolingual corpus and does not allow us to compute BLEU scores, so to gain some understanding of whether the model's translations reflect target translations from its training corpus, we use the same OPUS subset mentioned above to extract up to 500 source-target pairs with exact PIE matches for En-Nl, and collect translations using the En-Nl model. We label both the source-target pairs and model translations heuristically. Table 6.3 illustrates how the predicted translations' labels relate to the labels of target translations and provides BLEU and COMET-22 (Rei et al., 2022) scores per subset. 54% of the target translations are labelled as paraphrased instances, which is substantially higher than the percentage of paraphrased instances in the model's translations. Of the target translations labelled as a paraphrase, only half of the model's translations are also labelled as a paraphrase, signalling a lack of memorisation in the model. In terms of BLEU and COMET scores, model translations for examples with paraphrased target translations score substantially lower compared to those with word-for-word target translations, emphasising the negative impact of idioms on translation quality. When interpreting these results, we should, however, keep in mind that translation is a many-to-many mapping, and that an individual idiom can have multiple correct paraphrases, as discussed in §2.3.3. n-gram overlap could thus underestimate translation quality when idioms are involved. COMET relies on neural models for quality estimation and is thus more semantics-aware. Nonetheless, COMET could still underestimate the translation quality, since the neural models employed by COMET (among which cross-lingual `RoBERTa`, Conneau et al., 2020) likely *also* suffer from the tendency to interpret non-compositional phrases overly compositional, particularly for phrases that are very infrequent.

6.3 Attention analyses

We now turn to comparing how literal and figurative PIEs are processed by transformer. Whether a PIE is figurative depends on the context – e.g. compare "in culinary school, I felt *at sea*" to "the sailors were *at sea*". Within transformer, contextualisation of input tokens is achieved through the attention mechanisms, which is why they are expected to combine the representations of the idioms' tokens and embed the idiom in its context. This section discusses the impact of PIEs on the encoder's self-attention and the encoder-decoder cross-attention. Consider §2.1 for a review of how attention is defined, and the role attention plays within transformer.

Attention within the PIE For the En-Nl transformer, Figure 6.4a visualises the distribution of attention weights in the encoder's self-attention mechanism for incoming weights to one noun contained in the PIE from the remaining PIE tokens. In the figures, we refer to the subset of sentences that have a figurative PIE and a paraphrased translation as '*fig-par*'. The subset of sentences with a literal PIE and a word-for-word translation is indicated by '*lit-wfw*'. We compare those two subsets, as well as all instances of figurative PIEs ('*fig*') to all instances of literal PIEs ('*lit*'). Overall, there is increased attention for figurative occurrences of PIEs compared to literal instances. This difference is amplified for the subset of figurative PIEs yielding paraphrased translations. This pattern is consistent for all language pairs, as is displayed in Figure 6.4d that presents the difference between the mean attention weights of the figurative, paraphrased instances, and the mean weights of the literal instances translated word for word. In other words, figurative (paraphrased) PIEs are grouped more strongly than their literal counterparts. This is in line with previous work that identified explicitly representing idioms as one word can improve idiom translations (Zaninello and Birch, 2020) and idiomaticity detection in contextual embeddings (Hashempour and Villavicencio, 2020).

Attention between PIEs and context To examine the interaction between a PIE and its context, we obtain the attention weights from tokens within the PIE to nouns in the surrounding context (Figure 6.4b).[6] Similarly, the attention from the surrounding context to PIE nouns is measured (Figure 6.4c). There is reduced attention from PIEs to context for figurative instances, which mirrors the effect observed in Figure 6.4a: increased attention within the PIE is accompanied by reduced attention to the context. This pattern is consistent across languages (Figure 6.4d). Reduced attention to the context seems somewhat counter-intuitive, given that correctly interpreting and translating PIEs requires disambiguation through the context. This could potentially signal that the model does *not* adequately disambiguate PIEs and has mostly memorised how to translate PIEs independent of context. The fact that context may be inadequately leveraged by neural models during idiom detection was previously pointed out by related work (§2.3.3), and we previously observed some empirical evidence for that in §5.3.3.

From the context to the PIE, the average weight is slightly higher for literal PIEs, but the effect size is small, indicating only a minor impact of figurativeness on the context's attention weights. This will be further investigated in §6.4.

Cross-attention To analyse the encoder-decoder interaction, we decode translations with a beam size of five and extract the cross-attention weights for those translations. Afterwards, alignments are computed for the models' predictions by, together with 1M sentences from the OPUS corpus per target language, aligning them using the `eflomal` toolkit (Östling and Tiedemann, 2016). The alignment is used to measure attention from a token aligned to a PIE's noun to that noun on the source side.

How does the automated aligner handle paraphrases when automatically aligning sentences with PIEs to translations labelled as a paraphrase? For many PIEs ($\leq$ 34% of the *fig-par* sentences for all language pairs), the paraphrases do not have a word in the translation aligned to the PIE keyword on the source side. These examples are excluded.

[6]We consider a context of 10 tokens to left, and 10 tokens to the right, or smaller, as sentence length permits. The mean total context size is 15 for both figurative and literal examples.

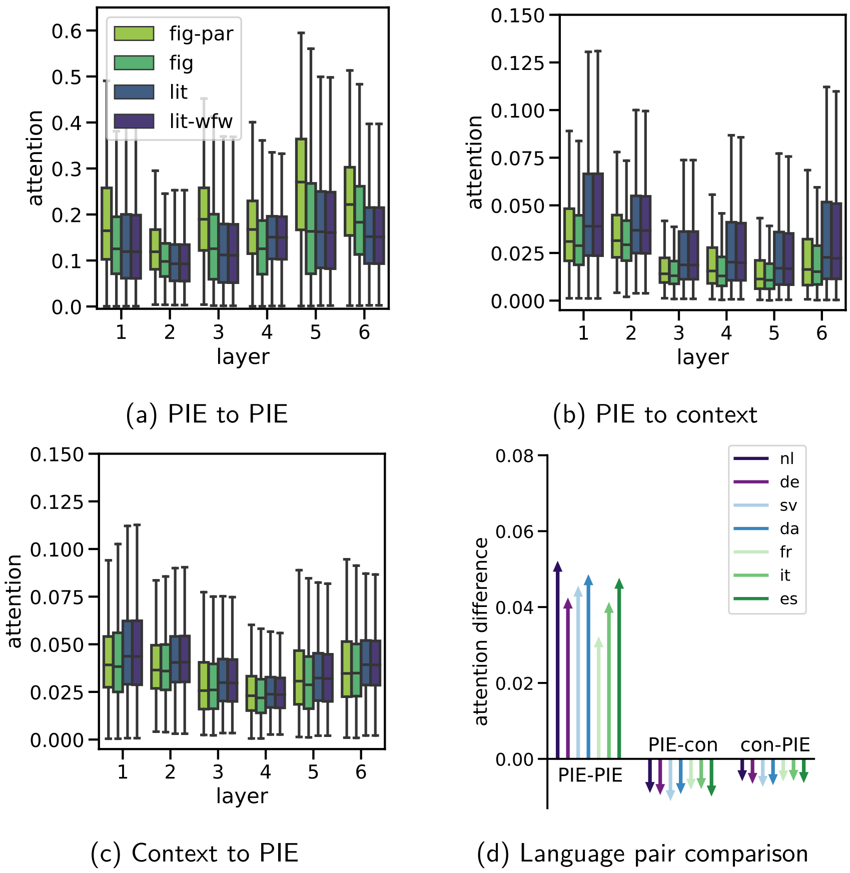
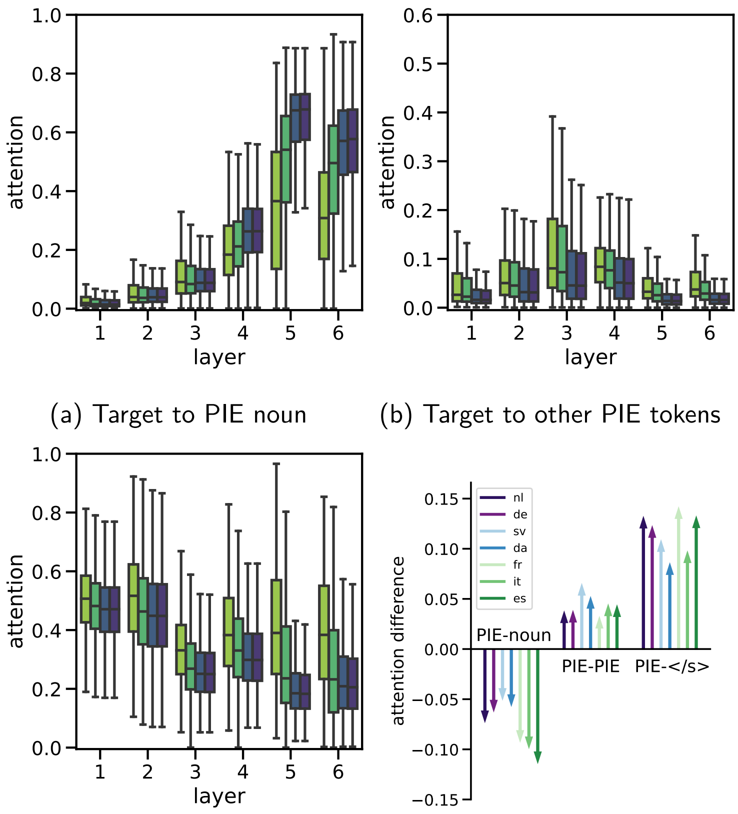


Figure 6.4: Weight distributions of the encoder's self-attention for En-Nl (a-c), and the mean difference of *fig-par* and *lit-wfw* for all language pairs (d). Boxes represent quartiles; whiskers show the distribution, excluding outliers.

However, for a subset that appears more well-known, there are common paraphrases that the PIE keyword aligns with. We provide examples for Dutch in Table 6.4. The examples provided in the table together cover 48% of all aligned sentences used in the cross-attention analysis for the *fig-par* category, and all are reasonable alignments, strengthening the reliability of the cross-attention results.

We now turn to the cross-attention patterns observed: Figure 6.5a presents the attention distribution for the weights that go from the noun's translation to that PIE noun on the source side, for the En-Nl model. There is a stark difference between figurative and literal PIEs, through reduced attention on the source-side noun for figurative PIEs. This difference is particularly strong for the figurative sentences that are paraphrased during the translation: when paraphrasing, the model appears to rely less on the source-side noun than when translating word for word. Where does the attention flow, instead? To some extent, to the remaining PIE tokens (Figure 6.5b). A more pronounced pattern of increased attention on the EOS token is shown in Figure 6.5c. Figure 6.5d compares the mean attention weights of the seven language pairs for the figurative inputs that are paraphrased to the literal samples that are translated word



(a) Target to PIE noun (b) Target to other PIE tokens

(c) Target to </s> (d) Language pair comparison

Figure 6.5: The cross-attention for target-side tokens aligned to PIE nouns for En-Nl (a-c), and the mean difference between *fig-par* and *lit-wfw* for all language pairs (d).

for word, confirming that these patterns are not specific to En-Nl.

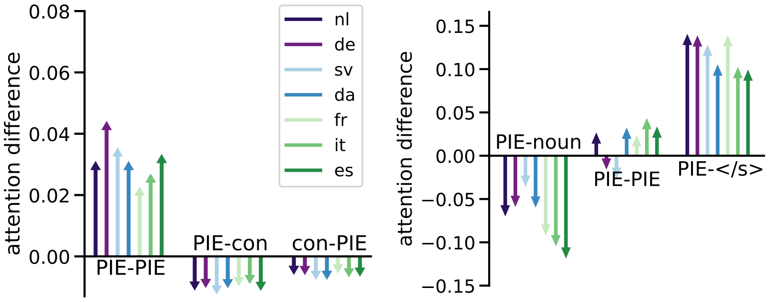
The large attention weights for the EOS token resemble attention patterns that have been observed by Clark et al. (2019b) for BERT's [SEP] token. Since the [SEP] token is in itself a contextualised token, Clark et al. entertained the hypothesis that the token might aggregate information from an entire segment, which could be used by attention heads by attending to [SEP]. However, gradient-based feature importance analyses, qualitative analyses of attention heads, and the fact that [SEP] was found to mostly attend to itself, instead suggested that this behaviour indicates a *no-operation*.[7] Ferrando and Costa-jussà (2021) later analysed the same phenomenon for NMT transformers' cross-attention mechanisms, concluding – by analysing the norms of EOS tokens' value vectors – that large cross-attention weights for the source-side EOS token have a similar no-operation function, regulating the amount of information the decoder collects from the input sequence. In our case, the increased attention to the EOS token for figurative-paraphrased PIEs could thus suggest that the decoder collects

[7]Following the publication of this chapter, a related phenomenon was observed in autoregressive LLMs, in which a lot of attention flowed to the BOS token. This became known as the 'attention sink' phenomenon (Xiao et al., 2024). See Ferrando et al. (2024) for a discussion on parallels between the two phenomena.

PIE	**Dutch paraphrase (literal backtranslation)**	**Aligned tokens**
across the board	over hele linie (*over the whole line*)	board → linie
behind the scenes	achter de schermen (*behind the screens*)	scenes → schermen
break new ground	nieuwe weg inslaan (*take a new road*)	ground → weg
by heart	uit het hoofd (*from the head*)	heart → hoofd
by the same token	op dezelfde manier (*in the same way*)	token → manier
come to mind	in me opkomen (*come up in me*)	mind → me
come of age	volwassen worden (*become an adult*)	age → volwassen
face to face	oog in oog (*eye in eye*)	face → oog
follow suit	het voorbeeld volgen van (*follow the example of*)	suit → voorbeeld
for good measure	in goede mate* (*in good measure*)	measure → mate
from scratch	vanaf nul (*from zero*)	scratch → nul
from the word go	vanaf het begin (*from the start*)	word → begin
get a move on	schiet op (*hurry*)	move → schiet
get the picture	een completer beeld krijgen (*get a more complete vision*)	picture → beeld
get to grips with	(aan)pakken (*take on*)	grips → pakken
give someone the creeps	kriebels krijgen (*getting tickles*)	creep → (krie)bel
in broad daylight	op klaarlichte dag (*on a luminous day*)	day(light) → dag
in full swing	in volle gang (*in full progress*)	sw(ing) → gang
in the flesh	in levende lijve (*in the living body*)	flesh → lij(ve)
in the long run	op de lange termijn (*on the long term*)	run → termijn
in the short run	op de korte termijn (*on the short term*)	run → termijn
keep a low profile	zich gedeisd houden (*to lay low*)	profile → (gede)is(d)
off the record	onofficieel (*unofficial*)	record → (onoffici)eel
on someone's mind	iets aan je hoofd hebben (*have something on your head*)	mind → hoofd
once in a while	af en toe (*on and off*)	while → toe
out of the blue	uit het niets (*out of nothing*)	blue → niets
out of the question	uit de boze (*from the bad*)	question → boze
set eyes on	zien / zag (*see / saw*)	eyes → zag
small print	in de kleine lettertjes (*in the little letters*)	print → (letter)tjes
take a back seat	op de achterbank* (*on the back bench*)	seat → bank
take stock	de balans opmaken (*make up the balance*)	stock → balans
to all intents and purposes	in alle opzichten (*in all aspects*)	intent → opzichten
to boot	opstarten* (*to start*)	boot → (op)starten
to the tune of	voor het bedrag van (*for the amount of*)	tune → bedrag
with a view to	met het oog op (*with the eye on*)	view → oog

Table 6.4: PIEs for which the word most commonly aligned to the keyword occurs > 20 times. Together, these keywords determine 48% of all the alignments used to perform the cross-attention analysis for *fig-par* in the English-Dutch model. Subwords shown in brackets are due to the tokens used in Marian-MT: `eflomal` aligns the parts outside of the brackets to one another. *Example of a PIE for which the heuristic annotation missed out on a potential literal translation.

less information from the source side than it does for expressions translated word for word.



(a) Encoder self-attention (b) Cross-attention

Figure 6.6: The self-attention and cross-attention differences between the figurative-paraphrase and literal-word-for-word subsets, for examples with identical PIE matches.

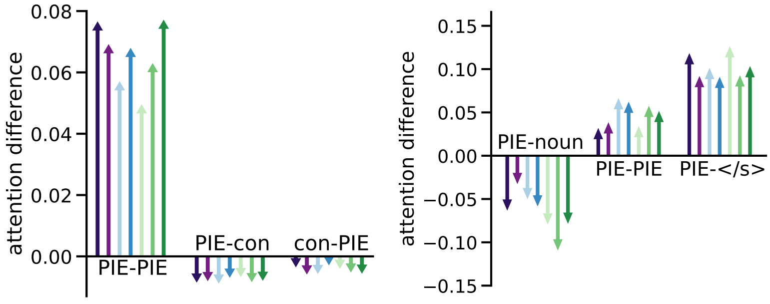
Revisiting attention for data subsets The aforementioned results include all MAGPIE examples previously mentioned in §6.2. To further investigate whether the differences in attention patterns observed are due to factors other than figurativeness and paraphrasing, we recompute the attention patterns for three additional data subsets.

Firstly, we consider **PIE identical matches**: this subset only includes samples for which MAGPIE reports an identical match between the PIE and the English sentence, which applies to 17k samples. This subset excludes sentences with modifications to the typical surface form of a PIE, such as inflected variants, variants with words inserted, or passivised variants (e.g. "You get the loot when the beans are spilled in (...)"). Figure 6.6 shows the attention patterns previously discussed for the encoder's self-attention and cross-attention, providing the same qualitative findings as before.

The second subset considered only contains idioms that are among **all of the subsets** of figurative, literal, paraphrased and word-for-word instances, covering 11k examples from the dataset. The results for the self-attention and cross-attention patterns are shown in Figure 6.7. These results lead to the same qualitative findings as previously mentioned, and, in the encoder, the PIE to PIE attention patterns for figurative and literal PIEs are even more different than before.

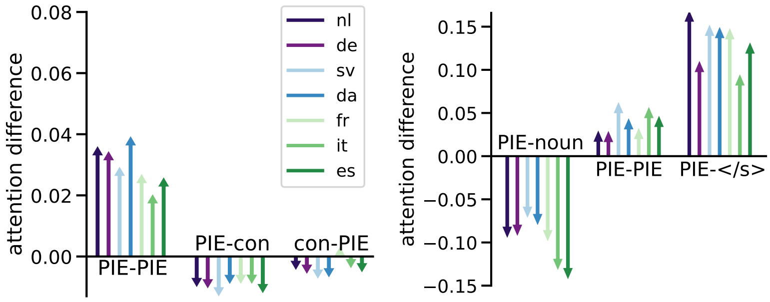
Lastly, we restrict **the length of a PIE** by selecting a subset of examples that contains three tokens annotated as belonging to the PIE, with one non-PIE token in between. This covers a subset of approximately 7k samples, with small variations between languages due to slightly different tokenisation of the English words. Figure 6.8 presents the results for the self-attention and cross-attention analyses, respectively. Qualitatively, our findings for this subset do not differ from the previous findings, although in absolute terms, the differences in the encoder attention within the PIE are of a smaller magnitude than before.

Collectively, the results provide the observations depicted in Figure 6.1. When paraphrasing a figurative PIE, the model groups idioms' parts more strongly than it would



(a) Encoder self-attention (b) Cross-attention

Figure 6.7: The self-attention and cross-attention differences between the figurative-paraphrase and literal-word-for-word subsets, for PIEs that are in the intersection of all four labels (figurative, literal, paraphrase, word-for-word).



(a) Encoder self-attention (b) Cross-attention

Figure 6.8: The self-attention and cross-attention attention differences between the figurative-paraphrase and literal-word-for-word subsets, for stimuli that were length-controlled.

otherwise – i.e. it captures the PIE more as one unit. Increased attention within the PIE is accompanied by reduced interaction with context, indicating that when PIEs are paraphrased, they are translated more in a stand-alone manner compared to PIEs translated word for word. We also observed reduced cross-attention to the source-side PIE and increased attention to the </s> token when the model emits the translation of figurative (paraphrased) PIEs. This suggests that when paraphrasing, the decoder acts more detached from the encoder compared to when translating word for word.

6.4 Hidden representations analyses

Within transformer, the encoder's upper layers have previously been found to encode semantic information (e.g. Raganato and Tiedemann, 2018). PIEs' hidden states are expected to transform over layers due to contextualisation and become increasingly more indicative of figurativeness. This section focuses on the impact of PIEs on the hidden states of transformer's encoder. We first discuss how much these hidden states change between layers. Secondly, we measure the influence of a token by masking it out in the attention and analysing the degree of change in the hidden representations of its neighbouring tokens. This analysis is performed to consolidate findings from §6.3, since the extent to which attention can explain model behaviour is a topic of debate (as discussed in §2.1.4).

6.4.1 PIE changes over layers

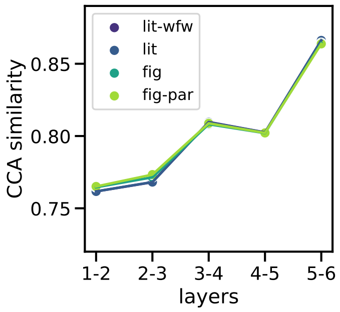
To compare representations from different layers, we apply CCA (Hotelling, 1936) (previously detailed in §2.1.4), using an implementation from Raghu et al. (2017). CCA linearly transforms two sets of representations of the same datapoints, such as to maximise the correlations between the transformed representations. We perform CCA using >60k random token vectors for a previously unused subset of the MAGPIE corpus – the subset of sentences that did not contain nouns in the PIEs – to compute the CCA projection matrices that are then used to project new datapoints before measuring the datapoints' correlation. The CCA similarity reported in the graphs is the average correlation of projected datapoints. We do not perform CCA separately per data subset due to the small subset sizes and the impact of vocabulary sizes on CCA correlations for small datasets (in Appendix D.3 we demonstrate how CCA is sensitive to these factors).[8]

We compute the CCA similarity for hidden states from consecutive layers for PIE and non-PIE nouns. Figurative PIEs in layer l are typically less similar to their representation in layer $l-1$ compared to literal instances (shown in Figures 6.9b for the En-Nl transformer), and that difference is larger for figurative PIEs that are paraphrased. Figure 6.9c summarises the difference between literal-word-for-word and figurative-paraphrased cases for the seven different language pairs. The results for non-PIE nouns (see Figure 6.9a for the En-Nl results) do not differ across data subsets, suggesting that changes observed for figurative PIEs are indeed due to figurativeness, and not due to other differences between sentences from the different data subsets. The differences in similarity are the largest for layers three to five.

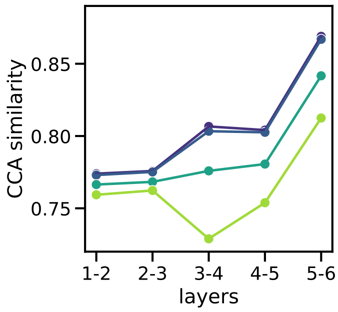
6.4.2 Intercepting in attention

We now compute similarities of representations for our models in two setups: with and without one token masked in the attention mechanism, similar to Voita et al. (2019a). Masking a token means that other tokens are forbidden to attend to the chosen one; this is implemented via the mask from Equation (2.2) as discussed in §2.1. This can reveal whether the attention patterns discussed in §6.3 are indicative of the influence

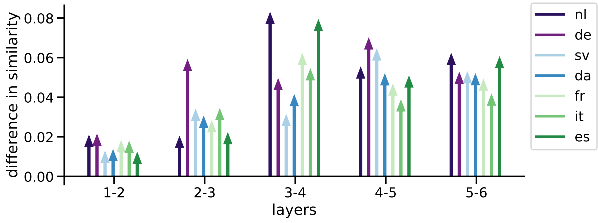
[8] Extensions of CCA have been proposed that limit the number of CCA directions over which the correlation is computed, to only include directions that explain a large portion of the variance (Raghu et al., 2017; Morcos et al., 2018). We do not remove directions, such as to avoid removing smaller variance components that could still cue figurativeness (the focus of our work).



(a) Non-PIE nouns



(b) PIE nouns



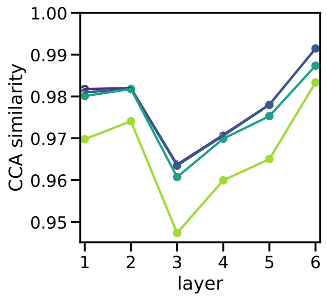
(c) Language pair comparison for PIE nouns

Figure 6.9: CCA similarity for layers l and $l+1$, for PIE and non-PIE nouns. The language comparison displays the difference in similarity between *lit-wfw* and *fig-par*.

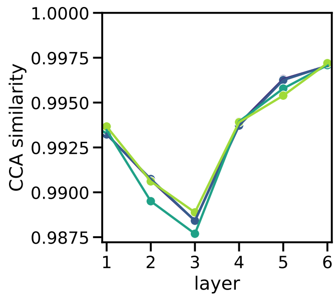
tokens have on each other's hidden representations.[9] The first representation is the hidden representation from layer l for a token encoded as usual. The second one is the hidden representation of layer l when applying the first $l-1$ layers as usual and masking one token in the lth layer. CCA is again performed on separate data to provide the projection matrices applied before computing similarities in the remainder of this subsection.

Masking a PIE token To estimate the influence of PIE nouns, we first compute the CCA similarity between two representations of tokens from the PIE's context while masking one PIE noun in the attention for one of those representations. Similarly, we measure the influence on other tokens within the PIE when masking one PIE noun. Within the PIE, the influence is the largest for figurative-paraphrased instances (see Figure 6.10a for En-Nl and Figure 6.10e, 'PIE-PIE' for averages over layers for all language pairs). This is in line with the attention pattern observed. However, when inspecting the influence of masking a PIE noun on context tokens, there are barely any differences between figurative and literal PIEs (see Figure 6.10b, and 'PIE-con' in Figure 6.10e). This indicates that the slight difference in attention from the context to the PIE for figurative and literal PIEs observed in §6.3 does not necessarily affect the hidden representations.

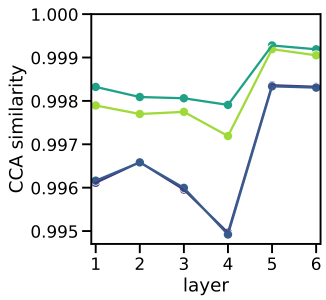
[9]Note that it does not necessarily mean that the representations do not encode any information about the masked token, since due to contextualisation prior to layer l the remaining tokens may also partially encode the masked token's identity.



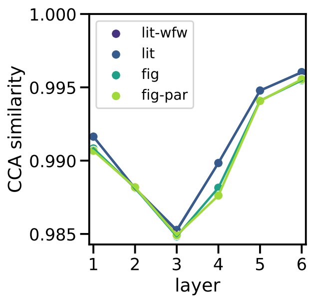
(a) Influence of masking a PIE noun on the PIE



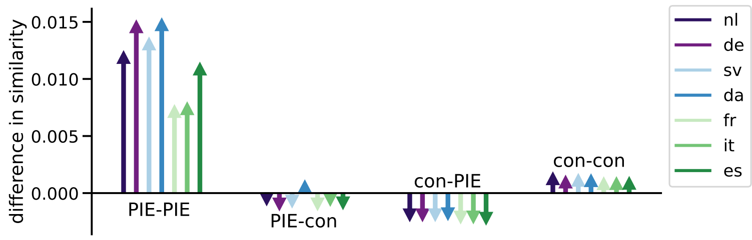
(b) Influence of masking a PIE noun on the context



(c) Influence of masking a context noun on the PIE



(d) Influence of masking a context noun on the context



(e) Language pair comparison

Figure 6.10: Influence of masking a PIE noun in the attention on (a) other PIE tokens, (b) other context tokens. Influence of masking a non-PIE noun on (c) PIE tokens and (d) other non-PIE tokens. (e) shows the difference in similarity between *lit-wfw* and *fig-par*.

Masking a context token Lastly, we measure the influence of masking a noun in the context of the PIE on PIE tokens and non-PIE tokens. Within the PIE, as shown in Figure 6.10c for En-Nl and Figure 6.10e ('con-PIE') for all language pairs, figurative PIE occurrences are less affected by the masked context noun compared to literal PIE occurrences. Again, this mirrors the patterns observed for attention, where less attention was paid to the context for figurative PIEs. When masking a non-PIE noun and measuring the influence on non-PIE tokens, one would hardly expect any differences between data subsets, as is confirmed in Figures 6.10d and 6.10e ('con-con').

In summary, these analyses confirm most of the trends noted for attention patterns. Intercepting in the attention through masking indicated that for PIE tokens, there is less interaction with the context. However, this does not necessarily mean that the

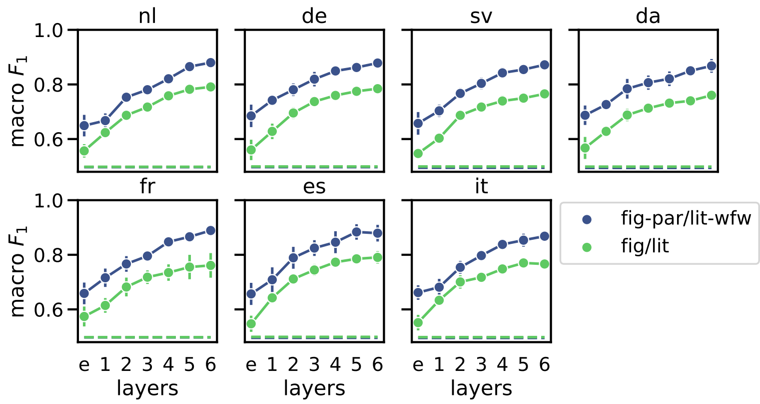


Figure 6.11: Macro F_1-score for probes predicting PIEs' labels. Error bars show standard deviations over folds; the dashed lines represent a random baseline.

context interacts less with figurative PIEs compared to literal PIEs, even if there was a slight difference in attention in §6.3. The CCA analyses furthermore showed that nouns from figurative PIEs are distinct in how they change over layers, compared to non-PIE nouns.

6.5 (Amnesic) probing for figurativeness

The previous analyses compared the hidden states for figurative and literal PIEs, but do not use these labels otherwise. We now train logistic regression *behavioural probing classifiers* (Conneau et al., 2018; Hupkes et al., 2018, i.a.) – previously introduced in §2.1.4 – to predict the label from hidden representations. Afterwards, we use amnesic probes to alter the models' translations.

Behavioural probes The probes' inputs are the hidden states of PIE tokens, and the F_1-scores are averaged over five folds. All samples from one PIE are in the same fold, such that the classifier is evaluated on PIEs that were absent from its training data. The results (Figure 6.11) indicate that figurativeness can be predicted from these encodings, with performance increasing until the top layer for all language pairs. Although there is not one individual layer that stands out in terms of a particularly large increase in F_1, the highest layers show reduced increases compared to the lower layers, suggesting that, in line with our previous results, figurativeness affects much more than just the highest layers, and figurative PIEs become gradually more distinct. F_1-scores for the embeddings already exceed a random baseline, indicating some idioms are recognisable independent of context.

Amnesic probes Finally, we use probing classifiers to change models' PIE translations through *amnesic probing* (Elazar et al., 2021) – previously introduced in §2.1.4 –

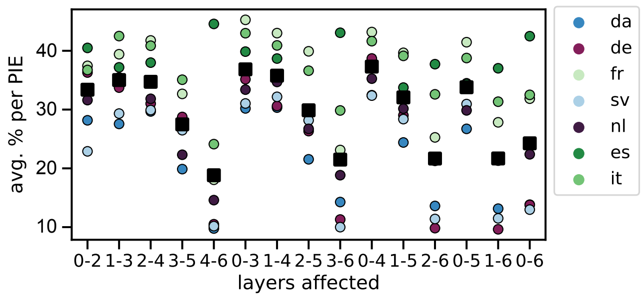


Figure 6.12: Impact of the selection of layers affected by INLP. Squares indicate the mean % over languages.

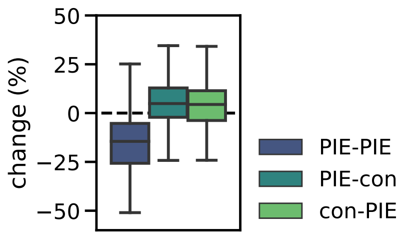


Figure 6.13: Changes in attention patterns after INLP for En-Nl.

	Figurative probe		Frequency probe	
	%	BLEU	%	BLEU
nl	36	75	34	75
de	33	68	33	69
sv	27	77	27	77
da	32	77	27	78
fr	37	77	30	76
it	39	76	34	77
es	40	78	30	78

Table 6.5: The effect of amnesic probing, measured using the mean success rate per PIE (%), and the BLEU score of paraphrased translations that changed into a word-for-word translation, comparing the translation before and after INLP.

which removes features from hidden states by projecting them onto the null-space of trained linear probes using INLP (Ravfogel et al., 2020) and measures the effect of these interventions. We train 50 classifiers to distinguish figurative and paraphrased PIEs from literal PIEs translated word for word, and use them to calculate the INLP projection matrix. Afterwards, we run the sentences with previously paraphrased PIEs through the model while removing information from the PIEs' hidden states using the projection matrix. Per PIE, we record the percentage of translations that are no longer paraphrased.

We separate the data into five folds, of which we use one to determine where in the model to intervene, based on the average success rate per PIE (where success means achieving a word-for-word translation). As shown in Figure 6.12, there is quite some variation among languages, but generally, intervening in the lower layers of transformer is the most successful. We now continue computing results on the remaining four folds by intervening in $l \in \{0,1,2,3,4\}$. We report the percentages along with BLEU scores comparing translations that changed label before and after INLP.

Table 6.5 presents the results. When intervening in the hidden states for all layers

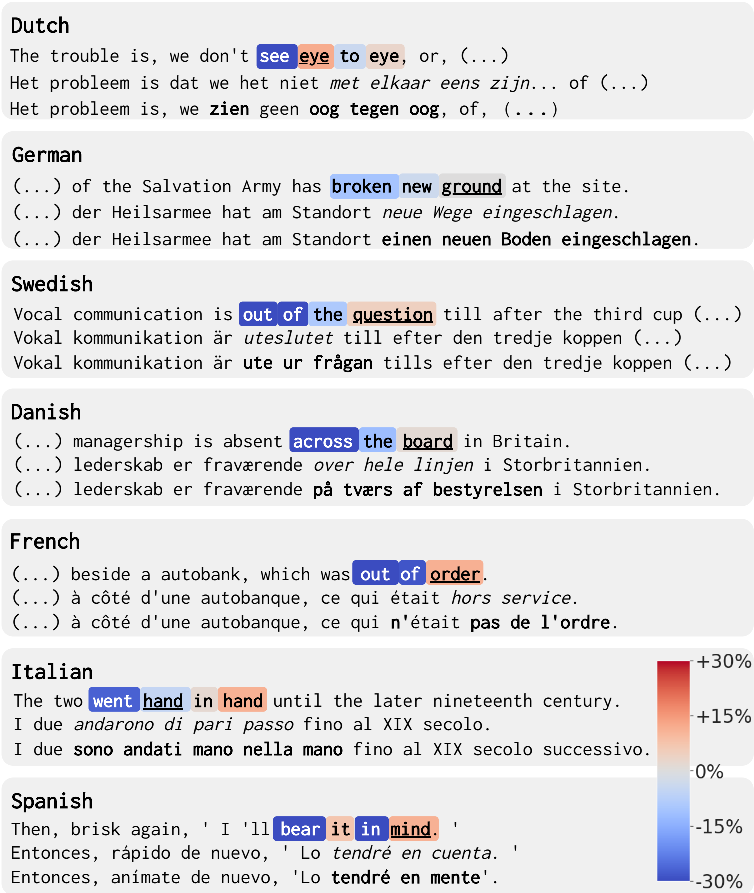


Figure 6.14: Source sentences and translations before and after INLP. PIEs and word-for-word translations are in bold font; paraphrases are in italics. Colours indicate attention changes with respect to the underlined nouns.

$l \in \{0, 1, 2, 3, 4\}$, the average success rate per PIE ranges from 27% (for Swedish) to 40% (for Spanish). At the same time, the high BLEU scores for examples that changed in terms of their translation's label demonstrate that the translations are still very similar; we did not impair the model's ability to translate, and the interventions successfully targeted the PIE only. If we now extract the attention patterns for the translations that changed, we note that the interventions reduced attention within the PIE and increased interaction with the context (see Figure 6.13 for En-Nl). Table 6.5 also provides results for a baseline probe predicting whether the half-harmonic mean of the Zipf-frequency of PIE tokens is below or above average. This probe is successful too, emphasising how brittle idiomatic translations are: when removing information from the hidden states, the model reverts to compositional translations.

Figure 6.14 provides example translations before and after the application of INLP, while indicating how the attention on the underlined noun changes. Generally, the attention on that noun reduces for PIE tokens other than itself.

In summary, the behavioural probing accuracies differed across layers and suggested figurative PIEs become more distinct across the different layers. When we applied amnesic probing to the encoder's hidden states, we identified that we can alter models' translations through this, and that the attention patterns resemble patterns for literal PIEs more when doing so. This further consolidates a causal connection between the model's paraphrasing of figurative PIEs and the attention. However, amnesic probing did not change the paraphrases for all idioms. This could either be because figurativeness is not merely linearly encoded in the hidden states, or because the decoder might be able to recover information about the figurativeness of PIEs when we intervene in the encoder only.

6.6 Conclusion and discussion

PIEs pose challenges for MT because adequately translating them requires disambiguating whether the PIE is used figuratively, and paraphrasing the memorised meaning of figurative PIEs in the target language. Behavioural evaluation has shown that this is challenging for transformer NMT systems (e.g. Fadaee et al., 2018; Zaninello and Birch, 2020), and in §3.4 we, ourselves, already noticed that sentences containing idioms score lower in terms of both memorisation during training and evaluation during testing compared to control stimuli. Yet, it is largely unknown how this process affects transformer internally, and which mechanisms positively contribute to cases where the model *does* paraphrase the idiom.

In this chapter, we focused on the internal mechanisms of NMT systems that enable idiomatic translations, a specific form of memory recall since paraphrased translations of idioms do not straightforwardly follow from the translations of the individual words contained in the idiom, and have to be memorised. To examine this at scale, we used an English idiom corpus and heuristically labelled translations for seven target languages. We compared hidden states and attention patterns for figurative and literal PIEs. We identified that, in the encoder, figurative PIEs are grouped more strongly in the attention as one lexical unit than literal instances and interact less with their context. The effect is stronger for paraphrased translations, suggesting that capturing idioms as single units and translating them in a stand-alone manner aids idiom processing. By analysing the hidden states, we confirmed that the changed attention patterns do, in fact, lead to changes in the residual stream. The finding regarding the grouping of tokens agrees with results from Zaninello and Birch (2020), who ascertain that encoding an idiom as one word improves translations. It also agrees with the amnesic probing, which yielded more compositional translations whilst changing the attention. By relying less on the encoder's output, the decoder determines the meaning of figurative PIEs more independently than for literal ones.

The various experiments performed underscored that encoding figurativeness of PIEs is not something specific to individual layers, or to the highest layers of transformer, which have been suggested to capture high-level semantic features (e.g. Raganato and Tiedemann, 2018). PIEs become gradually more distinct over layers, and attention patterns and hidden representations of figurative PIEs stand out from layer one onwards.

Although we learnt about the mechanics involved in idiomatic translations, the vast majority of translations were still word-for-word, indicating that non-compositional processing does not emerge well (enough) in transformer. Paradoxically, for many years, it was a trend in NLP to encourage more compositional processing in NMT, to increase systems' robustness and reduce their volatility as discussed in chapter 5 (Raunak et al., 2019; Chaabouni et al., 2021; Li et al., 2021a, i.a.). We recommend caution with this: while favouring compositional translations might improve robustness, it is likely to harm translations of formulaic or figurative language. It may be beneficial to evaluate the effect of compositionality-favouring techniques on non-compositional phenomena like idioms to ensure their effect is not detrimental to these phenomena.

6.6.1 Limitations

We identify four main limitations of our work. Firstly, our **heuristic labelling is inherently limited**. As discussed in §6.2, the annotation studies conducted to assess our heuristic labelling method showed that it does not achieve perfect accuracy. The method performed best for word-for-word translations of literal PIEs and worst for paraphrased translations of literal PIEs. Consequently, our analysis has primarily focused on the differences between the figurative-paraphrased and literal-word-for-word subsets. Nonetheless, even the figurative-paraphrased subset contains some erroneous annotations, as illustrated in Table 6.4. We consider the method sufficiently accurate to reveal general trends in differences between subsets, which has been the focus of our analysis. However, the labelling does not guarantee that what is labelled as a 'paraphrase' is in fact a faithful paraphrase. It also only labels translations based on the absence or presence of translations of (mostly) the idiom's nouns, ignoring how verbs are translated. We opted for this to make the labelling more accurate, since nouns have fewer surface forms than verbs do.

The second limitation is that we examined only **seven language pairs** and a single **architecture** (`transformer-base`, Vaswani et al., 2017). All language pairs belong to the Germanic and Romance branches of the Indo-European language family, which restricts the generalisability of our conclusions. We limited ourselves to these languages because they are generally high-resource and are likely to exhibit reasonable coverage of the idioms used in training data. Even within this limited scope, only around 20% of figurative PIEs were paraphrased by the models; for lower-resource languages, this figure would likely be even lower. We chose `transformer-base` as it has been the predominant architecture used in open-source pretrained translation systems since transformer's introduction in 2017.

Thirdly, we emphasise that our focus was primarily on **the role of the encoder** in paraphrasing idiomatic translations. Our only decoder-focused experiment was the cross-attention analysis. Decoder-based analyses are more difficult to scale, owing to the many surface forms that idiomatic paraphrases may take and the potential inaccuracies of alignment tools used in the process. We manually evaluated a subset of our `eflomal` alignments to ensure the reliability of our results. We encourage future research to investigate the decoder's role more thoroughly – particularly the causal influence of the attention patterns we identified for figurative-paraphrased PIEs.

Lastly, although we analysed the mechanisms behind idiom processing, we did not explore **methods for improving idiomatic translations**. We encourage future work to leverage our insights. In §6.2, we noticed the relevance of the frequency of PIEs in the training material. Dedicated PIE corpora are scarce, but even without those, one can filter standard NMT corpora based on frequently-occurring surface forms of PIEs, and perform some continual learning on a training data subset. One could also explicitly incorporate the non-compositional nature of idioms in the training objective, e.g. by penalising high probabilities for literal translations of source-side idioms, or learning specialised attention heads to group multi-word expressions.

6.6.2 Retrospective and outlook

In the years that followed the publication of this chapter, great progress has been made in the translation of idioms and related model analyses. Three articles, in particular, present findings highly related to our work:

- **On heuristic annotation**: Baziotis et al. (2023) proposed a heuristic labelling method similar to ours to identify literal translation errors. Their `LitTER` metric similarly uses bilingual dictionaries to collect potential literal translations of the source idiom in a 'blocklist'. Different from our method, they update their blocklist using target translations by removing words occurring in the reference from the blocklist. `LitTER` was used to demonstrate that for EuroParl data and English to French/Spanish translation, literal translation errors are highly frequent, and that pretraining a model on monolingual data boosts performance.

- **On the attention patterns**: Directly inspired by our findings, Lim et al. (2024) examine how well various features predict the translation difficulty of segments, including attention within the segment, attention to the context and attention to the EOS-token. They quantify difficulty based on human reading times and identify that although attention is not directly predictive of reading time, source segments that are harder to translate direct less attention to the context. They also establish that target segments that are harder to translate show increased cross-attention to the EOS token. Idiomatic paraphrases, which we explored, are one example of segments that are harder to translate, and the findings from Lim et al. thus suggest that the qualitative patterns we identified might not be unique to idiomatic translations.

- **On idiom analyses over layers**: Haviv et al. (2023) studied how idiomatic predictions are formed over the many layers of `BERT` and `GPT-2`. They present idioms without context, requiring their models to predict the final token. They identify a two-stage process, where the earlier layers quickly increase the rank of the predicted token, and the later layers boost the probability of that token. For idioms that the model memorised correctly, the predicted token starts out at a lower rank compared to non-memorised examples, but ends with a much higher probability. By intervening in the feedforward modules of the earliest (but not later) layers, the idiomatic memories were suppressed. In spite of the difference in models analysed and the task considered, our results do also suggest that encoding idiomaticity already starts in the lowest layers: both attention patterns (§6.3) and the masked hidden state analyses (§6.4) show the 'figurative-paraphrased' subset diverging from layer one, and INLP-based amnesic probing was also more effective in early layers (§6.5).

The lack of parallel idiom corpora has been a major limitation when it comes to analysing or improving idioms' translations. While this remains a core issue, some initiatives have released new data, such as a Korean-English challenge set with multiple types of figurative language (Lee et al., 2025), IdiomsInCtx-MT by Stap et al. (2024) including test sets for idioms in context for three language pairs, and ACES by Amrhein et al. (2022), which includes overly-literal translations in German-English as one of 68 core challenging phenomena used to evaluate common MT quality metrics.

Solutions for improving idiomatic translations have primarily targeted fine-tuning models using parallel data (Santing et al., 2022) and incorporating the idiom's meaning in the translation. The meaning can be incorporated by encouraging the model to paraphrase the idiom before translating it (Santing et al., 2022), including information about the idiom and alternative idioms in the target language from an external database (Li et al., 2024b; Donthi et al., 2025), or including k-NN based retrieval while translating such as to retrieve training examples including the same idiom (Liu et al., 2023a).

Even though these solutions specific to the nature of idioms have shown moderate improvements, the largest benefits still appear to have come from simply scaling the training data and model sizes, which is demonstrated both by targeted experiments showing improvements from monolingual pretraining (Baziotis et al., 2023; Stap et al., 2024), and the general superiority of larger LLMs that were released post 2022. Raunak et al. (2023), for instance, demonstrate that `GPT-3.5` predecessors produced more

Figure 6.15: Interaction with ChatGPT, `GPT-4o mini`, retrieved on April 14, 2025.

non-literal translations of idioms compared to both academic and other commercial translation models. And yet, for the most powerful commercial and non-commercial translation systems and LLMs, researchers continue to echo our findings of overly compositional translations for a range of languages, such as German, Spanish and Japanese (Ferrando et al., 2023), Arabic (Obeidat et al., 2024), Urdu (Basit et al., 2024) and Indonesian (Dewayanti and Margana, 2024). Figure 6.15 illustrates how `GPT-4o mini`, for instance, makes such a mistake of translating an idiom word for word when translating from English to Dutch ("chickens came home to roost" is translated as "kippen kwamen thuis om te rusten"). Interestingly, upon querying the model, it is able to point out its own mistake, demonstrating it has memorised the correct meaning, yet does not produce it in the translation. This underscores how non-compositionality poses challenges, and how these are amplified in the multi-step process of translating idioms, involving disambiguation, memory recall *and* adequate paraphrasing.

Chapter 7

Conclusion

In this thesis, I discussed the topic of memorisation and the extent to which it stands in contrast with generalisation. I considered memorisation as a quantifiable phenomenon that is the result of a model and its training procedure, and memorisation for the case study of non-compositional idiomatic expressions. I also discussed the topic of compositionality more broadly, studying to what extent transformer behaves compositionally, while acquiring non-compositional idiom processing. The conclusions from the four content chapters provide us with the following seven lessons concerning the general research questions laid out in chapter 1.

What characterises memorised examples?

With respect to RQ1, we have learnt that **(1) memorisation is not a mysterious phenomenon, but a process that is predictable based on datapoints' features**. In chapter 3, we created a resource of memorisation metrics for NMT, describing datapoints through their training memorisation (TM), counterfactual memorisation (CM) and generalisation scores, emphasising that memorisation is not necessarily binary, but exists along a continuum. A large part of the variance in these scores can be explained based on surface-level features, such as the overlap between the source and target sequences, and these features affected systems from five Indo-European language pairs in the exact same way. Examples with high CM scores are mostly datapoints showing natural variation in translations, and are not simply misaligned training data. When it comes to idioms, frequency influences whether or not NMT models manage to memorise and paraphrase them (chapter 6), but only for idioms whose non-compositionality is preserved within the training material.

Moreover, **(2) what is thought of as requiring memorisation, is not necessarily what is memorised by models within a standard training regime**. In chapter 3, we studied what models actually memorise within a rather standard experimental setup. That does not necessarily mean that examples that we want models to memorise *are* memorised. In general, NMT systems do not show adequate memorisation of idioms (which, to some extent, also applies to non-compositional noun compounds and proverbs), since they translate them mostly in a word-for-word fashion (chapter 6). This can be explained based on surface-level features of these examples, such as a low word overlap or a high edit distance between the source and its backtranslated target.

Which model-internal mechanisms enable memorisation?

With respect to RQ2, **(3) memorisation is not a process that can be easily localised to individual layers or parameters but is a cooperative process of many layers**. We investigated this in chapter 4, where we performed localisation of the memorisation of mislabelled examples. By applying four localisation methods to four fine-tuned transformer-based LMs on twelve tasks, we established that none of the individual layers implement memorisation of the noisy labels. Instead, many layers cooperate to gradually move noisy examples towards their newly assigned label, and we made this visually explicit through our centroid analysis. Contrary to what a subset of previous work suggests, models' deepest layers do not play a special role here. We did identify a subtle influence of the classification task investigated, where memorisation shifts up or down in the model. This effect is correlated with models' generalisation performance on unseen data, suggesting that the more distinct memorised examples are from the remaining datapoints, the more separation occurs internally of performing the main task and encoding the memorised noisy label.

In chapter 6, we identified that **(4) mechanisms for paraphrasing memorised idioms in translation involve 'grouping' on the source side and disconnecting from the encoder on the target side**. To identify this, we contrasted the attention distributions of figurative, paraphrased (and thus memorised) idioms to literal expressions translated word for word, using NMT systems from seven Indo-European language pairs. We found increased attention within the idiom and reduced interaction with the context – i.e. idioms that are paraphrased are processed more as one non-compositional unit than the literal phrases. When paraphrasing, the decoder attends less to the idiom's tokens on the source side, directing attention to the EOS token instead. In line with chapter 4, we observed that hidden representations for memorised idioms gradually become more distinct over the course of the encoder's layers rather than suddenly standing out in one particular layer, emphasising that idiom memorisation similarly is a distributed process.

To what extent are memorisation and generalisation at odds with one another?

With respect to RQ3, we established that **(5) memorisation of atypical examples can be beneficial for models' generalisation capabilities, presumably due to**

the variation that characterises natural language. The fact that memorisation, in general, occurs in neural networks trained on natural language tasks is beneficial for their generalisation capabilities. We demonstrated this for NMT, where both experiments that leave out subsets of the memorisation continuum, or solely train on subsets of the memorisation continuum, show a benefit of examples with higher CM scores (chapter 3). This conclusion is, however, specific to the amount of noise contained in the dataset and how much of that noise models memorise over the course of training. In NMT, even with 1M training examples, the misaligned examples were not memorised, but it is not a guarantee for all tasks and setups that memorisation will always be beneficial.

In chapter 5, we determined that **(6) idiom acquisition is a multi-phase process involving overgeneralisation first, and memorisation second** by training English-Dutch NMT systems and tracing translations of 20 idioms over the course of training. Although this is a natural consequence of the non-compositional nature of idioms, we were the first to demonstrate this for idioms in a natural language context. If a part of the experimental setup involves model selection based on the convergence of standard evaluation metrics, the models selected can show inadequate idiom memorisation when generalisation and memorisation do not temporally align. For high-resource idioms, models indeed transition to the second phase of emitting memorised paraphrased translations, but we also identified in chapter 6 that for a wide range of idioms, models remain in that overgeneralisation phase, emitting overly compositional translations.

Finally, we discussed generalisation not just as quantified using standardised or IID test sets, but also for compositional generalisation, by redefining three tests from the literature for the scenario of English-Dutch NMT systems trained on natural language corpora in chapter 5 (as opposed to synthetically generated datasets as done by most related work). This allowed us to determine that **(7) transformer does not treat inputs in a locally compositional manner, which might be beneficial for memorisation, but is detrimental for compositional generalisation**. Memorisation of paraphrased idiomatic translations requires treating an idiom as one unit and using the context to disambiguate whether its meaning is figurative or literal. This stands in opposition to the notion of strong or local compositionality, where the meanings of phrases are composed bottom-up. Our NMT systems showed large volatility in our systematicity and substitutivity tests, which showed translation changes following small input modifications. This occurred both when the modification was linearly and syntactically close to changes in the translation (e.g. in the `S→NP VP` condition of systematicity), and when the modification was further away (e.g. in the `S→S CONJ S` condition of systematicity). That volatility harms the robustness of translation systems, and, ideally, models should learn to better modulate when to apply local and global compositional processing.

Summarising, the findings suggest that while transformer-based models memorise a great deal about their training data, and that even though this is beneficial given the natural variation observed in test data, they simultaneously do not memorise enough of what they should when it comes to formulaic language. Paradoxically, transformer is not compositional enough and too compositional at once, while locally and globally compositional strategies would be beneficial for different types of inputs. Mechanisms for memorisation emerge naturally, but in a dispersed manner throughout many layers, and these mechanisms are not mature enough to cope with language's formulaic nature.

Going forward

Based on the findings contained in this thesis, I encourage future work to explore four directions. Below, I give concrete suggestions for what to focus on and warn about the potential dangers of state-of-the-art approaches in NLP.

Towards an explicit incorporation of (non-)compositionality That compositional generalisation does not simply emerge when training standard transformers has been underscored by the wide range of modelling techniques proposed to improve compositional generalisation (some of which were mentioned in §2.3.2 and §5.5.2). Specialised architectures that improve upon this are, however, primarily designed with a local compositionality assumption: meaning can be composed bottom-up. This will improve compositional generalisation, but can be detrimental for non-compositional generalisation. I firstly encourage papers on improving compositional generalisation for natural language to run auxiliary tests on non-compositional phenomena, monitoring situations in which their method might *not* be beneficial. Secondly, instead of assuming both types of generalisation emerge naturally, I encourage modelling techniques that explicitly distinguish the two types of processing, using separate compositional and non-compositional representations in line with the *two-track mind* that has been discussed for humans' compositional and formulaic processing (van Lancker Sidtis, 2012), or in Baggio (2021)'s proposal for a parallel mechanism. An example of a technique like that was proposed by Zeng and Bhat (2023). Even without designing new architectures, the dual nature of natural language can be explicitly incorporated, e.g. through compositionality-aware prompting. For the overgeneralised `GPT4o-mini` idiom translation provided in §6.6.2, simply varying the prompt in the following three ways yields a correct translation:

- "Translate from English to Dutch, literally translating where possible and figuratively translating where needed."
- "Translate from English to Dutch. First identify the formulaic language contained in the input, and only then translate."
- "Translate from English to Dutch. Translate compositionally except for figurative phrases."

Towards memorisation circuits Our attempts at localising the memorisation of mislabelled examples and examining internal mechanisms that support non-compositional idiom translations emphasised the distributed nature of memorisation. Similarly, much of the related work on the implementation of memorisation elaborated on in §2.2 was focused on identifying the individual layers or even neurons that encode specific information, such as a factual relation between named entities. If memorisation is not local, but a cooperative process of many layers and mechanisms, we should move towards a more holistic understanding of how memories are encoded from the token embeddings up to the final layers. In mechanistic interpretability, a wide range of recent studies have identified circuits that encode a particular phenomenon, such as greater-than computations in arithmetic (Hanna et al., 2023) and indirect object identification (Wang et al., 2022). I encourage future work to adopt a similar approach for memorisation. At first glance, circuit discovery may seem at odds with memorisation, as the success of the discovery is measured based on the generalisation to new inputs encoding the same general phenomenon with new instantiations. Yet, with modifications to the experimental setup, this could be a fruitful way forward, both for memory discovery and memory editing. One could, for example, measure success as the ability to recall the memory in various contexts and to improve the recall by modifying activations within the circuit, while ensuring that compositional variations of a non-compositional memory do not activate the circuit. If memorisation is not local, neither should localisation, unlearning, or editing be local.

Towards a memorisation curriculum Research on memorisation in LLMs predominantly emphasises the sheer volume of training examples that LLMs memorise (e.g. Carlini et al., 2022; Nasr et al., 2023) for both open- and closed-source models, and this increases as model sizes continue to grow. Alongside research quantifying memorisation, surveys have appeared that summarise the different memorisation metrics, the different types of data memorised, and modelling techniques that rely on these findings, such as privacy-aware training, unlearning or model editing (Hartmann et al., 2023; Usynin et al., 2024; Wei et al., 2024; Satvaty et al., 2024). The emphasis tends to be on privacy, security, and copyright risks, with less work on distinguishing beneficial from harmful memorisation or linking memorisation to task performance. I do not mean to downplay the risk associated with memorisation, but would like to underscore that memorisation can be beneficial when it comes to, among other things, formulaic language, factual information, copyright-free lyrics and books, influential quotes, named entities and infrequent words and phrases. In an ideal scenario, we would move towards a memorisation curriculum, pre-identifying what we would and would not want to be memorised, oversampling the former and obscuring the latter datapoints. This would be too much to ask for state-of-the-art LLMs, whose pretraining data is unavailable or of a scale that virtually makes pretraining modifications infeasible. However, we could start to move towards this for initiatives with more control over the pretraining corpus, such as the BabyLM challenge (Warstadt et al., 2023; Hu et al., 2024).

On the dangers of being overly compositional Because frequency influences idiom acquisition, it is reasonable to assume that scaling pretraining corpora may partly address transformer's shortcomings for idioms and non-compositional phrases more generally. Yet, as discussed in §6.6.2, issues with overly literal translations are still being reported for commercial and non-commercial (translation) systems, and these issues will be exacerbated for low-resource languages. I recommend caution when it comes to reliance on systems to synthesise training corpora (e.g. for the purpose of distilling larger models into smaller models, Wu et al., 2024), or augment or auto-annotate datasets (e.g. Ubani et al., 2023; Li et al., 2023, 2024c; Kim et al., 2024). LMs tend to underestimate probabilities of sequences from the long tail (LeBrun et al., 2022), and when training repeatedly on model-generated data, LLMs lose information about the long tail of the original data distribution (Shumailov et al., 2024). As a result, many non-compositional patterns may risk fading as synthetic data enters general-purpose, internet-based corpora. In §3.4.2, we noticed a similar issue of overly literal proverb translations in our NMT training corpus – possibly due to subpar NMT outputs circulating online. Which parts of natural language are we losing if transformer's own predictions become a part of that language? This warrants carefully crafted investigations, such as measuring the prominence of formulaic language in real and synthesised corpora.

Appendix A

Supplementary material for chapter 3

A.1 Technical setup

Before commencing training, we tokenise the data using the Moses tokeniser,[1] and then compute tokens using BPE[2] (Sennrich et al., 2016) to create a joint vocabulary per language pair, with a size of 64k tokens. Model training was performed using `fairseq`, version 0.12.1.[3] We train `transformer-base` models using the following setup, with the number of total training steps being dependent on the experiment conducted:

- To obtain memorisation scores in §3.2, we trained for 100 epochs on training datasets of 500k sentence pairs. This involves model training beyond the point of convergence to investigate memorisation.
- The remaining models discussed are all trained for up to 50 epochs.

We train using the following command, modelled after exemplar Fairseq translation setups. We did not further tune hyperparameters but did increase `max-tokens` to better utilise the GPU capacity.

```
fairseq-train <DATA_DIR> \
    --arch <MODEL> --save-dir <MODEL_DIR> --share-all-embeddings \
    --fp16 --max-update 200000 \
    --optimizer adam --adam-betas '(0.9,0.98)' --clip-norm 0.0 \
    --lr 0.0005 --lr-scheduler inverse_sqrt \
    --warmup-updates 4000 --warmup-init-lr '1e-07' \
    --label-smoothing 0.1 --criterion label_smoothed_cross_entropy \
    --dropout 0.3 --weight-decay 0.0001 \
    --max-tokens 10000 --update-freq 2 \
```

[1] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl
[2] https://github.com/rsennrich/subword-nmt
[3] https://github.com/facebookresearch/fairseq

```
--save-interval 50 --max-epoch <MAXEPOCH> \
--seed <SEED> --validate-interval 5 \
--eval-bleu --eval-bleu-args '{"beam":5}' --eval-bleu-remove-bpe
```

Tesla V100-SXM2-32GB GPUs are used for model training in §3.2. We train each model on a single GPU, on which one epoch of a 500k training set lasted up to 4 minutes, and full training approximately 6 hours. Training all seeds for the five language pairs thus costs 1.2k GPU hours. In §3.6 we train 3 seeds for 54 coordinates using NVIDIA A100-SXM-80GB GPUs, and the training of one model can take up to 2.5 hours. This thus cost approximately 400 GPU hours.

Regression models for memorisation proxies In §3.5 we trained MLPs to predict TM, GS and CM values. The MLPs were trained for 20 epochs maximum, using Adam according to default hyperparameters in the `sklearn.neural_network.MLPRegressor` class. The MLP takes 28 inputs when training with features only, and 28 + 6 when adding the training signals, and has two hidden layers of 100 units. It predicts TM, GS and CM at the same time.

A.2 Datasets

We obtain the training data from the Tatoeba repository (version `v2021-08-07.md`), postprocessed to obtain parallel corpora as detailed in §3.2. The Tatoeba repository has the license `Attribution-NonCommercial-ShareAlike 4.0 International`, which allows us to use and redistribute the data, given appropriate attribution.

The compound dataset of Tayyar Madabushi et al. (2021) used in §3.4 is available under the `GNU General Public License v3.0` license that allows usage, modification and distribution. The MAGPIE dataset is available under the `Creative Commons Attribution 4.0 International` license.

Datapoints' surface-level features In §3.3 we analysed how datapoints' surface-level features correlate with memorisation metrics. To do so, we compute 28 language-independent features that, together, we believe cover a broad spectrum of surface-level features from both the source and target. The following features describe the source or target length or frequency, and are computed once for white-space tokenisation and once for BPE tokenisation:

1-4. Source / target length;
5-8. Average log frequency of source / target tokens;
9-12. Minimum log frequency of source / target tokens.

The following features, too, are computed over the source or target only:

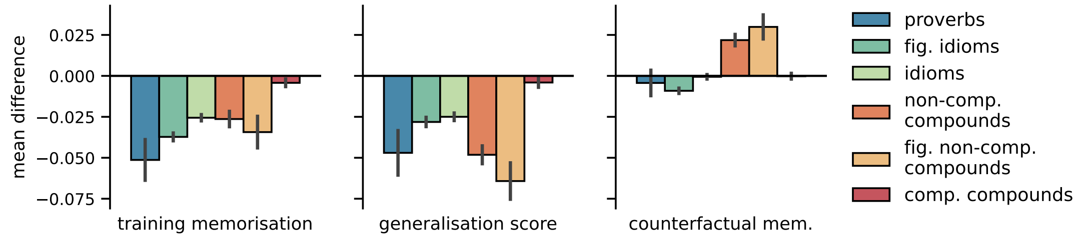
13. Number of repetitions of this target. Note that the 1M source sequences are unique, but there could still be repeated targets if the same target maps to multiple source sequences;
14,15. Segmentation of the source (or target): $1 - \frac{|s_{ws}|}{|s_{BPE}|}$, 0 means no segmentation beyond the token level;
16. Digit ratio: how many digits are included in the source;
17. Punctuation ratio: how many tokens in the source are punctuation.

The remaining features capture source-target interactions:

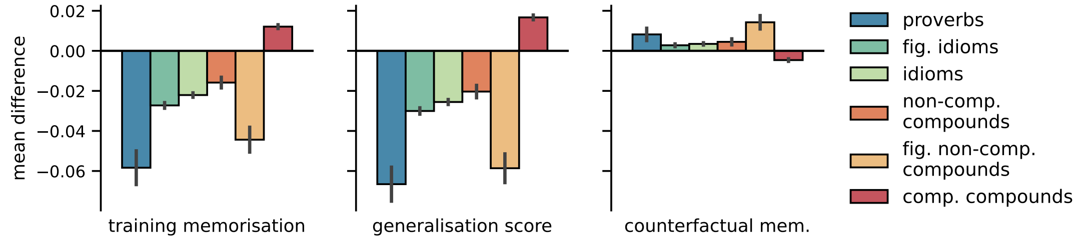
18. Token-level Levenshtein edit-distance between the source and the target;
19,20. The length ratio between the source and the target: $\frac{|s|}{|t|}$ (2x, BPE tokenised and white space tokenised);
21. Comparison by backtranslation, obtained with `Marian-MT` models trained on OPUS by Tiedemann and Thottingal (2020), by computing the token-level Levenshtein edit-distance between the source and the backtranslated target;
22,23. $|s| - |t|$ (2x, BPE tokenised and white space tokenised);
24,25. Ratio of unaligned source / target words, alignments are obtained with `eflomal` (Östling and Tiedemann, 2016);
26. Alignment monotonicity, computed as the Fuzzy Reordering Score (Talbot et al., 2011), implementation obtained from Voita et al. (2021);
27. Token overlap: the ratio of tokens from the source that also occur in the target;
28. Word overlap: similar to token overlap, but excluding punctuation.

A.3 Formulaic phrases revisited

In §3.4, we observed that sentences containing formulaic phrases (proverbs, idioms, non-compositional compounds) have lower TM and GS compared to control stimuli. However, following the discussion first introduced in §2.3.3 concerning the hypothesis that non-compositional phrases may have more accurate paraphrased translations than compositional phrases, we could entertain an alternative hypothesis. What if the differences are merely due to inadequate evaluation metrics, which cannot properly quantify the TM or GS scores, due to over-reliance on the exact surface forms of individual target translations, thus ignoring the existence of paraphrases? The LL metric is based on target tokens' probabilities, but if paraphrases exist for a particular phrase, models may spread the probability mass over multiple paraphrases. In that scenario, LL would underestimate the memorisation of the underlying meaning of the formulaic phrase. Although the problem of paraphrased translations not being adequately assessed by evaluation metrics is not unique to formulaic language, it is likely exacerbated due to the non-compositional nature of certain formulaic phrases.



(a) LL-based



(b) COMET-based

Figure A.1: Differences in memorisation scores (computed using two underlying metrics) when comparing formulaic stimuli to control stimuli of the same source and target length, for En-Nl. Error bars show the standard error.

We cannot possibly enumerate all paraphrased translations of a certain phrase to fully reject this hypothesis. Yet, we can analyse our existing data in a new light. To do so, we take the generated hypotheses for OPUS training data from the 40 En-Nl models – the hypotheses were previously used for the BLEU-based memorisation map (§3.3.2) – and compute TM, GS and CM based on the neural quality estimator COMET-22 (Rei et al., 2022).[4] In this scenario, we thus do not rely on the exact surface-form translation of a formulaic phrase that was included in the corpus, but rely on the neural quality estimator that is expected to be aware (to some extent) of the underlying semantics. Using these new scores, we can again measure the difference between formulaic phrases and control stimuli, as is displayed in Figure A.1. In this scenario, too, the formulaic phrases have lower TM and GS scores than control stimuli. The largest differences are that compositional phrases now score slightly higher than controls, and that the formulaic stimuli now all have *slightly* higher CM scores compared to controls. Overall, this suggests that the difference between formulaic and control stimuli was not simply due to the LL metric relied on in the main text.

[4]Specifically, `wmt22-comet-da` with `comet` v1.2.0. Note that COMET-based scores are not necessarily suited for direct comparisons to our memorisation maps; for instance, due to scores of 0 and 1 being near-impossible to obtain and empty hypotheses receiving non-zero scores. We merely use it to test this hypothesis, and do not think it can be used to quantify memorisation out of the box.

Appendix B

Supplementary material for chapter 4

B.1 Technical setup and datasets

We ran the experiments for the 12-layer models on NVIDIA GeForce RTX 1080/2080 Ti GPUs. We train the small models using a batch size of 8 (due to GPU restrictions, or 4 in the few cases where we still get memory errors, which happens for `Reuters`, in particular) and an initial learning rate of 1e-5 for 50 epochs, capping sequences at 512 tokens. 50 epochs is beyond the point of convergence since the aim is to investigate memorisation rather than optimise models for their generalisation capabilities. For the models from §4.2.4 where the main task can only modify two layers at a time, we rerun training with an increased learning rate if the training accuracy does not exceed .99. For every model trained, we store checkpoint θ_{M_1} when the training accuracy exceeds .993, and store checkpoint θ_{M_2} at the end of training. The most time-consuming experiments are model training and layer retraining:

- §4.2.4: 11 datasets × 3 control setups to obtain θ_M + 11 datasets × 3 control setups to obtain θ_O + 11 datasets × 1 frozen model = 77 setups trained for each of the 4 models, taking 1 - 6 hours each

- §4.3: 12 datasets × 3 seeds for θ_M + 12 datasets × 3 seeds for θ_O + 12 datasets × 1 frozen model = 84 setups trained for each of the 4 models, taking 1 - 6 hours each
  Layer retraining: 12 datasets × 3 seeds θ_M × 78 windows = 2808 setups trained for each of the 4 models, taking 3 to 45 minutes each

The experiments discussed in §4.5.1 are run on NVIDIA A100-SXM80GB GPUs. `OPT-1.3B` is trained with an initial learning rate of 5e-6 and a batch size of 32 or 16. We train two models per dataset (θ_M and θ_O), and individual training runs take

45 minutes to 6 hours, depending on the dataset. Visit our codebase here: https://github.com/vernadankers/memorisation_localisation.

We use the `transformers` library[1] to obtain the models/tokenisers and train them, implementing the remaining analyses ourselves.

Models The licenses of all models, which are `Apache 2.0` (`BERT-base`), a custom license for `OPT` models[2] and the MIT License (`Pythia-160m`, `GPT-Neo-125m`) allow non-commercial use for research purposes.

Datasets The datasets contained in GLUE and SuperGlue are available under licences that allow use and redistribution for research purposes (Wang et al., 2019b,a). `Stormfront` is available under `CC-by-SA-3.0`; `ImplicitHate` is not explicitly assigned a license, but the corresponding repository is available under the `MIT` license; `Reuters` is available under the `CC-BY-4.0` license; for `TREC` the license is unknown, and `Emotion` should be used for educational and research purposes only, and has no license, otherwise[3].

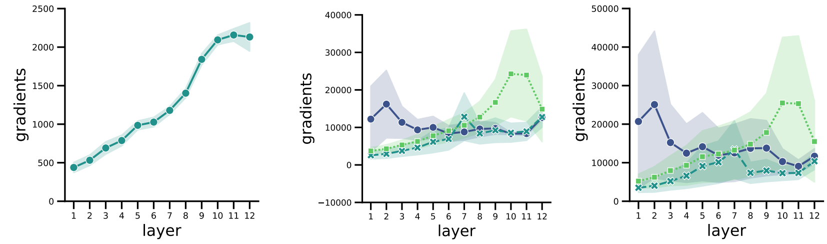
B.2 Postprocessing gradients

As described in §4.2, forgetting gradients are one of the signals we examine to perform memorisation localisation. We average them over all noisy examples, or over a similar amount of clean examples. Preliminary experiments indicated that, taken at face value, the gradients do not necessarily pinpoint the correct layers in a control setup. Using two validation tasks (`MRPC` and `TREC`), we consider taking the L_1-norm or the L_2-norm over gradients and applying two ways of normalising the forgetting gradients of the noisy examples: i) subtract the forgetting gradients of clean examples, ii) normalise the per-layer norm by the norm obtained using a frozen model. The final post-processing step applied afterwards is that the weights of the 12 layers are normalised to sum to 1 to allow for the computation of the M-CoG coefficients, and to reduce variation among tasks.

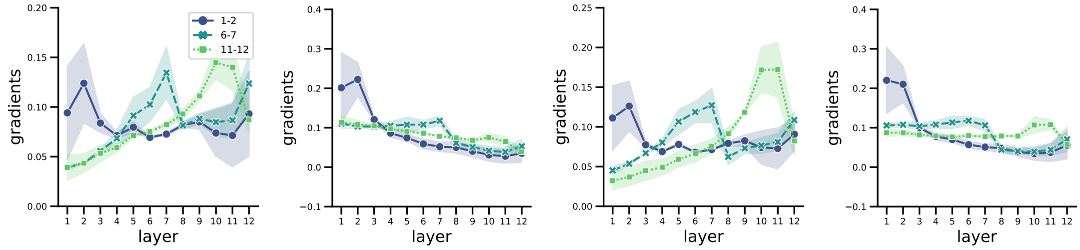
Figure B.1a illustrates the L_1-norm for 'forgetting' gradients for a frozen `BERT`, that tend to point to the final layers; Figure B.1b and Figure B.1c demonstrate forgetting gradients for clean and noisy examples in the control setup. Both point to similar layers, but the norms are higher for noisy examples.

Figure B.1d-B.1g *do* apply the within-dataset normalisation that normalises layer weights to sum to one. Figure B.1d again demonstrates for noisy examples that without further post-processing, the gradients overestimate the relevance of later layers in `BERT`. Both post-processing steps i) and ii) dampen that. When measuring the success of the

[1] https://huggingface.co/docs/transformers
[2] https://github.com/facebookresearch/metaseq/blob/main/projects/OPT/MODEL_LICENSE.md
[3] https://github.com/dair-ai/emotion_dataset



(a) Frozen model gradients (b) 'Clean' gradients (c) 'Noisy' gradients



(d) 'Noisy' gradients, normalised (e) Divide by frozen, normalised (f) Subtract clean, normalised (g) Divide, subtract, normalised

Figure B.1: Effect of the gradient analysis postprocessing steps on the MRPC and TREC tasks for the BERT model when using the L_1-norm.

post-processing steps using the accuracy metric, included in Table B.1, the combination of both is most successful at recovering the layers in which memorisation had taken place in the control setups, and the L_1-norm leads to more accurate results than the L_2-norm.

These post-processing steps improve the accuracy for all models but Pythia. Across the board, applying both steps i) and ii) and using the L_1-norm yields the highest accuracy, so we apply both of these steps in chapter 4.

		Pythia		GPT-N		BERT		OPT	
subtracing clean	**norm. frozen**	L_1	L_2	L_1	L_2	L_1	L_2	L_1	L_2
×	×	0.08	0.08	0.58	0.25	0.58	0.50	0.58	0.17
×	✓	0.08	0.08	0.67	0.42	0.50	0.50	0.50	0.42
✓	×	0.08	0.08	0.58	0.25	0.58	0.50	0.67	0.33
✓	✓	0.08	0.00	0.75	0.25	0.75	0.58	0.58	0.50

Table B.1: Effect of the gradient analysis postprocessing steps on the MRPC and TREC tasks, measured as the average accuracy of the highest scoring layers.

Appendix C

Supplementary material for chapter 5

C.1 Dataset and preprocessing

Training data Our training data contains the English-Dutch subset of the MT corpus OPUS (Tiedemann and Thottingal, 2020). OPUS is a continually growing resource; we used the subset as provided by Tiedemann (2020). This dataset contains 69M source-target pairs, and can be found on `https://github.com/Helsinki-NLP/Tatoeba-Challenge/blob/master/data/README-v2020-07-28.md`.[1]

Preprocessing We tokenise the data using the tokenisation script from the SMT library Moses. Following the number of subwords suggested by Tiedemann (2020), we generate a subword vocabulary applying 60k BPE merge-operations. To do so, we use the `learn_bpe.py` script provided in the `subword_nmt` repository.

Different corpora We train models on three different sizes of corpora: SMALL, MEDIUM and FULL. To generate these corpora, we first shuffle the OPUS training data using the bash function `shuffle`. To generate the SMALL and MEDIUM corpora, we take the first 8582811 and 1072851 sentences of this shuffled corpus, which corresponds to $\frac{1}{8}$th and $\frac{1}{64}$th of the full training corpus, respectively. For each setting, we train models with seeds $\{1, 2, 3, 4, 5\}$.

Test and validation data Initially, we aimed to evaluate our models using the commonly used MT test sets OPUS-100 and the test partition of the TED talk corpus. However, it turned out that both these test sets were almost fully contained in our training corpus. We, therefore, adopted the newer FLORES-101 corpus (Goyal et al., 2022),

[1] The Tatoeba repository has the license `Attribution-NonCommercial-ShareAlike 4.0 International`, which allows us to use and redistribute the data, given appropriate attribution.

of which we used both the 'dev' and the 'devtest' set. To compute BLEU scores, we tokenised the data with the Moses tokenisation script mentioned above, and then used the commandline script `fairseq-generate` to compute scores.

We furthermore use several evaluation sets to assess the compositional abilities of our trained models. The data for these tests, as well as scripts to run them and visualise the results, can be found in our `compositionality_paradox_mt` GitHub repository.

Semi-natural templates The semi-natural data that we use in our test sets is generated with the library `DiscoDOP`, developed for data-oriented parsing (Van Cranenburgh et al., 2016). We generate the data with the following seven step process:

Step 1. Sample 100k English OPUS sentences.

Step 2. Generate a treebank using the disco-dop library and the `discodop parser en_ptb` command. The library was developed for discontinuous data-oriented parsing. Use the library's `-fmt bracket` to turn off discontinuous parsing.

Step 3. Compute tree fragments from the resulting treebank (`discodop fragments`). These tree fragments are the building blocks of a Tree-Substitution Grammar.

Step 4. We assume the most frequent fragments to be common syntactic structures in English. To construct complex test sentences, we collect the 100 most frequent fragments containing at least 15 non-terminal nodes for NPs and VPs.

Step 5. Selection of three VP and five NP fragments to be used in our final semi-natural templates. These structures are selected through qualitative analysis for their diversity.

Step 6. Extract sentences matching the eight fragments (`discodop treesearch`).

Step 7. Create semi-natural sentences by varying one lexical item and varying the matching NPs and VPs retrieved in Step 6.

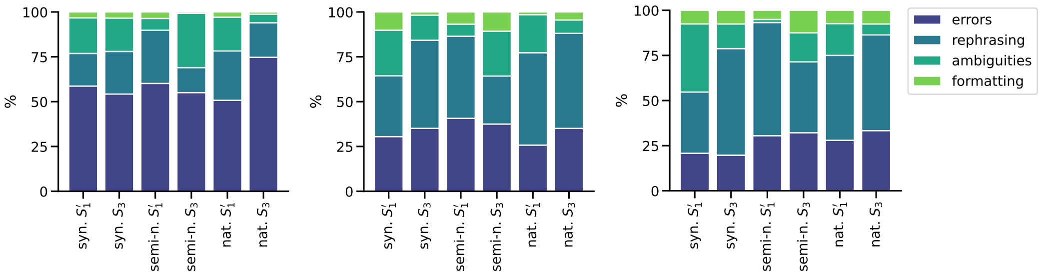
In Table 5.3, we provided examples for each of the ten templates used, along with the internal structure of the complex NP or VP that is varied in the template.

C.2 Technical setup

As reported in chapter 5, we focus on English-Dutch translation, and all our models are `transformer-base` models (Vaswani et al., 2017), as implemented in `fairseq` (Ott et al., 2019). Specifically, we used the implementation as it was on May 12, 2021. With our vocabulary, the models have a total of around 80M trainable parameters.

To train our models, we follow the training procedure suggested by Ott et al. (2018).[2] To summarise, we share all embeddings between the encoder and the decoder, use Adam

[2] https://github.com/facebookresearch/fairseq/tree/d151f2787240cca4e3c7e47640e647f8ae028c37/examples/scaling_nmt



(a) Small training set (b) Medium training set (c) Full training set

Figure C.1: Distribution of error types for sentences that contain inconsistencies in systematicity, detailed per model trained on the training set sizes in the subcaptions.

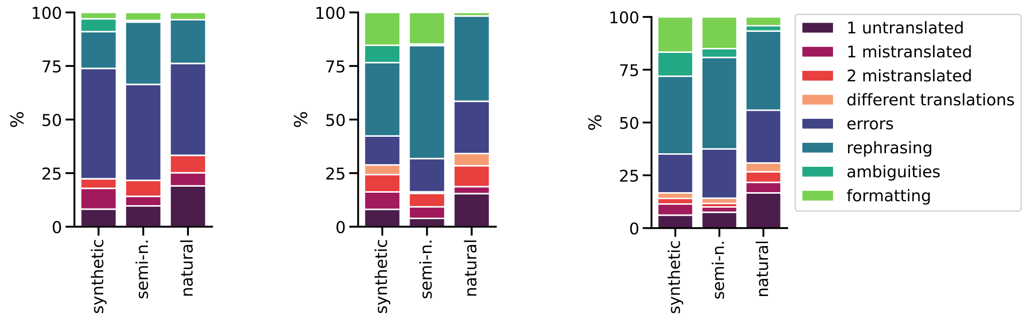
as optimiser with β-values (0.9, 0.98), starting from an initial warmup learning rate of 1e-07 for 4000 warmup updates and a learning rate of 0.0005 afterwards, using inverse square root as the learning rate scheduler. We use a clip-norm of 0.0, dropout of 0.3, weight-decay of 0.0001, label-smoothing of 0.1. The maximum number of tokens in a batch is 3584, we simulate larger batches by increasing the update frequency to 8. To determine early stopping, we use a patience of 10 (i.e. we stop training if a model does not improve on the dev set anymore for 10 epochs, and take the best checkpoint at that point). Any other hyperparameters involved follow the `fairseq` default.

Compute The main experiments were ran using Tesla V100 GPUs. Training a `transformer-base` model on our small, medium and full datasets takes on average 3.5, 17 and 113 minutes per epoch, respectively, on 32 GPUs. This makes the total training time for these models, which are trained for around 160, 60 and 30 epochs, 10, 17 and 56 hours, respectively (again, spread over 32 GPUs). In §5.3.3, for the 'going further' discussion, we reproduce the models from the main experiments using one model seed on NVIDIA GeForce RTX 1080 Ti GPUs, using 4 in parallel. We adjust the update frequency to simulate the same setup as previously mentioned.

C.3 Manual analysis

In chapter 5, we ran our tests for compositional generalisation. We focused on models' consistency under input perturbations, but these automated tests could not distinguish harmful inconsistencies from benign ones. We complemented these tests with an elaborate manual analysis, which provided more insight into the nature of the non-compositional behaviour we registered. We previously included the setup for this analysis and a summary of the results in §5.4. This section further elaborates on the results.

We include the results in Figure C.1 for systematicity and Figure C.2 for substitutivity. As a general trend, the results reflect that in models trained on smaller datasets, more mistakes are actually errors, rather than multiple correct alternatives. In the



(a) Small training set (b) Medium training set (c) Full training set

Figure C.2: Distribution of the types of inconsistencies observed in the substitutivity test, detailed per model trained on the training set sizes in the subcaptions. The purple colour scheme represents error types specific to this experiment.

systematicity test, 59% of the inconsistencies for the models trained on the smallest dataset are erroneous changes, versus 34% and 27% in the models trained on the medium and largest datasets, when we average the percentages over the different subsets annotated. For substitutivity, the percentage of erroneous changes unrelated to the synonyms comprises 46%, 18% and 22% for the small, medium and full datasets, respectively. On top of that, there were inconsistencies related to the synonyms, which represented 26%, 26% and 21% for the three dataset sizes, respectively. While this is expected, to some extent, it still constitutes a problem: for models trained on smaller amounts of data, being able to translate in a compositional manner is particularly relevant. Below, we further elaborate on the types of inconsistencies encountered per annotation category, including some examples.

Rephrasing A large portion of the inconsistencies concerns pairs where one translation can be considered a rephrased version of the other translation. A common cause of this is a **reordering of words** that does not impact the grammaticality or meaning of the Dutch sentence – e.g. in sentences with adverbs ("heeft de burgemeester zeker in de gaten" vs "heeft zeker de burgemeester in de gaten") or relative clauses with direct objects ("die genieten van de vakantie" vs "die van de vakantie genieten"). We could not trace these reorderings back to the specific change made in the systematicity or substitutivity tests. Consider, for instance, Example (1), where the reordering happens as a consequence of changing the word "king" to "father". Note also that while these translations both contain an error ("neemt . . . in de gaten"), this is not marked as an inconsistency, because it is shared between the translations.

(1) s EN: The aunts criticise the {king, father}, and the man definitely observes the mayor.

t_1 NL: (. . .) en de man neemt zeker de burgemeester in de gaten.

t_2 NL: (...) en de man neemt de burgemeester zeker in de gaten.

Another common type of rephrasing is one where the two translations include (nearly) **synonymous terms** in Dutch. Some examples are the translations of athlete ("sporter" vs "atleet"), wish ("wensen" vs "willen") and observe ("observeren" vs "waarnemen"). Some of them can appear in the same context but for others the two words would typically appear in different contexts. For instance, the word "dokter" is used in more informal contexts than the word "arts" (both translations of "doctor"). Again, we could not identify an interpretable pattern for when the model emits one instead of the other.

Source ambiguities An intriguing category that we had not anticipated were cases in which the source sentence contained ambiguities, such as **polysemous words** (e.g. "director" translated to "directeur", referring to the director of a company, and "regisseur", indicating the director of a movie). Other ambiguities encountered were **scope ambiguities**, which were particularly prominent for the systematicity test. In that test, we concatenate two sentences, and the ambiguity was often related to the verb in the first sentence – e.g. in "The friend wishes that the {lawyers, directors} scream, and the victims (...)". While we intended this to be a conjunction of two independent sentences, there is also a reading where "wishes" takes scope over the entire second conjunct. In Dutch, those two cases are distinguishable because they trigger a different word order in the embedded clause (SOV), which is not grammatical for main clauses. Such scope changes often lead to very questionable interpretations of the English sentence, as is the case for the source sentences in Example (2):

(2) s_1 EN: The victims want that the {doctors, mayors} run, and the victims read an article about the case of a procedure which includes a repayment plan.
s_2 EN: The women wish that the {painters, victims} walk consciously, and every 2CV or Dyane can basically be used as a donor.

Interestingly, we also observed SOV word order in the second conjunct when a scope change was not possible, for instance, when the second conjunct was a question, or the verb in the first sentence did not allow to take scope over the second conjunct without the presence of the word "that". See Example (3). We underline the incorrect part of the translation, here and in erroneous examples that follow. These examples indicate that the interpretation of scope change might not be applicable here and that instead, the model is applying some heuristic where particular words trigger an SOV order.

(3) s EN: The victim observes the {leader, king}, and the fathers carefully avoid the president.
t_1 NL: Het slachtoffer observeert de leider en de vaders

de president zorgvuldig vermijden.

t_2 NL: Het slachtoffer observeert de koning en de vaders vermijden voorzichtig de president.

Single word target errors In the category 'target errors', single-word errors are often due to missing, wrongly translated or untranslated words. Changes due to **missing words** can be very minor but nevertheless render one of the translations ungrammatical (e.g. "De tante achter de truck bewonderde de directeur", correct, vs "De tante achter de truck bewonderde directeur", incorrect), or semantically incorrect. **Untranslated source words** being present in one of the two translations occurred for our synthetic templates (e.g. "uncles"/"ooms", "butchers"/"slagers") but also with words from the natural sentences (e.g. "extrusion"/"extrusie", "soils"/"bodem"). Lastly, we observed cases of **mistranslated words**, where words unrelated to the input perturbation received a wrong translation in one of the two sentences but a correct one in the other, for example: "poets" being translated as "dichters" (correct) vs "de potten" (incorrect), or "general" as "generaal" (correct) vs "wandeling" (incorrect).

Multi-word target errors Other types of errors are less easily located to individual words but indicate an overall misinterpretation of the input, such as the **change in agreement** displayed in Example (4). In this particular case, the source of confusion is explainable: "begrijpen" should agree with "schilder" but instead agrees with the word "doctors", much earlier in the sentence. A more locally compositional approach to translating would have yielded the correct translation.

(4) s EN: The doctors that laugh admire the {president, baker}, and the painter that admires her understands the king.

t_1 NL: (...) de schilder die haar bewondert, begrijpen de koning.

t_2 NL: (...) de schilder die haar bewondert begrijpt de koning.

Another example of a multi-word error relates to the **semantic role assigned to agents**. For instance, in Example (5), "the fathers" is removed from the main clause and moved into the relative clause, leaving "read" without its agent.

(5) s EN: The group of painters behind the truck forgets the {president, friend} and an article about the previous EESC Opinion on alcohol related harm, which looked at f, is read by the fathers.

t_1 NL: (...) en een artikel over het eerdere advies van het EESC over alcoholgerelateerde schade, die door de vaders wordt onderzocht, wordt gelezen.

t_2 NL: (...) en een artikel over het eerdere advies van het EESC over alcoholgerelateerde schade, die naar f uitkeek, wordt door de vaders gelezen.

Formatting We marked inconsistencies as formatting changes if they were related to punctuation, capitalisation, hyphenation or the usage of spaces. Most of these inconsistencies were related to comma usage: in one translation, a relative clause or two conjuncts were separated by a comma, whereas in the other translation the comma was left out. When it comes to space usage ("tumormassa" vs "tumor massa"), there is a slight difference in correctness: in Dutch, compound nouns are not separated by spaces. Given how minor these mistakes are, we did not mark them as errors. Example (4) above provides an example for inconsistent usage of commas. Formatting changes are, relatively speaking, more prominent in models trained on larger training corpora.

Inconsistencies in synonym translations The synonym errors are subdivided into cases where synonyms are simply translated differently (we observed this mostly for the models with larger training set sizes), cases where both translations were incorrect, cases in which only one translation is wrong, and cases in which one synonym was not translated but directly copied from the source. Sometimes, the changes were quite peculiar, to give some examples from our natural corpus:

(6) s EN: The child admires the king that eats the {doughnut, donut}.
t_1 NL: Het kind bewondert de koning die de donut eet.
t_2 NL: Het kind bewondert de koning die de <u>ezel</u> eet.

(7) s EN: - Yeah, a barbecue sauce {moustache, mustache} contest.
t_1 NL: - Ja, een barbecue <u>[missing 'sauce']</u> met snor <u>[missing 'contest']</u>.
t_2 NL: - Ja, een barbeceu saus snor wedstrijd.

Some synonyms often remain untranslated (for "ladybird", "flautist"), some receive many different correct translations (for "shopping trolley"), yet others have very synonym-specific inconsistencies (e.g. "eggplant" being translated as "egg"+"plant", an interesting case because it reflects processing that is too local). For all synonyms – except for the model with the small training set that cannot translate "flautist" and "ladybug" – we have observed correct translations.

Further, it should be noted that while our substitutivity experiment provides insight into how the model copes with individual synonyms, the majority of the inconsistencies observed were unrelated to the synonym substitution. For instance, considering that the synonym changes were related to British and American spelling, and occasionally changed the tone of the sentence (e.g. "aeroplane" could be considered more archaic than "airplane"), one could anticipate changes in word choice in Dutch reflecting this change of style. However, the substitutivity inconsistencies were virtually indistinguishable from those annotated for systematicity.

Appendix D

Supplementary material for chapter 6

D.1 Technical setup and the dataset

The code for the analyses, along with a modified implementation of the models to allow for our analyses, is available via the `mt_idioms` GitHub repository. The models were pretrained by Tiedemann and Thottingal (2020) with the Marian-MT framework (Junczys-Dowmunt et al., 2018). The models are available via the Huggingface hub:

1. `https://huggingface.co/Helsinki-NLP/opus-mt-en-nl`
2. `https://huggingface.co/Helsinki-NLP/opus-mt-en-de`
3. `https://huggingface.co/Helsinki-NLP/opus-mt-en-da`
4. `https://huggingface.co/Helsinki-NLP/opus-mt-en-sv`
5. `https://huggingface.co/Helsinki-NLP/opus-mt-en-fr`
6. `https://huggingface.co/Helsinki-NLP/opus-mt-en-es`
7. `https://huggingface.co/Helsinki-NLP/opus-mt-en-it`

The translations and the corresponding attention patterns and hidden representations were extracted using a NVIDIA GeForce GTX 1080 Ti GPU; the remaining analyses are performed using CPUs only.

The MAGPIE dataset is available via the `MAGPIE` GitHub repository. The dataset is available under the `Creative Commons Attribution 4.0 International` license.

When applying COMET, we use `wmt22-comet-da`, using Comet v1.2.0.

D.2 Survey details

D.2.1 Crowd-sourcing annotations for Dutch

In an early phase of the research, the quality of the heuristic annotation method was estimated through a survey conducted using the Qualtrics platform by annotators from

MAGPIE	Predicted Translations								
	Paraphrase			*Word for word**			*Copy**		
	#	%	agr	#	%	agr	#	%	agr
Figurative	96	86	84	64	84	77	24	83	58
Literal	32	73	59	64	91	80	24	69	88

Table D.1: Survey statistics: the number of sentence pairs used (#), the % of labels the algorithm and annotators agreed on, and inter-annotator agreement. Agreement means an average of 4 annotators agreed on the label unanimously. *Categories merged in chapter 6.

Prolific. These annotators were native speakers of Dutch and fluent in English. To guard the quality of the data collection, participants went through a pre-screening process that consisted of a shorter version of the survey with three practice questions and seven regular questions. Participants were selected for the full study if they correctly answered practice questions, used all three of the labels (paraphrase, word for word, copy), and did not choose 'copy' if the keyword was clearly absent from the translation. The main survey consisted of three parts: (1) An explanation of what an idiom is, of potential literal and figurative usage of PIEs, the meaning of the three labels, and the format to be used in the study. (2) One practice exercise where three potential translations of one sentence had to be connected to the correct label. (3) Lastly, 38 questions were filled out: 12 instances that were figurative and were paraphrased by the model, 4 literal instances paraphrased by the model, 8 literal instances that were translated word for word, 8 figurative instances that were translated word for word, 6 copies (3 figurative, 3 literal).

If the participant indicated that it was a word-for-word translation, a follow-up question asked the participant to indicate the keyword's literal translation. We repeated the instruction of what constitutes a word-for-word translation since participants would often select the (conventionalised) idiomatic translation in the pre-screening phase – e.g. "handbereik" for "fingertips", for which a literal translation would be "vingertoppen".

Table D.1 summarises the results. The annotators and the heuristic method agreed in 83% of the cases. For 77% of examples, the annotations agreed on the label unanimously.

D.2.2 Collecting annotations for seven languages

Later on, the analyses were applied to heuristically annotated data for all seven languages. Postgraduate students from the University of Edinburgh were invited to annotate the data in exchange for payment, where one annotator annotated all 350 samples for a language. To reduce the cognitive load of the experiment, only sentences with ≤ 40 tokens were shown to the participants. The annotators were native speakers of the target language and were fluent in English, with the exception of the Swedish

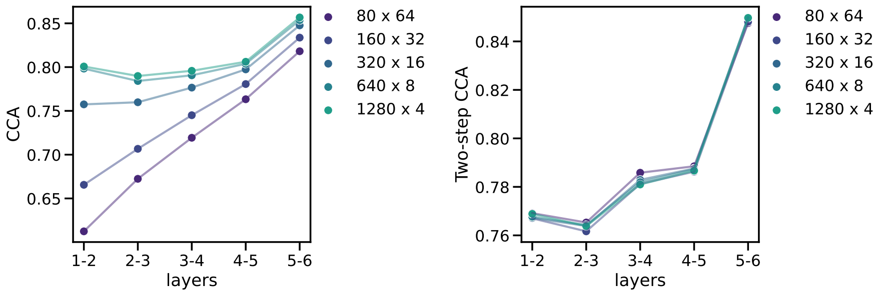
Question The following sentence contains "at your fingertips": "Using the latest in audio visual technology, the wonders of these six fascinating 'worlds' are at your fingertips." Now categorise the translation of the red word from above in this sentence: "Met behulp van de nieuwste audio visuele technologie, zijn de wonderen van deze zes fascinerende werelden binnen handbereik." ∘ paraphrase ∘ word-by-word ∘ copy
Follow-up question *If you did not select 'word for word', leave blank.* What is the translation of the red keyword in "at your fingertips" in the sentence below: (. . . insert sentence. . .) (. . . free text response box. . .)

Table D.2: Format of the questions shown to participants via the Qualtrics platform.

speaker, who was a native speaker of Norwegian and Finnish and fluent in Swedish and English. The annotators participated in a similar pre-screening test with language-specific explanations and examples, and seven practice questions. If the annotators' answers differed from what was expected, the instructions were discussed with the annotator before they proceeded with the full survey, and they filled out the remainder of the survey without intermediate help or instructions. Table D.2 shows an example question for Dutch. We previously discussed the results of this survey in §6.2.

D.2.3 Ethical considerations

The two aforementioned surveys were both approved through the university's research ethics process (applications 2019/83180 and 2021/44709, respectively), where an independent committee assessed the setup of the survey, the research's potentially harmful impacts and the compensation for the participants. In collecting data annotations, participants were shown data from the MAGPIE corpus, available under the CC-BY-4.0 License. All other information shown to them was either collected from the computational model or written up by the authors. Any identifiable information about the participants was stored separately from the participants' annotations, for the purpose of compensation. Participants provided informed consent to data collection and anonymised data being used in academic publications. They were given the opportunity to withdraw at any time. Participants were compensated above the minimum hourly wage of the country in which they were residents when participating in the study.



(a) CCA per subset (b) Two-step CCA

Figure D.1: Illustration of the impact of recomputing CCA with data subsets of differently composed vocabularies for a dataset size of 5k.

D.3 Two-step CCA

CCA can be used to compare representations over different layers of the same network or different networks in a way that is invariant to affine transformations (Raghu et al., 2017). The CCA similarity expresses the extent to which two representations contain the same information while accounting for transformations in the two views of the data. Nonetheless, the similarity depends on the data used to perform CCA. Even with a dataset that is at least an order of magnitude larger than the dimensions in the hidden representations, the composition of the dataset affects the outcome. Particularly relevant in the context of our work is the vocabulary size.

We illustrate this by measuring how hidden representations change over layers, randomly sampling tokens and considering multiple dataset compositions, varying from 64 occurrences of 80 unique tokens, to 4 occurrences of 1280 unique tokens. Recomputing CCA per subset yields the similarities shown in Figure D.1a. Although the overall pattern of lower similarity between lower layers and higher similarity between higher layers is present for all subsets, the absolute similarity measures differ between subsets. In Figure D.1b, however, where the projection matrix is computed on a separate dataset, subsets show comparable similarities. The differences between the methods decrease as the number of hidden representations used to perform CCA grows.

Performing CCA separately per (relatively small) subset of the MAGPIE corpus could thus reflect vocabulary differences rather than systematic differences due to figurativeness. We merely want to apply CCA to account for differences between layers and differences with and without masking attention, and thus apply two-step CCA, computing projection matrices on a separate dataset.

Bibliography

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ekin Akyürek, Afra Feyza Akyürek, and Jacob Andreas. 2021. Learning to recombine and resample data for compositional generalization. In *The Ninth International Conference on Learning Representations*.

Ekin Akyürek, Tolga Bolukbasi, Frederick Liu, Binbin Xiong, Ian Tenney, Jacob Andreas, and Kelvin Guu. 2022. Towards tracing knowledge in language models back to the training data. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2429–2446.

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes. In *The Fifth International Conference on Learning Representations*.

Chantal Amrhein, Nikita Moghe, and Liane Guillou. 2022. Aces: Translation accuracy challenge sets for evaluating machine translation metrics. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 479–513.

Dimitra Anastasiou, Chikara Hashimoto, Preslav Nakov, and Su Nam Kim, editors. 2009. *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications (MWE 2009)*.

Jacob Andreas. 2020. Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566.

Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. 2019. Intrinsic dimension of data representations in deep neural networks. *Advances in Neural Information Processing Systems*, 32:6111–6122.

Chidanand Apte, Fred Damerau, and Sholom M Weiss. 1994. Towards language independent automated learning of text categorization models. In *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 23–30.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR.

Eleftherios Avramidis, Vivien Macketanz, Ursula Strohriegel, and Hans Uszkoreit. 2019. Linguistic evaluation of German-English machine translation using a test suite. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 445–454.

Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *ArXiv*, abs/1607.06450.

Giosuè Baggio. 2021. Compositionality in a parallel architecture for language processing. *Cognitive Science*, 45(5):e12949.

Giosuè Baggio, M Van Lambalgen, and Peter Hagoort. 2012. The processing consequences of compositionality. In *The Oxford handbook of compositionality*, pages 655–672. Oxford University Press.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *The Third International Conference on Learning Representations*.

Mona Baker et al. 1992. *In Other Words*. Routledge.

Robert Baldock, Hartmut Maennel, and Behnam Neyshabur. 2021. Deep learning through the lens of example difficulty. *Advances in Neural Information Processing Systems*, 34:10876–10889.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of* *the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72.

Anabela Barreiro, Johanna Monti, Brigitte Orliac, Fernando Batista, et al. 2013. When multiwords go bad in machine translation. In *Proceedings of the Workshop on Multi-word Units in Machine Translation and Translation Technology*, pages 26–33.

Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. 2020. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070.

Abdul Basit, Abdul Hameed Azeemi, and Agha Ali Raza. 2024. Challenges in Urdu machine translation. In *Proceedings of the The Seventh Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2024)*, pages 44–49.

Jasmijn Bastings, Marco Baroni, Jason Weston, Kyunghyun Cho, and Douwe Kiela. 2018. Jump to better conclusions: SCAN both left and right. In *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pages 47–55.

Jasmijn Bastings and Katja Filippova. 2020. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 149–155.

Christos Baziotis, Prashant Mathur, and Eva Hasler. 2023. Automatic evaluation and analysis of idioms in neural machine translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3682–3700.

Mohammad Mahdi Bejani and Mehdi Ghatee. 2021. A systematic review on overfitting control in shallow and deep neural networks. *Artificial Intelligence Review*, 54(8):6391–6438.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. 2019. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854.

Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raff. 2024. Emergent and predictable memorization in large language models. *Advances in Neural Information Processing Systems*, 36:28072–28090.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow.

Samuel A Bobrow and Susan M Bell. 1973. On catching on to idiomatic expressions. *Memory & cognition*, 1(3):343–346.

Heather Bortfeld. 2003. Comprehending idioms cross-linguistically. *Experimental psychology*, 50(3):217.

Shaked Brody, Uri Alon, and Eran Yahav. 2023. On the expressivity role of layernorm in transformers' attention. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 14211–14221.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Nyssa Z Bulkes and Darren Tanner. 2017. "Going to town": Large-scale norming and statistical analysis of 870 American English idioms. *Behavior research methods*, 49(2):772–783.

Stéphanie Caillies and Kirsten Butcher. 2007. Processing of idiomatic expressions: Evidence for a new hybrid view. *Metaphor and Symbol*, 22(1):79–108.

Kate Cain, Andrea S Towse, and Rachael S Knight. 2009. The development of idiom comprehension: An investigation of semantic and contextual processing skills. *Journal of experimental child psychology*, 102(3):280–298.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021.

Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Rudolf Carnap. 1947. *Meaning and necessity: A study in semantics and modal logic*, volume 30. University of Chicago Press.

Rahma Chaabouni, Roberto Dessì, and Eugene Kharitonov. 2021. Can transformers jump around right in natural language? Assessing performance transfer from scan. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 136–148.

Kent Chang, Mackenzie Cramer, Sandeep Soni, and David Bamman. 2023. Speak, memory: An archaeology of books known to ChatGPT/GPT-4. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7312–7327.

Ting-Yun Chang, Jesse Thomason, and Robin Jia. 2024. Do localization methods actually localize memorized data in LLMs? A tale of two benchmarks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3190–3211.

Satrajit Chatterjee. 2018. Learning and memorization. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 755–763. PMLR.

Angelica Chen, Ravid Shwartz-Ziv, Kyunghyun Cho, Matthew L Leavitt, and Naomi Saphra. 2023. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in MLMs. In *The Twelfth International Conference on Learning Representations*.

Yuheng Chen, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2024. Journey to the center of the knowledge neurons: Discoveries of language-independent knowledge neurons and degenerate knowledge neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 17817–17825.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4324–4333.

Zheng Chia. 2024. *Exploring Optimal Settings for Machine Translation of Irony with Application to Multilingual Irony Detection*. Phd thesis, Kitami Institute of Technology.

Zheng Lin Chia, Michal Ptaszynski, Marzena Karpinska, Juuso Eronen, and Fumito Masui. 2024. Initial exploration into sarcasm and irony through machine translation. *Natural Language Processing Journal*, 9:100106.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019a. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019b. What does BERT look at? An analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

Gilad Cohen, Guillermo Sapiro, and Raja Giryes. 2018. DNN or k-NN: That is the generalize vs. memorize question. *arXiv preprint arXiv:1805.06822*.

Henry Conklin, Bailin Wang, Kenny Smith, and Ivan Titov. 2021. Meta-learning to compositionally generalize. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3322–3335.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136.

Mathieu Constant, Gülşen Eryiğit, Johanna Monti, Lonneke Van Der Plas, Carlos Ramisch, Michael Rosner, and Amalia Todirascu. 2017. Multiword expression processing: A survey. *Computational Linguistics*, 43(4):837–892.

Gonçalo M. Correia, Vlad Niculae, and André F. T. Martins. 2019. Adaptively sparse transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2174–2184.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Róbert Csordás, Kazuki Irie, and Juergen Schmidhuber. 2021. The devil is in the detail: Simple tricks improve systematic generalization of transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 619–634.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502.

Verna Dankers, Elia Bruni, and Dieuwke Hupkes. 2022a. The paradox of the compositionality of natural language: A neural machine translation case study. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4154–4175.

Verna Dankers*, Anna Langedijk*, Kate McCurdy, Adina Williams, and Dieuwke Hupkes. 2021. Generalising to German plural noun classes, from the perspective of a recurrent neural network. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 94–108. *Equal contribution.

Verna Dankers and Christopher Lucas. 2023. Non-compositionality in sentiment: New data and analyses. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5150–5162.

Verna Dankers, Christopher Lucas, and Ivan Titov. 2022b. Can transformer be too compositional? Analysing idiom processing in neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3608–3626.

Verna Dankers and Vikas Raunak. 2025. Memorization inheritance in sequence-level knowledge distillation for neural machine translation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 760–774.

Verna Dankers and Ivan Titov. 2022. Recursive neural networks with bottlenecks diagnose (non-)compositionality. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4361–4378.

Verna Dankers and Ivan Titov. 2024. Generalisation first, memorisation second? Memorisation localisation for natural language classification tasks. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 14348–14366.

Verna Dankers, Ivan Titov, and Dieuwke Hupkes. 2023. Memorisation cartography: Mapping out the memorisation-generalisation continuum in neural machine translation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8323–8343.

Xander Davies, Lauro Langosco, and David Krueger. 2023. Unifying grokking and double descent. In *NeurIPS ML Safety Workshop*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506.

Ona de Gibert, Naiara Pérez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255.

Louis des Tombe, Doug Arnold, Lieven Jaspaert, Rod Johnson, Steven Krauwer, Mike Rosner, Nino Varile, and Susan Warwick. 1985. A preliminary linguistic framework for EUROTRA, June 1985. In *Proc. Conf. on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, pages 283–288.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Desakh Putu Setyalika Putri Dewayanti and Margana Margana. 2024. The impact of contextual understanding on neural machine translation accuracy: A case study of Indonesian cultural idioms in English translation. *Englisia: Journal of Language, Education, and Humanities*, 12(1):223–236.

Tom Dietterich. 1995. Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, 27(3):326–327.

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, et al. 2024. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128.

Sundesh Donthi, Maximilian Spencer, Om B. Patel, Joon Young Doh, Eid Rodan, Kevin Zhu, and Sean O'Brien. 2025. Improving LLM abilities in idiomatic translation. In *Proceedings of the First Workshop on Language Models for Low-Resource Languages*, pages 175–181.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, G Heigold, S Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. In *The Eighth International Conference on Learning Representations*.

David Dowty. 2007. Compositionality as an empirical problem. *Direct compositionality*, 14:23–101.

Daniel D'souza, Zach Nussbaum, Chirag Agarwal, and Sara Hooker. 2021. A tale of two long tails. In *ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning*.

Emmanuel Dupoux. 2018. Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner. *Cognition*, 173:43–59.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175.

Mai ElSherief, Caleb Ziems, David Muchlinski, Vaishnavi Anupindi, Jordyn Seybolt, Munmun De Choudhury, and Diyi Yang. 2021. Latent hatred: A benchmark for understanding implicit hate speech. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 345–363.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2018. Examining the tip of the iceberg: A data set for idiom translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Marzieh Fadaee and Christof Monz. 2020. The unreasonable volatility of neural machine translation models. In *Proceedings of the Fourth Workshop on Neural Generation and Translation, NGT@ACL 2020, Online, July 5-10, 2020*, pages 88–96.

Vitaly Feldman. 2020. Does learning require memorization? A short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959.

Vitaly Feldman and Chiyuan Zhang. 2020. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:2881–2891.

Javier Ferrando and Marta R Costa-jussà. 2021. Attention weights in transformer NMT fail aligning words between sequences but largely explain model predictions. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 434–443.

Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R Costa-jussà. 2024. A primer on the inner workings of transformer-based language models. *arXiv preprint arXiv:2405.00208*.

Javier Ferrando, Matthias Sperber, Hendra Setiawan, Dominic Telaar, and Saša Hasan. 2023. Automating behavioral testing in machine translation. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1014–1030.

Javier Ferrando and Elena Voita. 2024. Information flow routes: Automatically interpreting language models at scale. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17432–17445.

Charles J Fillmore, Paul Kay, and Mary Catherine O'Connor. 1988. Regularity and idiomaticity in grammatical constructions: The case of let alone. *Language*, 64(3):501–538.

Catherine Finegan-Dollak, Jonathan K Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, Rui Zhang, and Dragomir Radev. 2018. Improving text-to-SQL evaluation methodology. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 351–360.

James Fodor, Simon De Deyne, and Shinsuke Suzuki. 2025. Compositionality and sentence meaning: Comparing semantic parsing and transformers on a challenging sentence similarity dataset. *Computational Linguistics*, 51(1):139–190.

Jerry A Fodor. 1987. *Psychosemantics: The problem of meaning in the philosophy of mind*, volume 2. MIT press.

Jerry A Fodor and Zenon W Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.

Gottlob Frege et al. 1892. Über sinn und bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, 100(1):25–50.

Daniel Furrer, Marc van Zee, Nathan Scales, and Nathanael Schärli. 2020. Compositional generalization in semantic parsing: Pre-training vs. specialized architectures. *arXiv preprint arXiv:2007.08970*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Marcos García, Tiago Kramer Vieira, Carolina Scarton, Marco Idiart, and Aline Villavicencio. 2021a. Assessing the representations of idiomaticity in vector models with a noun compound dataset labeled at type and token levels. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2730–2741.

Marcos García, Tiago Kramer Vieira, Carolina Scarton, Marco Idiart, and Aline Villavicencio. 2021b. Probing for idiomaticity in vector space models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3551–3564.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495.

Raymond W Gibbs Jr, Raymond W Gibbs, and Jr Gibbs. 1994. *The poetics of mind: Figurative thought, language, and understanding*. Cambridge University Press.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela

Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

H. Paul Grice. 1975. Logic and conversation. *Syntax and Semantics*, 3:41–58.

H. Paul Grice. 1989. *Studies in the Way of Words*. Harvard University Press.

Nuno M Guerreiro, Elena Voita, and André FT Martins. 2023. Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1059–1075.

Hessel Haagsma, Johan Bos, and Malvina Nissim. 2020. Magpie: A large corpus of potentially idiomatic expressions. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 279–287.

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2023. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *Advances in Neural Information Processing Systems*, 36:76033–76060.

Valentin Hartmann, Anshuman Suri, Vincent Bindschaedler, David Evans, Shruti Tople, and Robert West. 2023. SoK: Memorization in general-purpose large language models. *arXiv preprint arXiv:2310.18362*.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. 2023. Does localization inform editing? Surprising differences in causality-based localization vs. knowledge editing in language models. *Advances in Neural Information Processing Systems*, 36:17643–17668.

Reyhaneh Hashempour and Aline Villavicencio. 2020. Leveraging contextual embeddings and idiom principle for detecting idiomaticity in potentially idiomatic expressions. In *Proceedings of the Workshop on the Cognitive Aspects of the Lexicon*, pages 72–80.

Adi Haviv, Ido Cohen, Jacob Gidron, Roei Schuster, Yoav Goldberg, and Mor Geva. 2023. Understanding transformer memorization recall through idioms. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 248–264.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Lucas Torroba Hennigen, Adina Williams, and Ryan Cotterell. 2020. Intrinsic probing through dimension selection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 197–216.

Jonathan Herzig, Peter Shaw, Ming-Wei Chang, Kelvin Guu, Panupong Pasupat, and Yuan Zhang. 2021. Unlocking compositional generalization in pre-trained models using intermediate representations. *arXiv preprint arXiv:2104.07478*.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Edward Holsinger. 2013. Representing idioms: Syntactic and contextual effects on idiom processing. *Language and speech*, 56(3):373–394.

Paul Horwich. 2001. Deflating compositionality. *Ratio*, 14(4):369–385.

Harold Hotelling. 1936. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the second BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 1–21.

Dandan Huang, Kun Wang, and Yue Zhang. 2021. A comparison between pre-training and large-scale back-translation for neural machine translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1718–1732.

Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. 2022. Are large pre-trained language models leaking your personal information? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2038–2047.

Jing Huang, Diyi Yang, and Christopher Potts. 2024. Demystifying verbatim memorization in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10711–10732.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intellgence Research*, 67:757–795.

Dieuwke Hupkes, Mario Giulianelli, Verna Dankers, Mikel Artetxe, Yanai Elazar, Tiago Pimentel, Christos Christodoulopoulos, Karim Lasri, Naomi Saphra, Arabella Sinclair, et al. 2023. A taxonomy and review of generalization research in NLP. *Nature Machine Intelligence*, 5(10):1161–1174.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926.

Daphne Ippolito, Florian Tramer, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher Choquette Choo, and Nicholas Carlini. 2023. Preventing generation of verbatim memorization in language models gives a false sense of privacy. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 28–53.

Pierre Isabelle, Colin Cherry, and George Foster. 2017. A challenge set approach to evaluating machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2486–2496.

Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556.

Theo MV Janssen. 1998. Algebraic translations, correctness and algebraic compiler construction. *Theoretical Computer Science*, 199(1-2):25–56.

Theo MV Janssen and Barbara H Partee. 1997. Compositionality. In *Handbook of logic and language*, pages 417–473. Elsevier.

Yichen Jiang and Mohit Bansal. 2021. Inducing transformer's compositional generalization ability via auxiliary sequence prediction tasks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6253–6265.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121.

Jerrold J Katz and Jerry A Fodor. 1963. The structure of a semantic theory. *Language*, 39(2):170–210.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, et al. 2019. Measuring compositional generalization: A comprehensive method on realistic data. In *The Seventh International Conference on Learning Representations*.

Eugene Kharitonov, Marco Baroni, and Dieuwke Hupkes. 2021. How BPE affects memorization in transformers. *arXiv preprint arXiv:2110.02782*.

Najoung Kim and Tal Linzen. 2020. COGS: a compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105.

Seungone Kim, Juyoung Suk, Xiang Yue, Vijay Viswanathan, Seongyun Lee, Yizhong Wang, Kiril Gashteovski, Carolin Lawrence, Sean Welleck, and Graham Neubig. 2024. Evaluating language models as synthetic data generators. *arXiv preprint arXiv:2412.03679*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, et al. 2023. Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet. In *WMT23-Eighth Conference on Machine Translation*, pages 198–216.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, et al. 2024. Findings of the WMT24 general machine translation shared task: the LLM era is here but MT is not solved yet. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1–46.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, et al. 2022. Findings of the 2022 conference on machine translation (WMT22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1–45.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR.

Kris Korrel, Dieuwke Hupkes, Verna Dankers, and Elia Bruni. 2019. Transcoding compositionally: Using attention to find more generalizable solutions. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 1–11.

Alex Krizhevsky. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75.

Ryoma Kumon, Daiki Matsuoka, and Hitomi Yanaka. 2024. Evaluating structural generalization in neural machine translation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13220–13239.

Murathan Kurfalı and Robert Östling. 2020. Disambiguation of potentially idiomatic expressions with contextual embeddings. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons*, pages 85–94.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, pages 2873–2882. PMLR.

Brenden M Lake. 2019. Compositional generalization through meta sequence-to-sequence learning. In *Advances in Neural Information Processing Systems*, pages 9791–9801.

Brenden M. Lake, Tal Linzen, and Marco Baroni. 2019. Human few-shot learning of compositional instructions. In *Proceedings of the 41th Annual Meeting of the*

Cognitive Science Society, CogSci 2019: Creativity + Cognition + Computation, Montreal, Canada, July 24-27, 2019, pages 611–617. cognitivesciencesociety.org.

Yair Lakretz, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. 2019. The emergence of number and syntax units in LSTM language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 11–20.

Benjamin LeBrun, Alessandro Sordoni, and Timothy J O'Donnell. 2022. Evaluating distributional distortion in neural language modeling. In *The Tenth International Conference on Learning Representations*.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2018. Hallucinations in neural machine translation. In *NeurIPS 2018 Workshop on Interpretability and Robustness for Audio, Speech, and Language*.

Minjae Lee, Youngbin Noh, and Seung Jin Lee. 2025. A testset for context-aware LLM translation in Korean-to-English discourse level translation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1632–1646.

Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale N Fung, Mohammad Shoeybi, and Bryan Catanzaro. 2022. Factuality enhanced language models for open-ended text generation. *Advances in Neural Information Processing Systems*, 35:34586–34599.

Pietro Lesci, Clara Meister, Thomas Hofmann, Andreas Vlachos, and Tiago Pimentel. 2024. Causal estimation of memorisation profiles. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15616–15635.

Chuanhao Li, Zhen Li, Chenchen Jing, Yuwei Wu, Mingliang Zhai, and Yunde Jia. 2024a. Compositional substitutivity of visual reasoning for visual question answering. In *European Conference on Computer Vision*, pages 143–160. Springer.

Shuang Li, Jiangjie Chen, Siyu Yuan, Xinyi Wu, Hao Yang, Shimin Tao, and Yanghua Xiao. 2024b. Translate meanings, not just words: Idiomkb's role in optimizing idiomatic translation with language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18554–18563.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Yafu Li, Yongjing Yin, Yulong Chen, and Yue Zhang. 2021a. On compositional generalization of neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4767–4780.

Yinheng Li, Rogerio Bonatti, Sara Abdali, Justin Wagle, and Kazuhito Koishida. 2024c. Data generation using large language models for text classification: An empirical case study. In *Workshop on Data-centric Machine Learning Research (DMLR): Datasets for Foundation Models @ ICML*.

Yuanpeng Li, Liang Zhao, Jianyu Wang, and Joel Hestness. 2019. Compositional generalization for primitive substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4293–4302.

Zhu Li, Zhi-Hua Zhou, and Arthur Gretton. 2021b. Towards an understanding of benign overfitting in neural networks.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic data generation with large language models for text classification: Potential and limitations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10443–10461.

Weiduo Liao, Ying Wei, Mingchen Jiang, Qingfu Zhang, and Hisao Ishibuchi. 2023. Does continual learning meet compositionality? New benchmarks and an evaluation framework. *Advances in Neural Information Processing Systems*, 36:33499–33513.

Zheng Wei Lim, Ekaterina Vylomova, Charles Kemp, and Trevor Cohn. 2024. Predicting human translation difficulty with neural machine translation. *Transactions of the Association for Computational Linguistics*, 12:1479–1496.

Adam Liška, Germán Kruszewski, and Marco Baroni. 2018. Memorize or generalize? Searching for a compositional RNN in a haystack. In *The Joint Workshop on Architectures and Evaluation for Generality, Autonomy and Progress in AI (AEGAP) at ICML*.

Patrick Littell, David R Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. Uriel and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14.

Emmy Liu, Aditi Chaudhary, and Graham Neubig. 2023a. Crossing the threshold: Idiomatic machine translation through retrieval augmentation and loss weighting. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15095–15111.

Emmy Liu and Graham Neubig. 2022. Are representations built from the ground up? An empirical examination of local composition in language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9053–9073.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023b. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM computing surveys*, 55(9):1–35.

Qian Liu, Shengnan An, Jian-Guang Lou, Bei Chen, Zeqi Lin, Yan Gao, Bin Zhou, Nanning Zheng, and Dongmei Zhang. 2020. Compositional generalization by learning analytical expressions. *Advances in Neural Information Processing Systems*, 33:11416–11427.

Yutong Liu. 2022. Compositional generalization in machine translation for low-resource languages. Master's thesis, University of Edinburgh.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *The Fifth International Conference on Learning Representations*.

João Loula, Marco Baroni, and Brenden Lake. 2018. Rearranging the familiar: Testing compositional generalization in recurrent networks. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 108–114.

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2022. Post-hoc interpretability for neural NLP: A survey. *ACM Computing Surveys*, 55(8):1–42.

Pratyush Maini, Michael C. Mozer, Hanie Sedghi, Zachary Chase Lipton, J. Zico Kolter, and Chiyuan Zhang. 2023. Can neural network memorization be localized? In *International Conference on Machine Learning*, pages 23536–23557. PMLR.

Gary F Marcus. 2003. *The algebraic mind: Integrating connectionism and cognitive science*. MIT press.

Andrea E Martin and Giosuè Baggio. 2020. Modelling meaning composition from formalism to mechanism. *Philosophical Transactions of the Royal Society B*, 375(1791):20190298.

R Thomas McCoy, Paul Smolensky, Tal Linzen, Jianfeng Gao, and Asli Celikyilmaz. 2023. How much do language models copy from their training data? Evaluating linguistic novelty in text generation using raven. *Transactions of the Association for Computational Linguistics*, 11:652–670.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Tarun Ram Menta, Susmit Agrawal, and Chirag Agarwal. 2025. Analyzing memorization in large language models through the lens of model attribution. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10661–10689.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. What happens to BERT embeddings during fine-tuning? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 33–44.

Fatemehsadat Mireshghallah, Archit Uniyal, Tianhao Wang, David K Evans, and Taylor Berg-Kirkpatrick. 2022. An empirical analysis of memorization in fine-tuned autoregressive language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1816–1826.

Begoña Villada Moirón, Aline Villavicencio, Diana McCarthy, Stefan Evert, and Suzanne Stevenson, editors. 2006. *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*.

Anssi Moisio, Mathias Creutz, and Mikko Kurimo. 2023. On using distribution-based compositionality assessment to evaluate compositional generalisation in machine translation. In *Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP*, pages 204–213.

Ari Morcos, Maithra Raghu, and Samy Bengio. 2018. Insights on representational similarity in neural networks with canonical correlation. *Advances in neural information processing systems*, 31.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning BERT: Misconceptions, explanations, and strong baselines. In *The Ninth International Conference on Learning Representations*.

Marius Mosbach, Anna Khokhlova, Michael A Hedderich, and Dietrich Klakow. 2020. On the interplay between fine-tuning and sentence-level probing for linguistic knowledge in pre-trained transformers. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 68–82.

Max Müller-Eberstein, Rob Van Der Goot, Barbara Plank, and Ivan Titov. 2023. Subspace chronicles: How linguistic information emerges, shifts and interacts during language model training. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13190–13208.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. 2023. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*.

Vasudevan Nedumpozhimana and John Kelleher. 2021. Finding BERT's idiomatic key. In *Proceedings of the 17th Workshop on Multiword Expressions (MWE 2021)*, pages 57–62.

Ryan M Nefdt. 2020. A puzzle concerning compositionality in machines. *Minds & Machines*, 30(1).

Vlad Niculae and Christof Monz. 2023. Joint dropout: Improving generalizability in low-resource neural machine translation through phrase pair variables. *MT Summit 2023*, page 12.

Jingcheng Niu, Andrew Liu, Zining Zhu, and Gerald Penn. 2024. What does the knowledge neuron thesis have to do with knowledge? In *The Twelfth International Conference on Learning Representations*.

nostalgebraist. 2020. Interpreting GPT: The logit lens. `https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens`. Aug 2020.

Geoffrey Nunberg, Ivan A Sag, and Thomas Wasow. 1994. Idioms. *Language*, 70(3):491–538.

Mohammed M Obeidat, Ahmad S Haider, Sausan Abu Tair, and Yousef Sahari. 2024. Analyzing the performance of Gemini, ChatGPT, and Google Translate in rendering English idioms into Arabic. *FWU Journal of Social Sciences*, 18(4).

Santiago Ontanon, Joshua Ainslie, Zachary Fisher, and Vaclav Cvicek. 2022. Making transformers solve compositional tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3591–3607.

Inbar Oren, Jonathan Herzig, and Jonathan Berant. 2021. Finding needles in a haystack: Sampling structurally-diverse training sets from synthetic data for compositional generalization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10793–10809.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov Chain Monte Carlo. *Prague Bulletin of Mathematical Linguistics*, 106:125–146.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 1–9.

Peter Pagin and Dag Westerståhl. 2010a. Compositionality i: Definitions and variants. *Philosophy Compass*, 5(3):250–264.

Peter Pagin and Dag Westerståhl. 2010b. Compositionality ii: Arguments and problems. *Philosophy Compass*, 5(3):265–282.

Peter Pagin and Dag Westerståhl. 2010c. Pure quotation and general compositionality. *Linguistics and Philosophy*, 33:381–415.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

Barbara Partee. 1984. Compositionality. *Varieties of formal semantics*, 3:281–311.

Prasanna Parthasarathi, Koustuv Sinha, Joelle Pineau, and Adina Williams. 2021. Sometimes we want ungrammatical translations. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3205–3227.

Andrew Pawley and Frances Hodgetts Syder. 1983. Natural selection in syntax: Notes on adaptive variation and change in vernacular and literary grammar. *Journal of pragmatics*, 7(5):551–579.

Francis Jeffry Pelletier. 1994. The principle of semantic compositionality. *Topoi*, 13(1):11–24.

Ana Pellicer-Sánchez and Frank Boers. 2018. Pedagogical approaches to the teaching and learning of formulaic language. *Understanding formulaic language*, pages 153–173.

Martina Penke. 2012. The dual-mechanism debate. In *The Oxford handbook of compositionality*. Oxford University Press.

Elisabeth Piirainen. 2012. *Widespread idioms in Europe and beyond: Toward a lexicon of common figurative units (International Folkloristics)*. Peter Lang Publishing Group.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. 2022. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*.

USVSN Sai Prashanth, Alvin Deng, Kyle O'Brien, Jyothir SV, Mohammad Aflah Khan, Jaydeep Borkar, Christopher A Choquette-Choo, Jacob Ray Fuehne, Stella Biderman, Tracy Ke, et al. 2024. Recite, reconstruct, recollect: Memorization in LMs as a multifaceted phenomenon. *arXiv preprint arXiv:2406.17746*.

H Putnam. 1975. Mind, language and reality. *Philosophical papers*, 2.

Linlu Qiu, Peter Shaw, Panupong Pasupat, Pawel Nowak, Tal Linzen, Fei Sha, and Kristina Toutanova. 2022. Improving compositional generalization with latent structure and data augmentation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4341–4362.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. Technical report, OpenAI.

Alex Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. Technical report, OpenAI.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. *EMNLP 2018*, page 287.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. 2017. SVCCA: singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6078–6087.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Vikas Raunak, Vaibhav Kumar, Florian Metze, and Jaimie Callan. 2019. On compositionality in neural machine translation. In *NeurIPS 2019 Context and Compositionality in Biological and Artificial Neural Systems Workshop*.

Vikas Raunak and Arul Menezes. 2022. Finding memo: Extractive memorization in constrained sequence generation tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5153–5162.

Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. 2021. The curious case of hallucinations in neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1172–1183.

Vikas Raunak, Arul Menezes, Matt Post, and Hany Hassan Awadalla. 2023. Do GPTs produce less literal translations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1041–1050.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7237–7256.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. 2021. Probing the probing paradigm: Does probing accuracy entail task relevance? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3363–3377.

Paul Rayson, Scott Piao, Serge Sharoff, Stefan Evert, and Begona Villada Moirón. 2010. Multiword expressions: Hard going or plain sailing? *Language Resources and Evaluation*, 44:1–5.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702.

Matīss Rikters and Ondřej Bojar. 2017. Paying attention to multi-word expressions in neural machine translation. In *Proceedings of Machine Translation Summit XVI: Research Track*, pages 86–95.

MT Rosetta. 1994. *Compositional translation*, volume 273. Springer Science & Business Media.

D E Rumelhart and J McClelland. 1986. On learning the past tenses of English verbs. In *Parallel distributed processing: Explorations in the microstructure of cognition*, pages 216–271. MIT Press, Cambridge, MA.

Jacob Russin, Jason Jo, Randall O'Reilly, and Yoshua Bengio. 2020. Compositional generalization by factorizing alignment and translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 313–327.

Ivan A Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *Computational Linguistics and Intelligent Text Processing: Third International Conference, CICLing 2002, Proceedings*, pages 1–15. Springer.

Lukas Santing, Ryan Sijstermans, Giacomo Anerdi, Pedro Jeuris, Marijn ten Thij, and Riza Batista-Navarro. 2022. Food for thought: How can we exploit contextual embeddings in the translation of idiomatic expressions? In *Proceedings of the 3rd Workshop on Figurative Language Processing (FLP)*, pages 100–110.

Diana Santos. 1990. Lexical gaps and idioms in machine translation. In *COLING 1990 Volume 2: Papers presented to the 13th International Conference on Computational Linguistics*.

Amartya Sanyal, Puneet K Dokania, Varun Kanade, and Philip Torr. 2020. How benign is benign overfitting? In *The Eighth International Conference on Learning Representations*.

Naomi Saphra and Adam Lopez. 2019. Understanding learning dynamics of language models with SVCCA. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3257–3267.

Naomi Saphra and Sarah Wiegreffe. 2024. Mechanistic? In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 480–498.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3687–3697.

Ali Satvaty, Suzan Verberne, and Fatih Turkmen. 2024. Undesirable memorization in large language models: A survey. *arXiv preprint arXiv:2410.02650*.

Beatrice Savoldi, Marco Gaido, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2022. On the dynamics of gender learning in speech translation. In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 94–111.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021a. WikiMatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021b. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6490–6500.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725.

Yutong Shao, Rico Sennrich, Bonnie Webber, and Federico Fancellu. 2018. Evaluating machine translation performance on Chinese idioms with a blacklist method. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Arnab Sen Sharma, David Atkinson, and David Bau. 2024. Locating and editing factual associations in Mamba. In *First Conference on Language Modeling*.

Shanya Sharma, Manan Dey, and Koustuv Sinha. 2022. How sensitive are translation systems to extra contexts? Mitigating gender bias in neural machine translation models through relevant contexts. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1968–1984.

Peter Shaw, Ming-Wei Chang, Panupong Pasupat, and Kristina Toutanova. 2021. Compositional generalization and natural language variation: Can a semantic parsing approach handle both? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 922–938.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *The Fifth International Conference on Learning Representations*.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. 2024. AI models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759.

Vered Shwartz and Ido Dagan. 2019. Still a pain in the neck: Evaluating text representations on lexical composition. *Transactions of the Association for Computational Linguistics*, 7:403–419.

Diana Sidtis, Gina Canterucci, and Dora Katsnelson. 2009. Effects of neurological damage on production of formulaic language. *Clinical linguistics & phonetics*, 23(4):270–284.

John J Sidtis, Diana Van Lancker Sidtis, Vijay Dhawan, and David Eidelberg. 2018. Switching language modes: complementary brain patterns for formulaic and propositional language. *Brain connectivity*, 8(3):189–196.

Paul Smolensky. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial intelligence*, 46(1-2):159–216.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Patrick Stadler, Vivien Macketanz, and Eleftherios Avramidis. 2021. Observing the learning curve of NMT systems with regard to linguistic phenomena. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 186–196.

David Stap, Eva Hasler, Bill Byrne, Christof Monz, and Ke Tran. 2024. The fine-tuning paradox: Boosting translation quality without sacrificing LLM abilities. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6189–6206.

Cory Stephenson, Suchismita Padhy, Abhinav Ganesh, Yue Hui, Hanlin Tang, and Sue Yeon Chung. 2021. On the geometry of generalization and memorization in deep neural networks. In *The Ninth International Conference on Learning Representations*.

Niklas Stoehr, Mitchell Gordon, Chiyuan Zhang, and Owen Lewis. 2024. Localizing paragraph memorization in language models. *arXiv preprint arXiv:2403.19851*.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Kaiser Sun, Adina Williams, and Dieuwke Hupkes. 2023. The validity of evaluation results: Assessing concurrence across compositionality benchmarks. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 274–293.

Maria Helena Svensson. 2008. A very complex criterion of fixedness: Non-compositionality. *Phraseology: An interdisciplinary perspective. John Benjamins Publishing Company*.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293.

Zoltan Szabó. 2012. The case for compositionality. *The Oxford handbook of compositionality*, 64:80.

Zoltán Gendler Szabó. 2004. Compositionality.

Sho Takase, Shun Kiyono, Sosuke Kobayashi, and Jun Suzuki. 2023. B2T connection: Serving stability and performance in deep transformers. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

David Talbot, Hideto Kazawa, Hiroshi Ichikawa, Jason Katz-Brown, Masakazu Seno, and Franz J Och. 2011. A lightweight evaluation framework for machine translation reordering. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 12–21.

Takaaki Tanaka, Aline Villavicencio, Francis Bond, and Anna Korhonen, editors. 2004. *Proceedings of the Workshop on Multiword Expressions: Integrating Processing*.

Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2018. An analysis of attention mechanisms: The case of word sense disambiguation in neural machine translation.

In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 26–35.

Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2019. Encoders help you disambiguate word senses in neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1429–1435.

Harish Tayyar Madabushi, Edward Gow-Smith, Carolina Scarton, and Aline Villavicencio. 2021. AStitchInLanguageModels: Dataset and methods for the exploration of idiomaticity in pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3464–3477.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.

Jörg Tiedemann. 2020. The Tatoeba translation challenge – realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480.

Kushal Tirumala, Aram Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. *Advances in Neural Information Processing Systems*, 35:38274–38290.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. 2019. An empirical study of example forgetting during deep neural network learning. In *The Seventh International Conference on Learning Representations*.

Catherine Torrington Eaton and Sarah Thomas. 2024. To make a long story short: A descriptive study of formulaic language use in post-stroke fluent aphasia. *Aphasiology*, 38(7):1180–1194.

Charles Travis. 1985. On what is strictly speaking true. *Canadian Journal of Philosophy*, 15(2):187–229.

Solomon Ubani, Suleyman Olcay Polat, and Rodney Nielsen. 2023. ZeroShotDataAug: Generating and augmenting training data with ChatGPT. *arXiv preprint arXiv:2304.14334*.

Dmitrii Usynin, Moritz Knolle, and Georgios Kaissis. 2024. Memorisation in machine learning: A survey of results. *Transactions on Machine Learning Research*.

Andreas Van Cranenburgh, Remko Scha, and Rens Bod. 2016. Data-oriented parsing with discontinuous constituents and function tags. *Journal of Language Modelling*, 4(1):57–111.

Diana van Lancker Sidtis. 2012. Two-track mind: Formulaic and novel language support a dual-process model. *The handbook of the neuropsychology of language*, 1:342–367.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, et al. 2018. Tensor2Tensor for neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 193–199.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019a. The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4387–4397.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019b. When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1198–1212.

Elena Voita, Rico Sennrich, and Ivan Titov. 2021. Language modeling, lexical translation, reordering: The training process of NMT through the lens of classical SMT. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8478–8491.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019c. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196.

Mila Vulchanova, Evelyn Milburn, Valentin Vulchanov, and Giosuè Baggio. 2019. Boon or burden? The role of compositional meaning in figurative language processing and acquisition. *Journal of Logic, Language and Information*, 28(2):359–387.

Yoav Wald, Gal Yona, Uri Shalit, and Yair Carmon. 2023. Malign overfitting: Interpolation and invariance are fundamentally at odds. In *The Eleventh International Conference on Learning Representations*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019b. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *The Seventh International Conference on Learning Representations*.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 billion parameter autoregressive language model.

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. 2020. Neural machine translation with byte-level subwords. In *Proceedings of the AAAI conference on artificial intelligence*, pages 9154–9160.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*.

Zhepeng Wang, Runxue Bao, Yawen Wu, Jackson Taylor, Cao Xiao, Feng Zheng, Weiwen Jiang, Shangqian Gao, and Yanfu Zhang. 2024. Unlocking memorization in large language models with dynamic soft prompting. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9782–9796.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Jiaheng Wei, Yanjun Zhang, Leo Yu Zhang, Ming Ding, Chao Chen, Kok-Leong Ong, Jun Zhang, and Yang Xiang. 2024. Memorization in deep learning: A survey. *arXiv preprint arXiv:2406.03880*.

Dag Westerståhl. 2002. On the compositionality of idioms. *Proceedings of LLC8. CSLI Publications*.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20.

Alison Wray. 1992. *The Focusing Hypothesis: The theory of left hemisphere lateralised language re-examined*, volume 3. John Benjamins Publishing.

Alison Wray. 2002. *Formulaic language and the lexicon.* ERIC.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. 2024. LaMini-LM: A diverse herd of distilled models from large-scale instructions. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 944–964.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. 2020. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR.

Fuzhao Xue, Yao Fu, Wangchunshu Zhou, Zangwei Zheng, and Yang You. 2023. To repeat or not to repeat: Insights from scaling LLM under token-crisis. *Advances in Neural Information Processing Systems*, 36:59304–59322.

Yongjing Yin, Jiali Zeng, Yafu Li, Fandong Meng, Jie Zhou, and Yue Zhang. 2023. Consistency regularization training for compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1294–1308.

Wlodek Zadrozny. 1994. From compositional to systematic semantics. *Linguistics and philosophy*, 17:329–342.

Andrea Zaninello and Alexandra Birch. 2020. Multiword expression aware neural machine translation. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3816–3825.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104.

Ziheng Zeng and Suma Bhat. 2023. Unified representation for non-compositional and compositional expressions. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11696–11710.

Qiusi Zhan, Richard Fang, Rohan Bindu, Akul Gupta, Tatsunori Hashimoto, and Daniel Kang. 2024. Removing RLHF protections in GPT-4 via fine-tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 681–687.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. In *The Fifth International Conference on Learning Representations*.

Chiyuan Zhang, Samy Bengio, and Yoram Singer. 2022a. Are all layers created equal? *The Journal of Machine Learning Research*, 23(1):2930–2957.

Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. 2023. Counterfactual memorization in neural language models. *Advances in Neural Information Processing Systems*, 36:39321–39362.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022b. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Xinze Zhang, Junzhe Zhang, Zhenhua Chen, and Kun He. 2021. Crafting adversarial examples for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1967–1977.

Wayne Xin Zhao, Naoki Yoshinaga, and Daisuke Oba. 2024a. Tracing the roots of facts in multilingual language models: Independent, shared, and transferred knowledge. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2088–2102.

Yiran Zhao, Jinghan Zhang, I Chern, Siyang Gao, Pengfei Liu, Junxian He, et al. 2024b. FELM: Benchmarking factuality evaluation of large language models. *Advances in Neural Information Processing Systems*, 36.

Hao Zheng and Mirella Lapata. 2023. Real-world compositional generalization with disentangled sequence-to-sequence learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1711–1725.

Xiaosen Zheng and Jing Jiang. 2022. An empirical study of memorization in NLP. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6265–6278.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

Vitor C Zimmerer, Mark Wibrow, and Rosemary A Varley. 2016. Formulaic language in people with probable alzheimer's disease: A frequency-based approach. *Journal of Alzheimer's Disease*, 53(3):1145–1160.