Temporality and Modality in Entailment Graph Induction

Sander Govert Cornelis

Bijl de Vroe

Doctor of Philosophy

Institute for Language, Cognition and Computation

School of Informatics

University of Edinburgh

2022

Abstract

The ability to draw inferences is core to semantics and the field of Natural Language Processing. Answering a seemingly simple question like ‘Did Arsenal play Manchester yesterday’ from textual evidence that says ‘Arsenal won against Manchester yesterday’ requires modeling the inference that ‘winning’ entails ‘playing’. One way of modeling this type of lexical semantics is with Entailment Graphs, collections of meaning postulates that can be learned in an unsupervised way from large text corpora.

In this work, we explore the role that temporality and linguistic modality can play in inducing Entailment Graphs. We identify inferences that were previously not supported by Entailment Graphs (such as that ‘visiting’ entails an ‘arrival’ before the visit) and inferences that were likely to be learned incorrectly (such as that ‘winning’ entails ‘losing’). Temporality is shown to be useful in alleviating these challenges, in the Entailment Graph representation as well as the learning algorithm. An exploration of linguistic modality in the training data shows, counterintuitively, that there is valuable signal in modalized predications. We develop three datasets for evaluating a system’s capability of modeling these inferences, which were previously underrepresented in entailment rule evaluations. Finally, in support of the work on modality, we release a relation extraction system that is capable of annotating linguistic modality, together with a comprehensive modality lexicon.

Acknowledgements

It’s a cliché, but I could truly never have written this thesis without the support of many amazing people.

Thank you Mark Steedman, for teaching me so much. I am grateful for your inspirational ability to focus on the most important ideas in the bigger picture, to identify real novelty and to always think of research as a long-term endeavour. Combined with your patience and being so generous with your time I think this allows your students to really develop into independent thinkers at their own pace. I will miss our wide-ranging discussions in the company of the most overflowing bookshelves I have ever seen, and I’ll fondly remember walking out of meetings feeling thoroughly galvanized.

Thank you Ido Dagan and Mirella Lapata for examining this thesis. You kindly made sure our discussion was an enjoyable one, and the thesis is much better for it. Thank you to my secondary advisors Bonnie Webber and Shay Cohen for valuable feedback on the project and ideas about new directions. It was comforting to know I had people to turn to for support. Thank you also to my Master’s thesis supervisor Wilker Aziz, for patiently giving me a first foundation in research.

Thank you to my talented co-authors! Liane Guillou, thanks for keeping each other sane during the pandemic, for all your support, and for all the lessons learned in project management, experimentation, collaboration, programming, etc., etc., etc. Thank you Thomas Kober for teaching me the ropes in the research world and for your patience. Thank you Javad Hosseini, for your kindness and all your guidance in building on your codebase. Nick McKenna, thank you for intellectual stimulation when I needed it most, for your positivity and curiosity — I will miss our explorations in philosophy of language. Mark Johnson, thank you for many useful (often crucial) words of advice over transcontinental group meetings. And thank you Miloš Stanojević, for enjoyable and motivating discussions, of which I hope there will be many more. Your enthusiasm (even for parsing algorithm complexity) is truly infectious.

Thank you Christos Christodoulopoulos for your excellent mentorship during my Amazon internship, and the rest of the Amazon Alexa research team in Cambridge. Thanks Joshua Wong for your pair programming wizardry when my hands started to protest.

Thank you Nate Chambers, Ivan Titov, Alex Lascarides, Paola Merlo, Sam Bowman, James Henderson, Michael Collins and many others in the research community, for being giving with your time and having conversations that gave a lot of perspective and direction.

Thank you to the academic family around me in Edinburgh and abroad; to Laura Aina, Daniel Angelov, Mattias Appelgren, Esma Balkır, Sameer Bansal, Ronald Cardenas, Lushi Chen, Liang Cheng, Carol Chermaz, Paula Czarnowska, Radina Dobreva, Daniel Duma, Denis Emelin, Federico Fancellu, Jess Gasson, Sorcha Gilroy, Seraphina Goldfarb-Tarrant, Maria Gorinova, Matt Grenander, Andreas Grivas, Zack Hodari, Bobby Ikonomov, Craig Innes, Elif Kaplan, Tianyi Li, Tianjang Liu, Erinn Los, Jonathan Mallinson, Yevgen Matusevych, Kate McCurdy, Andrew McLeod, Julie-Anne Meaney, Ellen Mears, Nikita Moghe, Elizabeth Nielsen, James Owers, Ratish Puduppully, Alex Robertson, Matt Rounds, Ramon Sanabría, Naomi Saphra, Christine Simpson, Stefanie Speichert, Ida Szubert, Clara Vania, Sarenne Wallbridge and Sabine Weber. You each really deserve paragraphs of your own, and your companionship made all of this even more worthwhile.

I want to thank friends outside academia who also lended essential support. Thank you to Edinburgh friends Ieva Gerliakaite, David Halliwell, Marta Idzak and Raf Taylor. To friends in the Netherlands for welcoming me back with open arms and much needed Skype chats over the pandemic; Fridi: Rens Bakker, Marleen Biezeno, Kavish Bisseswar, Joost Meulendijks, Floortje van Oosten, Franka Rang, Robert van Schaik, Mara Soekarjo, Lieke van Tooren, Diederik Verduldt and Annerijn Vink; and Licht 2012: Jonna de Vries, Hidde Hofhuis, Daniel Goodefrooij, Chris Berendse, Niels van Stijn, Evert Cofino, Koen Dreijer, Pim van Dorst, Sam Beenhakker, Thomas de Weers. You made sure I always returned to Edinburgh energized. Thank you to Lucas de Haas, Kay Lamerigts, Nicholas Otto and Max Whitby, conversations with you always inspire me.

I would like to extend my sincere thanks to the Talon community, whose open-source voice control software made it possible to continue doing research in spite of injury. Thanks also to James Beavis and the Qigong crowd in the Edinburgh meadows for helping so much in recovery.

Finally, thank you Frank, Maud, Govert-Jan and Loes Bijl de Vroe for your unflagging support. I am immensely grateful to have all of you in my life and I dedicate this thesis to you.

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.

(*Sander Govert Cornelis*
Bijl de Vroe)

Lay Summary

The field of Natural Language Processing has made an enormous amount of progress in the past decades. Many people now use *digital* personal assistants, like the Google Assistant or Amazon Alexa — something that was perhaps unthinkable 50 years ago. But anyone that has used these assistants knows they aren't perfect. We cannot ask them anything we want, and we cannot phrase our questions to them in all the ways that feel natural to us.

A central area we are working to improve is Natural Language Inference. Models of human language are currently able to extract much of the information that a sentence states *explicitly*, but they struggle to understand information contained *implicitly* — information that needs to be inferred. For example, if we read that "Arsenal won the game last night" and somebody later asks us "Did Arsenal play last night?", we can answer "Yes". As humans we can answer this question because we have somehow stored the information that if a sports team *wins*, they also *play*. In linguistics this relation between predicates is called *entailment*, and it is a type of knowledge that computational models, including Alexa and the Google Assistant, struggle to learn.

One strategy for answering these kinds of questions involves 1) building a *parser* which extracts structured information from text available on the web, 2) encoding that information in a Knowledge Graph and 3) building some sort of inference engine, that can make inferences about those facts. We can then combine these resources to answer a question: we use the parser to understand the question ("Did Arsenal play last night?"), search for facts that might answer it in the Knowledge Graph ("Arsenal won last night"), and see whether our inference engine leads us from our facts to an answer ("Yes").

This brings us to the material of this thesis. We try to learn better *Entailment Graphs* — models that can be part of an inference engine, that store the entailment relations between linguistic predicates like *win* and *play*. Entailment Graphs can be learned automatically using an algorithm. We give the algorithm a huge number of news articles (say, a million) and by looking at what entities (e.g. *Arsenal and Manchester United*) participate in what predicates (e.g. *win*), it can give us an Entailment Graph.

This thesis builds on previous research by improving Entailment Graphs using temporality and modality. We put time in the representation so that the graphs can make temporal inferences (for example, if you *are visiting* you *have arrived*, while it is not

the case that you *will arrive*.

We also add temporality to the learning algorithm. We realized that the graphs contained incorrect relations between predicates like *win* and *lose*, because these occur with the same entities (Sometimes Arsenal win; sometimes they lose). However, they always occur with those entities *at different times*. So, we used models that understand the part of language that refers to time — tense, aspect, our temporal expressions and explicit and implicit ordering of events — to automatically understand *when* all the events happened in the news articles. By taking these times into account in a new learning algorithm, we remove the false edges between *win* and *lose* and learn more accurate graphs. We also built three evaluation datasets containing both the temporal and the contradictory types of inferences, to test performance of different models.

By modality we mean the part of language that deals with hypotheticals, possibility, desire, and other descriptions of things that don't necessarily happen in the real world. Think of sentences like "Arsenal might win." or "if only Arsenal had won ... ". Our algorithm previously ignored this information, so that some of its data referred to events that never occurred, which might be confusing to the model. We build a parser that understands modality and show that in cases like the sports domain it is useful to remove modal data, while it can be useful to keep it in other cases.

So, the general research project is to improve Entailment Graphs, making them more accurate using time as a signal and putting time in the representation, and taking modality into account in the training data. Downstream, this could improve reasoning abilities of larger systems, which allows these systems to understand us humans in a more fluid way, because they become able to make the same implicit inferences that we make.

Table of Contents

Chapter 1

Introduction

Entailment is one of the central components of meaning in natural language. Montague called it "the basic aim of semantics" (Montague, 1970). The ability to recognize the entailments of a text is therefore essential to many Natural Language Processing (NLP) applications, such as Information Extraction (IE), Information Retrieval (IR), document summarization and Question Answering (QA) (Kozareva and Montoyo, 2006).

For example, when answering an open-domain question like "Did Arsenal play Manchester last night?", the unstructured textual data at our disposal might state that "Arsenal beat Manchester 1-0". This poses the challenge that the answer to the question is not explicitly stated in the text — it needs to be inferred using entailment knowledge. Only a system that can somehow recognize that *beat* $\models$ *play* will be able to provide the correct answer ("yes").

The tasks described above all run into one of the fundamental challenges of Natural Language Understanding (NLU): the many-to-many mapping between meanings and surface forms with which natural language confronts us. A single surface form can have multiple meanings (semantic ambiguity), and a single meaning can be expressed using multiple surface forms (semantic variability). Knowledge of entailment assists us in the semantic variability problem. It brings us closer to an understanding of text that is independent of the surface form — modeling a form-independent semantics.

Systems that reason with natural language often rely on entailment rules such as *beat* $\models$ *play*. These can form a kind of background world knowledge to be used by an inference system. In fact the background knowledge discovery problem is one of the main bottlenecks of improving these systems (Bos, 2014). It has therefore been a prominent research enterprise to create collections of entailment rules, which are preferably mined automatically (since they are expensive to design by hand), while still exhibiting high accuracy and high coverage.

This central problem has recently been tackled using *Entailment Graphs* (Berant, 2012), graph structures in which the nodes are linguistic predicates and the edges between them represent entailment relations. They can be learned in an unsupervised way from large collections of multi-authored news text, which avoids the expensive process of building such a lexicon manually. They also have the advantage of being both interpretable and explainable: the relations are represented explicitly so that a human can easily interpret the outputs, and it is possible to track down the evidence that supports any particular relation in the training corpus.

Their learning signal is based on the Distributional Inclusion Hypothesis (DIH). Using the argument pairs of a predicate as its context, this states that a predicate (e.g. *win against*) entails another (*play*) if the context set of the first predicate (e.g. {*(Arsenal, Manchester)*, *(Everton, Liverpool)*, *...*}) is included in that of the second (e.g. {*(Arsenal, Manchester)*, *(Manchester, Arsenal)*, *(Everton, Liverpool)*, *...*}). By example, because *win* entails *play*, whenever a team *wins* against another team, they must also *play* that team (while not every team that plays against another team wins against them). When we find this inclusion pattern occurring in the data between two predicates, it indicates that an entailment relation likely exists between them. Algorithms based on this intuition can be used to automatically induce Entailment Graphs.

Entailment Graphs and their induction process still face certain challenges, however. For one, until now they have not been expanded to model temporally contingent inferences. Atemporal Entailment Graphs might learn a general entailment relation between *visit* and *arrive*, but this does not fully reflect the relation between the predicates, since it ignores the fact that the arrival must happen before the visit. The challenge is that our models should only support inferences like “Obama is visiting Hawaii”⊨“Obama has arrived in Hawaii” and ⊨ “Obama will leave Hawaii”, while excluding temporally incorrect inferences like “Obama is visiting Hawaii”⊨“Obama will arrive in Hawaii” or ⊨ “Obama has left Hawaii”.

Another unexplored challenge is that the atemporal DIH breaks down for antonymous predicates that occur frequently with the same argument pairs. For example, predicates like *win* and *lose* will occur frequently with the same sports teams (say, *Arsenal* and *Manchester* — sometimes Arsenal win; sometimes they lose). The algorithm interprets this overlap as an indication of an entailment relation between *win* and *lose*, whereas the true relation is one of contradiction.

This thesis explores two possible signals with which to engage in these challenges.

On the one hand, we investigate temporality. We can access the tense and aspect of predications using techniques in parsing and part-of-speech tagging. We can also make use of recent NLP progress to understand *when* a described eventuality happens, by recognizing eventualities in the text, parsing temporal referring expressions (such as *on Monday two weeks from now*) and predicting the temporal orderings between them. Tense and aspect can form potential features of the predicate representation in the graph (modeling *was visiting* instead of simply *visit*, for instance). The eventuality time, on the other hand, has potential as a signal in the Entailment Graph induction algorithm, since although *win* and *lose* might occur with the same two arguments, they will never co-occur *simultaneously*.

We also investigate modality as a potential signal. Our languages have rich systems for describing eventualities that never actually happen — uncertain future events, hypotheticals, counterfactuals, desires, commands, etc. NLP has developed tools to understand this other kind of factive displacement, allowing us to tag predications with their modal category. It seems tenable that the Entailment Graph learning signal would benefit from this information, using the cleaner signal of only those predications actually asserted as happening. Otherwise, "Arsenal might win against Manchester." and the description "Arsenal lost to Manchester" (referring to the same sports event) would reinforce the spurious entailment between *win* and *lose*.

1.1 Thesis Statement

Temporal entailment is an unexplored and challenging problem for Entailment Graphs. It can be encoded in a Lexical Inference in Context task setup by varying the tense and aspect of the premise and hypothesis predications. The inferences can be modeled in Tensed Entailment Graphs by including tense and aspect in the predicate nodes.

Antonyms that are correlated through their argument pairs present another challenge to Entailment Graphs. Temporality can be injected as a useful signal into an Entailment Graph mining algorithm, learning the correct entailments while avoiding spurious entailments to correlated antonyms. This temporal formulation of the DIH is effective for predicates with particular properties, such as those occurring in the sports domain. Linguistic modality can be similarly useful in this regard — sports domain Entailment Graphs learned from data of exclusively asserted predications are of higher quality. This effect does not hold in the general domain, showing that modalized predications can be as valuable as asserted predications to entailment learning in certain circumstances.

1.2 Thesis Outline and Contributions

In Chapter 2, we discuss relevant literature from NLP and linguistics. We first summarize work on entailment and Entailment Graphs, before discussing temporality and modality.

In Chapter 3 we define the problem of temporal entailment, previously largely ignored in Natural Language Inference. We present a novel entailment dataset, TEA[1], containing these inferences, described in Section 3.2. The work was carried out with Thomas Kober, collaborating closely on the problem definition, dataset design and annotation. Thomas experimented with initial baselines on the dataset independently, and initial experimental results will therefore be presented only briefly). In Section 3.3 we present alterations to the Entailment Graph induction pipeline to create Tensed Entailment Graphs. We show that these graphs are able to learn novel temporal inferences, although due to sparsity they are limited to low recall. Again, the research was carried out in collaboration with Thomas Kober. While we collaborated closely, most of the implementation and experimentation work (altering the parser; running and evaluating models) was carried out by me. Most of the work in Section 3.2 was published as (Kober et al., 2019), while the work in Section 3.3 has remained unpublished.

In Chapter 4 we present a novel Entailment Graph induction algorithm that incorporates a temporal signal. To evaluate this idea we create the *Sports Entailment Dataset*[2] (Guillou et al., 2020), created using a new semi-automatic construction method based on clusters of paraphrases. The dataset is inspired by the theoretical point that the atemporal DIH is likely to conflate antonymy and entailment for some predicates. We show that temporality is a useful signal for learning entailment. Liane Guillou and I collaborated closely on engineering, experimentation and data construction in this project. The chapter is based on work published under (Guillou et al., 2020) (shared first authorship).

In Chapter 5 we generalize the experimentation in Chapter 4 to the news domain more broadly. We test the idea of applying a dynamic temporal comparison window, set separately for each eventuality in the data using a neural duration estimation model

[1] Available at https://github.com/tttthomasssss/iwcs2019.
[2] Available at https://gitlab.com/lianeg/temporal-entailment-sports-dataset.

(Zhou et al., 2020). We contribute the ANT[3] dataset, containing more general domain antonyms and entailments produced semi-automatically using WordNet (Miller, 1993). This demonstrates that the method works better on the sports domain specifically. We further analyse the conditions under which the method works, and show that there are other subdomains for which the temporal method might also be successful. Again, this work was carried out in close collaboration with Liane Guillou. The work is in the reviewing process at the time of submission of this thesis.

Chapter 6 presents work on a modality tagger based on a Combinatory Categorial Grammar (CCG) (Steedman, 2000) dependency parser. The tagger includes the most extensive modality lexicon to date, compiled from various other resources. In preparation of experimentation in Chapter 7 we verify that the tagger attains sufficiently high accuracy for downstream application. The work in this chapter was also performed with Liane Guillou; it was published as (Bijl de Vroe et al., 2021) under shared first authorship.

In Chapter 7 we use linguistic modality as a signal in Entailment Graph induction. Our experimentation shows that the signal is useful in the sports domain; a graph built using only asserted predications outperforms the standard graph. As with temporality, the effect does not carry over to the general domain. In that case, using a mix of asserted and modalized predications is more effective than an identical amount of asserted predication data, showing that modalized predications can still provide a useful signal for entailment in some circumstances. I again shared first authorship with Liane Guillou and the work is published as (Guillou et al., 2021).

At various points we refer to Multivalent Entailment Graphs (McKenna et al., 2021). Although I collaborated on that project (primarily in recognizing the problem and designing the initial solution), the project was defined further and carried out by Nick McKenna and is not presented in full in this thesis.

Finally, we conclude in Chapter 8. We present lessons learned in entailment evaluation dataset construction, and discuss the most promising directions revealed by the thesis work, both in including temporality in the semantic representation and in experimenting with temporality as a learning signal.

[3] To be made publicly available.

Chapter 2

Background

2.1 Introduction

In this Chapter, we discuss the relevant background for the remainder of the thesis. We first discuss the linguistic background of entailment, and its investigation under the lens of NLP (Section 2.2). We also take the opportunity here to discuss the important role of entailment in the lexicon and semantics more broadly. We separately discuss the Entailment Graph literature in Section 2.3.

We then move to the phenomena of temporality (2.4) and modality (2.5). In both cases we first touch on the relevant linguistic background, before moving to the approaches that NLP has taken in modeling and dataset construction.

2.2 Entailment

2.2.1 Definition

The formal definition of entailment can be given either proof-theoretically or model-theoretically (Shapiro, 2005). Proof-theoretically, a set of formulae *P* entails a formula *h* within a deductive system if there exists a sequence of formulae that ends in *h*, where every formula either 1) is an axiom of the system, 2) belongs to *P*, or 3) follows from previous formulae using the system's rules of inference. In model theory (Tarski, 1937), conversely, *P* entails *h* if every interpretation of the model *M* that satisfies all formulae in *P* also satisfies *h* (in other words, if *h* is true in all the possible worlds in which *P* holds). The latter approach provided an initial basis for formal semantics (Montague, 1970), and has since occupied an important role in the study of meaning, for instance for defining a semantics of linguistic modality (Kratzer, 1981).

At its core entailment is about drawing inferences, such as if *a team beat another team* then *the teams played*. Entailment datasets in computational semantics (more about which in Section 2.2.3) have increasingly described this using a more common-sense or probabilistic usage of the term entailment. For example, the Recognizing Textual Entailment task defines an entailment relation between two pieces of text (premise P and hypothesis H) as follows: "P entails H if, typically, a human reading P would infer that H is most likely true" (Dagan et al., 2006). Note that this softens the standard definition of logical entailment, in which an entailed statement is unquestionably true given a hypothesis, to a more probabilistic and human-centered interpretation. Although some argue that the use of *entailment* in this case diverges too much from the technical definition (Manning, 2006), it has certain practical advantages, such as more straightforwardly including the wide range of inferences that humans draw (including discourse phenomena such as implicatures), and allowing more actionable data annotation protocols.

For the remainder of this thesis, our usage of the term entailment follows the probabilistic interpretation. We describe potential entailment relations as holding between a *premise* and a *hypothesis* (*P-H*), as opposed to a *text* and *hypothesis* (*T-H*) preferred in some other literature. When an entailment relation holds (doesn't hold), we use the logical notation $P \vDash H$ ($P \nvDash H$). Since our entailments normally apply to binary predicates, we sometimes borrow the *p, q, r* notation for predicates, defaulting to predicate *p* as the premise and *q* as the hypothesis. Entailment relations can hold between many linguistic objects — between discourse level structures such as paragraphs, or at the sentence-level between utterances. At the lexical level they can hold between entities (e.g. $dog \vDash animal$), predicates ($run \vDash move$), or other categories of concepts. This thesis will pertain to entailment relations between predicates contextualized with argument types (e.g. $beat(sports\_team, sports\_team) \vDash play(sports\_team, sports\_team)$).

2.2.2 Entailment and (Lexical) Semantics

Entailment takes a central role in capturing meaning in language (Katz, 1972; Van Benthem, 2008). This centrality has also been recognized in NLP; in presenting FRaCas, one of the earliest entailment datasets, Cooper et al. (1996) state that "inferential ability is not only a central manifestation of semantic competence but is in fact centrally constitutive of it". In this view a lexical unit's meaning is in large part the sum of all its entailments to other lexical units. Under the entailment view of semantics an Entailment Graph can also be viewed as a lexicon, every node and its entailment edges constituting a kind of Carnapian meaning postulate (Carnap, 1952) (see also Section 2.3).

Many computational lexical semantic resources contain some form of entailment in support of this view. The WordNet lexicon (Miller, 1993) establishes various semantic relations between its word nodes, including entailment[1]. It is possible to use FrameNet (Baker et al., 1998) to mine entailment relations between predicates (Ben Aharon et al., 2010), although its primary purpose is to support inferences from predicates to the existence of the various typical arguments in a frame (with a lexical unit "evoking" a set of frame elements). The hierarchically organized lexicon VerbNet (Schuler, 2005) allows inferences between different levels in the hierarchy.

Constructing these lexical resources manually is a labor-intensive, iterative procedure. Various projects have therefore attempted to automatically gather inference-based lexical knowledge, such as the 100 million paraphrases of the PPDB (Ganitkevitch et al., 2013). In the expanded PPDB 2.0 (Pavlick et al., 2015), a notion of fine-grained entailment was one of the primary additions.

2.2.3 Datasets

2.2.3.1 Early Datasets

One of the earliest NLP datasets exploring entailment is the FraCaS test suite (Cooper et al., 1996). Each example is handcrafted, resulting in a small (340 examples), high-quality test set that is linguistically motivated. It covers a broad range of inference types, including examples that require knowledge of specific logical and linguistic concepts such as monotonicity, quantification, negation, anaphora and subordinate clauses. The dataset offers multiple premises that can be used to reason to a hypothesis, in contrast to the later style using single premises and hypotheses. It contains a subsection that focuses on temporality, although this is also fairly small. Example (1) shows one such case from FraCaS, requiring an understanding of tense and aspect, temporal operators like *since* and reasoning about time itself.

(1) *Premise*: Since 1992 ITEL has been in Birmingham.
Premise: It is now 1996.

[1] Incidentally, entailment can be seen as a generalization of many of the semantic relations in these lexical databases: with antonymy, $p \vDash \neg q \wedge q \vDash \neg p$ and with synonymy and paraphrase, $p \vDash q \wedge q \vDash p$. With hypernyms, $p \nvDash q \wedge q \vDash p$, while with hyponoms $p \vDash q \wedge q \nvDash p$.

Question: Was ITEL in Birmingham in 1993?
Answer: Yes

Monz and de Rijke (2001) take a less manual approach. They generate a set of premise-hypothesis pairs by combining subsections of different documents that describe the same topic. A human annotator then judges whether one of the pieces of text contains all the information expressed in the other. From 69 documents this results in a set of approximately 12,000 pairs, although the class distribution is heavily skewed toward non-entailments.

The task was further developed in the PASCAL Recognizing Textual Entailment shared task series, introduced in RTE-1 (Dagan et al., 2006). Instead of focusing on detailed logic-inspired entailments like the ones contained in FraCaS, they concentrate on entailment variation stemming from lexical relations and syntax. A binary entailment/non-entailment labeling scheme is used, and the examples always consist of a single premise and a hypothesis. The series ran until its 7th edition (Bentivogli et al., 2011), attracting attention to semantic inference and its usefulness in downstream tasks such as QA and IE (Information Extraction).

RTE-6 (Bentivogli et al., 2010) diverges from the original P-H setup, instead casting the problem within a summarization scenario in which systems return all sentences that entail a particular hypothesis from a set of candidates. The task setup showcases another practical application of entailment knowledge: in automatic summarization of multiple documents it can be used to reduce repetitiveness, by identifying and removing sentences that contain the same information. This also reframes the challenge as textual entailment in context, expecting systems to make entailment decisions with access to a paragraph or document, although that development was largely reverted in later entailment task definitions.

2.2.3.2 Lexical Inference in Context

Inference rules (such as *win* $\models$ *play*) formed a central part of early approaches to QA, IE and RTE (see also Section 2.2.4). However, evaluating inference rules indirectly in a downstream task such as RTE proves difficult because that task normally requires many types of inferences. Evaluating the rules directly by asking human annotators whether each rule is correct (e.g. as by Shinyama et al. (2002) and Szpektor et al. (2004)) also has significant drawbacks, because considering them outside a context makes them difficult to judge, resulting in low Inter Annotator Agreement (IAA). This led to the development of *instance-based evaluation*, which alleviates the problem by providing annotators with a sample of simple premise sentences containing the predicates, and judging a rule as correct if a sufficiently high percentage of the premises entail the hypothesis predicate (Szpektor et al., 2007). The added arguments in this sentence-based approach provide context that allows annotators to disambiguate more effectively, leading to high IAA.

This task setup inspired a series of datasets for evaluating entailment rule collections. The paradigm has retrospectively been called Lexical Inference in Context (LIiC) (Schmitt and Schütze, 2019). Each example in a dataset consists of a pair of simple sentences in which only the predicate is altered, labeled as **entailed** or **not entailed**. Using entire sentences also encourages models to contain more context-sensitive representations. For example, the Levy dataset (Levy and Dagan, 2016) contains the premise-hypothesis pairs in examples (2) and (3). Both expect the **entailed** label in spite of containing different senses of *kill*.

(2) *Premise*: "The salve kills cancers"
Hypothesis: "cancers may be treated by the salve"
Label: entailment

(3) *Premise*: "Crockett was killed at the Alamo"
Hypothesis: "Crockett died at the Alamo"
Label: entailment

Other datasets in this format include Berant's dataset (Berant et al., 2011) consisting of manually annotated edges sampled from 10 Entailment Graphs. In this case the context is inherent in the predicate nodes, which are annotated with argument types (e.g. *kill*(PERSON-PERSON) rather than the more general *kill*). Zeichner et al. (2012) point out the cost-related drawbacks of the lengthy guidelines and annotator training required in the methodology of Szpektor et al. (2007). They propose a cheap, high-IAA crowd-sourcing methodology that splits the task into a meaningfulness task and an entailment task.

A crucial inherent bias is present in these two datasets, however. They pre-select candidate entailments for manual annotation according to a similarity measure or system prediction. This means that existing entailments that are not captured by these similarity measures would be excluded from evaluation. This issue is addressed in the Levy dataset, used in multiple subsequent evaluations (e.g. by Hosseini et al. (2018, 2019))

Levy and Dagan (2016) (see examples (2) and (3)) framed the collection of entailments and non-entailments as a Question Answering task in which they presented human annotators with questions and possible answers and asked them to mark the answer as True or False, indicating whether the predicate in the answer entails the predicate in the question. Entities in the answer are replaced with tokens representing their type, e.g. *London* becomes *city*, so as to reduce the bias towards world knowledge.

Despite addressing the issues of bias, the labeling error rate for entailments that hold only in one direction is high in the Levy and Dagan (2016) dataset. Holt (2018) designed a manual annotation task to address this problem, adding the reverse entailment b $\models$ a for each entailment a $\models$ b and asking the annotators to directly annotate both directions as entailment / non-entailment. The reannotated Levy/Holt dataset is commonly used in evaluation (Hosseini et al., 2018; Hosseini, 2021) and is also used in some chapters in this thesis.

Another recent evaluation dataset in the LIiC paradigm is SherLIic (Schmitt and Schütze, 2019). They collect a large number (960K) of candidate inferences using a variant of the Sherlock procedure (Schoenmackers et al., 2010), and annotate a sample of this set through crowd-sourcing. The dataset addresses some of the artefacts discovered in the recently released large Natural Language Inference (NLI) datasets (discussed next in Section 2.2.3.3). However, it still suffers from the same bias as Berant and Zeichner's datasets, because their potential rules are collected from an entailment rule learning methodology.

2.2.3.3 Modern Benchmarks

Recently a series of large-scale entailment tasks have been introduced that have served as benchmarks for modern NLU models. They represent a growth in size of three orders of magnitude compared to early datasets such as FraCaS. The first of these is the Stanford Natural Language Inference (SNLI) corpus (Bowman et al., 2015), consisting of 570K premise-hypothesis sentence pairs that were collected as part of a crowd-sourced task grounded in image captioning. Human annotators were presented with a premise (in the form of an image caption), and were asked to spontaneously generate hypotheses that were definitely true (entailment), definitely false (contradiction), or neither ("neutral"). The image caption annotation setting leads to relatively concrete sentences — consider examples (4)-(6).

This task setup is referred to as Natural Language Inference, a rebranding of the RTE paradigm that distinguishes itself in practice mainly in the inclusion of the **contradiction** label (see example (6)). SNLI's size addressed the need for a sufficiently large corpus for training the neural network architectures available at the time.

(4) *Premise*: "A soccer game with multiple males playing."
Hypothesis: "Some men are playing a sport."
Label: entailment

(5) *Premise*: "An older and younger man smiling."
Hypothesis: "Two men are smiling and laughing at the cats playing on the floor."
Label: neutral

(6) *Premise*: "A black race car starts up in front of a crowd of people."
Hypothesis: "A man is driving down a lonely road."
Label: contradiction

After SNLI, a number of other large-scale datasets were introduced. The Multi-Genre Natural Language Inference Corpus (MNLI) (Williams et al., 2018) extends to ten domains containing both spoken language and written text, thus providing broader coverage than SNLI. Stepping away from the relatively concrete descriptions of visual scenes in SNLI leads to a substantially more challenging task that involves more complicated phenomena like linguistic modality, temporal reasoning and propositional attitude. For example, annotators are unlikely to generate a relatively abstract MNLI hypothesis like "Hundreds of students will benefit from your generosity" from an image description.

XNLI (Conneau et al., 2018) extends parts of the MNLI datasets to 15 languages, this time widening the scope to cross-genre and cross-lingual examples. They also include low resource languages such as Urdu, facilitating research in transfer learning for cross-lingual language understanding (XLU).

The Diverse NLI Collection (DNC) (Poliak et al., 2018) contains diverse inference types, achieved by casting datasets designed for other tasks into the NLI structure. For example, Named Entity Recognition annotations over the sentence in example (7) can be used to generate a premise-hypothesis pair, such as in example (8). The original sentence becomes a premise, and the NER label can be used to generate a hypothesis sentence. Many tasks and existing datasets can be repurposed for NLI using similar tricks to turn label annotations into hypotheses; for instance, a similar recasting strategy was used to create the Question Answering NLI (QNLI) corpus (Wang et al., 2018).

(7) *NER Sentence*: "Netanyahu met Weizman last Tuesday and voiced his opposition, Yedioth said."
Label: Netanyahu: <PERSON>

(8) *Premise*: "Netanyahu met Weizman last Tuesday and voiced his opposition, Yedioth said."
Hypothesis: "Netanyahu is a person."
Label: entailment

These advances in size and diversity have been essential to model development, but questions have also been raised regarding whether models trained on these datasets are actually able to generalize beyond their training domain. Gururangan et al. (2018) show that (large, neural) supervised models abuse spurious statistical correlations present in the datasets — they are capable of predicting the correct labels even when only the hypothesis is available to the model. Furthermore, they fail to capture simple lexical inferences (Glockner et al., 2018); when SNLI's premises have a single word replaced by a hypernym, synonym, co-hyponym or antonym, the systems trained on SNLI fail to generalize their knowledge. Recently, a large adversarial benchmark addressing these issues was released (Nie et al., 2020). Adversarial NLI (ANLI) uses a human-in-the-loop data creation process, letting non-experts find weaknesses in models. They repeat their process for three rounds, using intermediate versions of the dataset to inform subsequent iterations, and show that this provides some protection against the spurious statistical patterns present in the previous benchmarks.

NLI has become one of the central tasks in NLU and NLP more generally. Four of the nine tasks in the popular General Language Understanding Evaluation (GLUE) benchmark dataset were NLI-based (Wang et al., 2018), including datasets such as RTE and MNLI, among others. The updated SuperGLUE benchmark maintains this focus on inference (Wang et al., 2019).

In spite of the NLP community's attention to NLI and the recent advances in diversity of annotated inferences, there are still many types of inferences that are not covered by the existing datasets. In particular, there were previously no datasets focused on lexical entailment between predicates in which time plays a role (such as

Dataset	Contents	Size	Author, Year
FraCaS	Handcrafted, logical examples	340	(Cooper et al., 1996)
Monz	Pairs generated from news text	12K	(Monz and de Rijke, 2001)
RTE-1	First RTE Shared task	1.4K	(Dagan et al., 2006)
Szpektor	Instance-based evaluation	760	(Szpektor et al., 2007)
Berant	LIiC from Entailment Graphs	39K	(Berant et al., 2011)
Zeichner	Cheaper crowd-sourcing	6.6K	(Zeichner et al., 2012)
SNLI	Large-scale crowd-sourcing	570K	(Bowman et al., 2015)
Levy	LIiC from QA pairs	16.4K	(Levy and Dagan, 2016)
Levy/Holt	Levy reannotated	18.4K	(Holt, 2018)
MNLI	Multi-genre coverage	433K	(Williams et al., 2018)
XNLI	Cross-lingual coverage	113K	(Conneau et al., 2018)
QNLI	Cast from QA	108K	(Wang et al., 2018)
DNC	Cast from other diverse tasks	570K	(Poliak et al., 2018)
SherLIic	LIiC from Sherlock procedure	4K	(Schmitt and Schütze, 2019)
ANLI-R1	Adversarial annotation	19K	(Nie et al., 2020)
-R2	”	47K	”
-R3	”	103K	”
TEA	Temporal Entailment in LIiC	11K	(Kober et al., 2019)
Sports	Antonyms in LIiC (Sports)	1.3K	(Guillou et al., 2020)
ANT	Antonyms in LIiC (General)	6.3K	(Bijl de Vroe et al., 2022)

Table 2.1: An overview of Entailment Datasets

is currently visiting entails that *has arrived*, but not *will arrive*). There were also no datasets in the LIiC landscape in which antonyms (such as *win* and *lose*) were explored alongside entailments (such as *win* and *play*), where temporality and modality might be a useful learning signal. Evaluations therefore did not reflect mistakes that models were previously making in these domains. This thesis introduces three new datasets to address this: TEA (Kober et al., 2019) focuses on temporal entailments such as *visit-arrive* (Chapter 3), the Sports Entailments Dataset (Guillou et al., 2020) explores temporally separable antonyms in the sports news domain (Chapter 4) and ANT applies the data creation strategy in a more general-domain setting (Chapter 5). Each dataset is designed within the Lexical Inference in Context paradigm.

2.2.3.4 Antonym Detection

It is worth briefly mentioning the field of antonym detection, in which antonyms are distinguished from other semantic relations such as synonymy. Some of the work in this thesis (Chapters 4 and 5 in particular) focuses on the related but distinct task of Lexical Inference in Context in the presence of antonymy, which can be seen as a more challenging version of the typical LIiC setup. Antonymy detection is evaluated using various datasets, notably the relation classification-style EVALution dataset (Santus et al., 2015), PPDB-based dataset of Rajana et al. (2017), and the multiple-choice GRE question dataset (Mohammad et al., 2013). The datasets presented in Chapters 4 and 5 could also be used to evaluate antonymy detection models.

2.2.4 Systems

We mainly focus on Entailment Graphs as a model of entailment in this thesis, so they will be discussed in more detail in Section 2.3. Here we provide an overview of other methods that have been explored.

2.2.4.1 Logical Approaches

The logically designed FraCaS test suite and RTE datasets initially prompted logical approaches. The basic strategy employed is generally to parse every premise into a formal semantic representation, collect formally represented background knowledge, and verify whether the hypothesis can be proven using an appropriate logical formalism. For example, Bos and Markert (2005) use a CCG-based parser (Bos et al., 2004) combined with a representation based on Discourse Representation Theory (Kamp and Reyle, 1993) to obtain logical representations of the sentences. To perform inference they use the Automated Theorem Prover (ATP) Vampire (Riazanov and Voronkov, 2002), which can perform proof search for either the positive or negated version of the hypothesis. In parallel, they use the model builder Paradox (Claessen and Sörensson, 2003); if it succeeds in constructing a model for the negation of the hypothesis, the system can return **False** (and halt the ATP's proof search). Thus they combine a proof-theoretic and a model-theoretic approach to establishing the deducibility of the hypothesis.

Approaches in natural logic (MacCartney and Manning, 2007) are related to this direction. They make sequences of logic-informed edits to attempt to transform the premise into the hypothesis step-by-step. The logic operates entirely on natural language strings and focuses on the monotonicity (and polarity) of the linguistic expression. For example, truth is preserved in the transformation from *They lacked weapons* to *They lacked guns*, because *lack* is downward-monotone in its second argument. Most linguistic expressions will be upward-monotone, however: *They have guns* $\sqsubseteq$ *They have weapons*. A classifier can be used to determine whether the proposed sequence of edits corresponds to an entailment relation overall.

A number of related models are based on edit distance (Kouylekov and Magnini, 2005; Bar-Haim et al., 2007; Iftene and Balahur-Dobrescu, 2007). Although they are not always overtly logical, they are similar to the natural logic approach in attempting to gradually transform the premise into the hypothesis. These approaches are more syntactic in nature, using entailment rules over dependency parse trees of the premise and hypothesis. An entailment relation is more likely if the edit distance (expressed through a variety of cost functions) is small.

Logical approaches to entailment are still being explored. For instance, Chatzikyriakidis and Luo (2014) use a modern type theory (Luo, 2012) and the Coq automated prover (Coq, 2004) to tackle the FraCas Test Suite. Logical approaches have lately also been applied to the temporal subset of FraCaS (Bernardy and Chatzikyriakidis, 2021), by implementing a Montagovian semantics adapted for temporal purposes, including additions to the semantics such as a treatment of tense, aspect, temporal adverbials and temporal reference.

Although these systems are able to achieve relatively high precision, their recall is often low. Bos (2014) identifies background knowledge as the crucial bottleneck in achieving higher recall. Whereas both the inference systems and the parsers that translate sentences to their logical forms show relatively strong performance, it is still unclear how to mine and incorporate the background knowledge necessary to cover all the cases that evaluation datasets might contain. Entailment Graphs form one avenue of research towards alleviating this problem — the lexical knowledge stored in their entailment edges is an essential part of the background knowledge required.

2.2.4.2 Statistical Approaches

Datasets like RTE also inspired a number of methods that attempt to relate the premise and hypothesis through more probabilistic, distributional or statistical means. Note that a number of distributional approaches have also been used for the related task of learning collections of inference rules. These are more relevant to Entailment Graph learning, so we describe them in Section 2.3.3.2.

A number of approaches depend on measures of shallow semantic overlap between the premise and hypothesis. For example, Jijkoun and de Rijke (2005) treat each sentence as a bag-of-words, and compute a sentence-level directional similarity score as a weighted sum over word-level similarity scores. In spite of ignoring syntax and deeper semantic relations between the sentences, these approaches were comparatively successful.

Another approach has been to model the entailment relation probabilistically (Glickman et al., 2005a). The core idea is to assume a generative model (similar to contemporaneous methods in Statistical Machine Translation), and define that an entailment relation holds if and only if witnessing the premise increases the likelihood of the hypothesis being true (that is, $p(H = True|P) > p(H = True)$). The parameters of the generative model can be estimated using co-occurrence statistics in a large collection of news text or web text (Glickman et al., 2005b). These can in turn be used to model the sentence probabilities of the premise and hypothesis (decomposed into lexical item probabilities) in a text classification task setup. Harmeling (2007) propose another application of probabilistic models, imposing them on a dependency tree calculus similar to Bar-Haim et al. (2007). In this case, the probability that P entails H is decomposed into the probabilities of dependency tree transformations from P to H, where each probability corresponds to the chance that each successive transformation is truth-preserving.

Statistical approaches have also been applied to the discourse level. Hickl (2008) adopt a distributional approach in which the lexical alignment between the discourse commitments of the premise and hypothesis is taken into account. This allows for deeper semantic modeling, and should be especially beneficial when the context is large (i.e. the premise is a paragraph or an entire document).

2.2.4.3 Deep Learning Approaches

As in other areas of NLP (and Artificial Intelligence in general), neural network architectures of increasing complexity have been applied to the problem of modeling textual entailment. This was made more feasible by the release of evaluation benchmarks like SNLI, which are sufficiently large for data-hungry neural models. In a neural baseline model over SNLI, Bowman et al. (2015) construct sentence embeddings in a distributional space for the premise and hypothesis using a Long Short-Term Memory Network LSTM (Hochreiter and Schmidhuber, 1997). The embeddings are fed into a multi-layer perceptron classifier to obtain their entailment prediction, training the architecture end-to-end using gradient descent. Variations of neural architectures have been applied to the problem, such as incorporating an attention model over the premise (Rocktäschel et al., 2015), soft-aligning subphrases of the sentences using attention (Parikh et al., 2016), or using a match-LSTM to align the premise and hypothesis word by word (Wang and Jiang, 2016).

More recently, the state-of-the-art on large benchmarks has shifted to transformer architecture language models (Vaswani et al., 2017) that use contextualized word embeddings (Peters et al., 2018), such as BERT (Devlin et al., 2019), GPT (Radford and Narasimhan, 2018) and XLNet (Yang et al., 2019). They are now being trained with billions of tokens of data, and have grown to using billions or even trillions (Fedus et al., 2021) of parameters. Once these massive models are pre-trained using language modeling they can be applied to other tasks such as Natural Language Inference, either by fine-tuning the entire network to the task (Howard and Ruder, 2018; Radford and Narasimhan, 2018), or by using a transfer learning approach in which another network receives the pre-trained embeddings as input. The successive models increment the base transformer architecture in various ways. For example, BERT introduces a bidirectional encoder through a Masked Language Model pre-training objective — words are masked in the middle of the sentence and the model predicts them back from surrounding context on both sides. Its Next Sequence Prediction objective is also useful to NLI specifically, as it encourages the model to reason about pairs of sentences.

The large language models do have a number of drawbacks. As described in Section 2.2.3.3 large neural models suffer from overfitting, exploiting spurious training data artefacts when fine-tuned. Relatedly, some have pointed out their ability to memorize the data, which raises the question of whether they are able to learn generalizable linguistic representations (or something akin to semantics at all) (McCoy et al., 2021). This holds true especially as the number of parameters grows, as has been the recent trend (Carlini et al., 2022). Additionally, Bender and Koller (2020) recently raised the concern that any model cannot learn meaning from form alone, suggesting the need for some type of grounding in the training signal.

Thus, although state-of-the-art performance on entailment tasks is achieved by neural network architectures, built around large transformer models and the associated pretraining and transfer learning methods, there are indicators that this is not ultimately the right direction. Significant biases in our evaluation datasets, training datasets and methods mean that we may eventually need different solutions, which may hearken

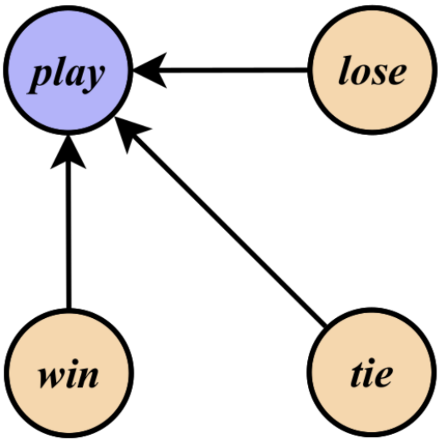


Figure 2.1: An example Entailment Graph for the predicates *win*, *play*, *lose* and *tie*, with edges representing entailment relations

back to logical approaches.

2.3 Entailment Graphs

Entailment Graphs have been proposed as a method for tackling the semantic challenges presented in Section 2.2. They are directed graphs in which the nodes represent predicates and the edges between them represent an entailment relation, and can be interpreted as a lexicon containing meaning postulates (Carnap, 1952). They have proven their usefulness in many downstream tasks that require NLU (see Section 2.3.2), and can be induced in an unsupervised way using the distributional behaviour of predicates and argument pairs in a large text corpus (Section 2.3.3). An example subgraph of an Entailment Graph is shown in Figure 2.1. Here the edge from the node for the predicate *lose* to the node for *play* means that if *lose* is seen in a particular sentential context (e.g. a description of a match between two sports teams), then *playing* most likely also occurs.

2.3.1 Representation

Previous work has considered a number of options for representing nodes in the graphs: typed binary predicates (Berant et al., 2011; Hosseini et al., 2018), Open-IE propositions (Levy et al., 2014), longer textual fragments (Kotlerman et al., 2015; Eichler et al., 2016) and eventualities (Yu et al., 2020).

In our work we use typed predicates, following Berant (2012) and Hosseini et al. (2018). Typed predicates are useful because entailment between predicates is ambiguous with respect to the arguments the predicates take. For example, $kills(:medicine, :disease) \vDash cures(:medicine, :disease)$, whereas $kills(:person, :person) \nvDash cures(:person, :person)$. Typed Entailment Graphs therefore alleviate some of the Word Sense Disambiguation (WSD) issues present in untyped graphs. A particular typed graph might contain predicates with arguments of type *person* and *location*. An edge in that graph from the relation *visit* to *be_in*, would imply that if the predicate *visit* occurs with arguments of the types *person* and *location*, then it is very likely true that the *person is_in* the *location*.

More formally we are interested in an Entailment Graph $\mathcal{G}$, the elements of which are typed directed subgraphs $G_{t_1,t_2} = (P(t_1,t_2), E)$, where the set of nodes $P(t_1,t_2)$ is the set of all predicates that use the types t_1, t_2: $P'(t_1,t_2) \cup P'(t_2,t_1)$. Here $t_n \in T$, the set of all types. The edges $E \subseteq P(t_1,t_2) \times P(t_1,t_2)$ are the entailment relations between the predicate nodes, where we use entailment to refer to the commonsense formulation described in Section 2.2.1. An edge $(p,q) \in E$ is an ordered pair that expresses the entailment relation $p \vDash q$, where $p \in P(t_1,t_2)$, $q \in P(t_1,t_2)$. We sometimes use p as shorthand for a typed predicate $p(:t_1, :t_2)$.

Note that a particular typed subgraph can contain predicates with a type pair in either order, (t_1,t_2) or (t_2,t_1). This allows us to model entailment relations between predicates with the arguments flipped. For example, in the *organization*, *person* graph this would allow us to model $work\_for(:person, :organization) \vDash pay(:organization, :person)$.

2.3.2 Use Cases

Inference rules and Entailment Graphs have shown their usefulness in many applications. For example, in Question Answering (McKenna et al., 2021) they can be used when the predicate in the question and evidence text have a different surface form. For example, with the question “Is Obama in Hawaii?” and evidence “Obama is visiting Hawaii”, the question becomes answerable using the entailment edge $is\_visiting \vDash is\_in$. Applications are also found in relation extraction (Eichler et al., 2017) and link prediction or Knowledge Graph completion (Hosseini et al., 2019) — if a system in any of these tasks encounters a triple, it can use Entailment Graphs to generate more triples by generating all entailed relations. Entailment Graphs have also been used in email categorization (Eichler et al., 2014).

One ambitious use case for Entailment Graphs in the long term is to form a crucial part of a logical inference engine. As described in Section 2.2.4.1, reasoning logically from text to hypotheses (useful to many applications that require NLU) can be performed using a combination of proof-theoretic and model-theoretic approaches. In that view, the edges of an Entailment Graph become part of the collection of rules used by the proof search system, either as the rules of deduction or as axioms over which to reason.

This paradigm is not without its challenges. The Entailment Graphs would need to be relatively complete and very accurate, since logical approaches leave little room for noise and error. Although entailment relations between predicates will be essential (since predicates carry much of the meaning in sentences), it will also be necessary to develop strong logical rule collections for other inference types. For example, the system would need to reason about arguments, time, pragmatics and modifiers such as adjectives, just to name a few categories, and for each of these should meet the same stringent accuracy demands. Again, Bos (2014) identifies background knowledge as a major bottleneck. Collecting this knowledge will be challenging, but the solution remains theoretically possible, and is a worthwhile pursuit while methods in automated theorem proving and (directed) proof search continue to improve.

On the other hand, Entailment Graphs can be useful in conjunction with neural and distributional models. One way of implementing this is by leveraging an Entailment Graph to strengthen a distributional link prediction model, which is designed to predict the score of an edge in a Knowledge Graph. Hosseini et al. (2019) show that ConvE (Dettmers et al., 2018), a convolutional neural network over Knowledge Graph embeddings, improves at link prediction when it has access to entailment scores. In their proposed model, the score of a particular triple S_{q,e_1,e_2} is high when the Entailment Graph predicts there are other high-scoring triples that would entail that triple, i.e. there are high triple scores S_{r,e_1,e_2} where $r \in R_{p \to q}$, the set of entailment relations that entail predicate q, derived from an Entailment Graph.

Another potential use of Entailment Graphs is as enrichment of a large language model. Models such as COMET (Bosselut et al., 2019) and K-BERT (Liu et al., 2020) have already demonstrated that the knowledge represented in Knowledge Graphs can be injected into language models, improving their downstream performance on a range of NLP tasks. Since Entailment Graphs can be viewed as a sort of Knowledge Graph, it may be similarly beneficial to inject their entailment knowledge into a language model. Ideally, future research would confirm that the resulting language model exhibits improved performance on NLI tasks specifically.

Incidentally, the reverse has already been demonstrated. McKenna and Steedman (2022) propose a smoothing method, which uses RoBERTa (Liu et al., 2019b) to combat Entailment Graph predicate sparsity issues. They show a significant improvement in recall when missing predicates are replaced with their nearest neighbours in RoBERTa's embedding space. For example, when the Entailment Graph does not contain information for a rare typed predicate like *obliterate(sports_team, sports_team)*, the language model can suggest paraphrases of the predicate which can be looked up in the Entailment Graph instead.

Entailment Graphs have already been applied to represent lexical knowledge in a range of different domains, including newswire (Hosseini et al., 2018), the health domain (Levy et al., 2014), and commonsense knowledge (Yu et al., 2020). Usually they are evaluated using LIiC datasets such as Levy/Holt (Levy and Dagan, 2016) or SherLIic (Schmitt and Schütze, 2019), although question answering tasks have recently also been employed (McKenna et al., 2021).

2.3.3 Learning Entailment Graphs

There are many ways to learn the graph representation described in Section 2.3.1 — ultimately any interconnected collection of rules can be viewed as an Entailment Graph. Recent approaches (Berant, 2012; Hosseini et al., 2018), however, have typically progressed in three steps. First, a corpus is analyzed with a relation extraction pipeline, producing a set of triples (e.g. $\{(win\_against(organization, organization), Arsenal, Manchester), ...\}$) that can be thought of as a Knowledge Graph. Next, in a step referred to as *local learning*, the triples are used to learn a set of independent entailment rules between typed predicates (e.g. $win\_against(organization, organization) \vDash play\_against(organization, organization)$). The output of this step can already be called an Entailment Graph. Finally, a *globalization* step uses properties of entailment (such as transitivity), combined with structural graph properties, to optimize the existing graph. We describe these steps in more detail here. Since much of the work in this thesis is inspired by the work of Hosseini et al. (2018), we will focus on their method in particular.

2.3.3.1 Relation Extraction and Open Information Extraction

We will first describe the relation extraction method used by Hosseini et al. (2018), before briefly touching on related approaches to relation extraction. The relation extraction system presented here describes the one used in the experimentation in Chapter 3. The Entailment Graph experimentation in Chapters 4, 5 and 7 uses a reimplementation based on a more modern CCG parser.

In the first step, we extract a set of typed triples, each of which consists of a predicate and two arguments, from a large collection of text. A typed predicate is denoted $p(:t_1,:t_2)$; as before $p \in P(t_1,t_2)$, the set of all predicates, and $t_n \in T$, the set of all types. We then define a typed triple as an instantiated typed predicate $p(a_1{:}t_1, a_2{:}t_2)$, where $a_n \in A$, the set of all arguments. An example of an instantiated triple is $beat_{1,2}(Arsenal{:}organization_1, Manchester{:}organization_2)$, which might be extracted from the sentence “Arsenal beat Manchester in Highbury last night”. We sometimes refer to the mention of a triple in text as a *predication*; that is, the predicate $beat_{1,2}(:organization_1,:organization_2)$ can be predicated of the arguments $Arsenal$ and $Manchester$ in a particular sentence.

Following segmentation and tokenization (Manning et al., 2014), the relation extraction system uses GraphParser (Reddy et al., 2014), based on CCG (Steedman, 2000). CCG is a constituency-based syntactic formalism in which each lexical entry is associated with a syntactic category, which can be combined together using a limited number of combinators (such as function application or composition). It is useful for relation extraction due to its transparent syntax-semantics interface, and is efficiently parsable. GraphParser uses CCG to arrive at a graph-based, neo-Davidsonian semantic representation. That is, it follows Davidson (1967) in recognizing that events can be existentially quantified over and referred to with variables, and flattens the semantics into a conjunction of thematic roles (Parsons, 1990).

GraphParser produces a logical form for our sentence shown in example (9). Note that the predicate is flattened into three different views of the same event variable. This is also the source of the subscript notation above; we let predicate subscripts correspond loosely to semantic roles. For example, $p_{1,2}$ refers to a predicate in which the first argument is the subject and the second the object.

(9) **Sentence**: “Arsenal beat Manchester in Highbury.”

Logical form: $\exists e[beat_1(e,Arsenal) \wedge beat_2(e,Manchester)$
$\wedge\ beat_{in}(e,Highbury)]$

Extractions:
$beat_{1,2}(Arsenal, Manchester)$
$beat_{1,in}(Arsenal, Highbury)$
$beat_{2,in}(Manchester, Highbury)$

Hosseini et al. (2018) then arrive at the triple extractions shown in the example, by applying a series of post-processing steps over the information present in the logical form. Principally, they construct a triple by considering all pairs of entities shared by the same predicate, arriving at three extractions for one event in the example above. A number of steps are then applied. For example, passives are recognized using the POS-tagger, and are converted to their active representations (e.g. "Manchester was beaten by Arsenal" would also result in $beat_{1,2}(Arsenal, Manchester)$). The relations are annotated with negation, prepositions and event modifiers where necessary. Compound predicates that contain arguments are constructed in certain cases via prepositional attachment. For example, the predicate *play_game_with* can be extracted from the sentence "Arsenal plays a game with Manchester". The triples are also lemmatized.

On the argument side, we first apply the Named Entity Recognition system included in CoreNLP Manning et al. (2014) to classify substrings in the sentence as either a named entity (N) or general entity (G, all other nouns and noun phrases). We sometimes refer to pairs of arguments as EE, EG, GE, or GG, depending on the types of entities involved[2]. Triples that contain pairs of general entities (GG) are discarded.

Next, we perform Named Entity Linking with AIDA-Light (Nguyen et al., 2014), mapping the arguments to their Freebase (Bollacker et al., 2008) IDs. AIDA-Light achieves a high accuracy with major improvements in efficiency compared to the previous system, AIDA (Hoffart et al., 2011). These improvements make it feasible to run the system on our news corpus. Both systems use a supervised model based on features in the sentential context. For example, one of AIDA-Light's features is the similarity of the context tokens of the *entity mention* to pre-defined key tokens of the *entity candidate* (e.g. the key token *president* for the candidate *President of the U.S.*). AIDA-Light uses simpler features, and adopts a two-stage prediction approach, using a simpler model for mentions that exhibit low ambiguity and reserving more expensive features for more difficult predictions.

The linking step in turn allows Entity Typing, by mapping the IDs to their FIGER

[2] Using a recent implementation of the relation extraction system, the distribution of general and named entity type pairs is fairly uniform (EE: 22.9%, EG: 34.3%, GE: 22.3%, GG: 20.5%).

types (Ling and Weld, 2012). FIGER is a fine-grained hierarchical typing scheme for entity recognition, originally designed to move beyond previous coarser typing schemes. It consists of 112 types — for example, *Lionel Messi* would be mapped to *person/athlete* and *Arsenal* to *organization/sports_team*. General entities (along with unmapped named entities) are typed with the generic *thing* type. Our graphs are based on the first level of the FIGER hierarchy. Using these tools at the argument side we arrive at the typed triples described earlier. The example relation triple can now be instantiated as $beat_{1,2}(Arsenal{:}organization, Manchester{:}organization)$.

Before using this data for the local learning step (Section 2.3.3.2), it is possible to filter noise from the data by applying two thresholds. Hosseini et al. (2018) filter out argument pairs based on the minimum number of predicates they occur with (*minPredforArgPair*), and conversely filter out predicates depending on the minimum number of argument pairs they occur with (*minArgPairforPred*). For example, if *minPredforArgPair* $= 4$, then any argument pair that occurred with fewer than 4 different unique predicates is excluded from the data entirely.

Note that there are many options for improving this filtering step. Chapter 7 provides one such option, using linguistic modality as a way of deciding whether a particular triple should be considered *noisy*. Another option is to develop more sophisticated models for filtering noise from the input data. For instance, it may be feasible to develop supervised models that incorporate various contextual features from the article, one of which could be modality. At a larger scale, it may also be worth taking into account the trustworthiness of the news outlet that published a particular article — formulations of the fake news detection task could become relevant here.

This open-domain relation extraction strategy was chosen to avoid the small and predefined set of relations to which relation extraction systems are often limited. Another related task definition that mines open-domain triples is Open Information Extraction (OpenIE) (Banko et al., 2007), which led to a range of proposed open-domain information extraction systems. OpenIE systems make use of patterns, which may be hand-crafted (Fader et al., 2011; Angeli et al., 2015) or learned through methods such as bootstrapping (Wu and Weld, 2010; Mausam et al., 2012). These patterns may be applied at the sentence level, or to semantically simplified independent clauses identified during a pre-processing step (Del Corro and Gemulla, 2013; Angeli et al., 2015). The majority of systems are restricted to the extraction of binary relations (i.e. relation *triples* consisting of a predicate and two arguments), but systems have also been proposed for the extraction of n-ary relations (Akbik and Löser, 2012; Mesquita et al., 2013). A comprehensive survey of OpenIE systems is provided by Niklaus et al. (2018). Note that these systems often ignore temporal phenomena and modal modifiers, which risks introducing noise in their downstream application. In Chapters 3 and 6 we make alterations to the extraction approach described here to take these phenomena into account.

2.3.3.2 Local Learning of Entailment Rules

Using the set of triples that relation extraction returns, we can learn entailment rules between predicates. Usually, the entailment rules are based on an entailment score between predicates, which can be calculated by their relative distributions of argument pair co-occurrences. The scores are often inspired by the Distributional Inclusion Hypothesis.

The Distributional Inclusion Hypothesis states that if the contexts that occur around a word v also occur around w, then v is expected to entail w (Geffet and Dagan, 2005)[3]. This can be seen as an entailment-specific version of the Distributional Hypothesis (Firth, 1957), which claims that words with similar meanings will have similar contexts in a corpus. The Distributional Hypothesis can be used to define symmetric scores, useful for modeling symmetric relations like similarity or synonymy, whereas the Distributional Inclusion Hypothesis is generally used to inspire directional scores, which are necessary for modeling directional relations like entailment. Directionality should be an essential property for an entailment score, because if an entailment relation holds in only one direction ($p \vDash q \land q \nvDash p$), the scores should be different. For example, we would like the score from *play* to *win* to be low, whereas the score from *win* to *play* should be high.

Applied to predicates, and taking arguments as the context set, we interpret the DIH as follows: if the argument pairs that a predicate p applies to are also arguments of a predicate q, we expect p to entail q[4]. For example, consider the predicates *visit* and *be_in*. Most of the argument pairs (e.g. $\{(Obama, Hawaii)$, $(Obama, London)$, $(Clinton, London), \ldots\}$) that occur with *visit* would be expected also to be observed with *be_in*, which would support the entailment relation *visit* $\vDash$ *be_in*. The unsupervised signal derived from this hypothesis allows us to define similarity scores.

An entailment relation exists between predicates p and q when their entailment

[3] And vice versa, that is if v entails w, then we expect the contexts around v to also occur around w.

[4] When applied to predicates the context set has mostly been taken to refer to argument pairs (e.g. by Berant et al. (2011) and Hosseini et al. (2018)). In Chapters 4 and 5 we suggest that this formulation needs to be sharpened for some predicates, proposing temporality as an auxiliary signal.

score $s_{pq} \in [0,1]$ is larger than some chosen threshold δ. The scores are computed using the set of features $F(p)$ of each predicate p. A feature $f \in F$ is a particular argument pair (a_1, a_2), such as $(Arsenal, Manchester)$. We use $N(p, f)$ to denote the count of feature f occurring with the predicate p in the corpus — the number of occurrences of a triple (p, a_1, a_2). Equivalently $N(p)$ and $N(f)$ are the total counts of p and f in the corpus, and N the total count of observed triples.

When computing scores, the feature values $v(p, f)$ used can be chosen either as $N(p, f)$ for count-based scores or as the Pointwise Mutual Information $PMI(p, f)$ between the predicate and the argument pair for PMI-based scores. The PMI is defined as

$$\mathrm{PMI}(p,f) = log_2 \frac{\mathbb{P}(p,f)}{\mathbb{P}(p)\mathbb{P}(f)},$$

where $\mathbb{P}(n)$ is the probability of n, estimated using the occurrences in the corpus: $\mathbb{P}(p) = N(p)/N$, $\mathbb{P}(f) = N(f)/N$ and $\mathbb{P}(p,f) = N(p,f)/N$. Various scores have been defined according to these feature values.

Lin (1998) define a symmetric similarity score as follows:

$$\mathrm{Lin}(p,q) = \frac{\sum_{f \in F_p \cap F_q} [v(p,f) + v(q,f)]}{\sum_{f \in F_p} v(p,f) + \sum_{f \in F_q} v(q,f)}$$

Their score is based on mutual information: when the information required to describe the features the predicates have in common is similar to the information required to describe the predicate separately, the similarity between the predicates will be high.

Weed's Precision and Weed's Recall are two measures inspired by Information Retrieval methods (Weeds and Weir, 2003), defined as:

$$\mathrm{Weed's\_Prec}(p,q) = \frac{\sum_{f \in F_p \cap F_q} v(p,f)}{\sum_{f \in F_p} v(p,f)}, \mathrm{Weed's\_Rec}(p,q) = \frac{\sum_{f \in F_p \cap F_q} v(q,f)}{\sum_{f \in F_q} v(q,f)}$$

Both Weed's Precision and Recall are directional: Weeds's_Precision(p,q) is not necessarily the same as Weeds's_Precision(q,p). Weed's similarity, also defined in (Weeds and Weir, 2003), is the geometric mean of the two scores.

Balanced Inclusion (BInc) (Szpektor and Dagan, 2008) is defined as the geometric mean of Weed's Precision and Lin's similarity, directional in virtue of the directionality of Weed's precision:

$$\mathrm{BInc}(p,q) = \sqrt{Lin(p,q) \cdot Weed's\_Prec(p,q)}$$

In previous work, BInc has proven particularly successful (cf. Hosseini et al. (2018)), likely because it combines the useful directional properties of Weed's with the non-directional signal of Lin's similarity. In Chapters 4 and 5, we design temporal versions of these similarity scores, by exchanging atemporal features v for temporal features v_t (see Section 4.2.2 in particular). The results in this thesis show that the strength of BInc often carries over to our particular entailment datasets (for example, see results in Section 4.4.2), although the purely directional Weed's score is useful on purely directional evaluation data subsets (see Section 4.4.4).

Finally, Cosine similarity is sometimes used as a symmetric baseline:

$$\cos(p,q) = \frac{\sum_{f \in F_p \cap F_q} v(p,f) \cdot v(q,f)}{\sqrt{\sum_{f \in F_p} v(p,f)^2} \cdot \sqrt{\sum_{f \in F_q} v(q,f)^2}}$$

An entailment score s_{pq} can be calculated using Lin, Weed's Precision and BInc, among other options. The edges E of an Entailment Graph, then, are the set of ordered predicate pairs with scores above a threshold: $\{(p,q)|s_{pq} \geq \delta\}$. If both $s_{pq} \geq \delta$ and $s_{qp} \geq \delta$, we have an entailment relation in both directions and can speak of a paraphrase.

Entailment Graphs are usually evaluated on LIiC-style datasets (see Section 2.2.3.2), using the area under the precision-recall curve (AUC) as a comparison metric. Each point on the curve corresponds to a graph produced with a different threshold δ over the scores — a high threshold corresponds to low recall, with recall over the dataset increasing as the threshold is lowered. The maximum recall is achieved when the threshold is set to 0, which includes all pairs of predicates that both co-occur with some feature.

Within the context of this project we often select BInc as the entailment score due to its strength in previous research. Chapter 4 contains some experimentation with the other scores mentioned here. Furthermore, we mostly use PMI as $v(p,q)$, as this helps prevent word frequency from having an effect on the scores.

Note also that while most research has applied these inclusion-based scores to sparse vector representations such as the argument pair-based ones described here, it may also possible to apply them to dense word representations such as `word2vec` (Mikolov et al., 2013) or contextualized word representations such as BERT (Devlin et al., 2019). In that case the hypothesis would remain that the features of the entailing predicate should generally include those of the entailed predicate. As far as we know this option has not yet been explored in detail (although Hosseini et al. (2019) propose a related ConvE MC entailment score over dense representations).

It is unclear, however, whether existing dense spaces (e.g. those learned by word2vec and BERT) can directly be used for inferences with these particular scores. For the similarity scores above, the sparse vectors have the advantage of an intuitive interpretation of set inclusion: if a particular feature is positive for predicate p (i.e. the argument pair occurs in the data), and it is also positive for q, then that part of the feature set of q is included in that of p. For dense representations this does not work: since the vector's information is shared across the few dimensions it has, there will be very few features with value 0. Therefore if we simply apply BInc as is, we risk summing over all features. Additionally, the embedding values in a dense vector can also be negative (and there is no inherent meaning in their being positive or negative), which has no intuitive correspondence in the set inclusion scores. Note that empirical doubts have recently also been raised regarding the abilities of dense language models on the directional part of NLI (Li et al., 2022).

A promising avenue of future research is to adapt dense representations to this setting: to develop a dense representation space (and associated training method) that adheres to the directional properties that are essential to entailment. This would give access to the benefits of a dense representation, such as the ability to estimate the (directional) similarity of predicate pairs that do not share sparse features in the data[5]. Dense representations are also cheaper to store in memory, and require less compute at inference time (at least according to this paradigm).

A crucial drawback of the sparse representations is that they require a lot of data, preventing them from estimating similarities for many related predicates, and leading to low recall scores. They are straightforward to estimate (by constructing them with the results of machine reading), although their accuracy is limited by machine reading accuracy. The also have the benefits of explainability and interpretability, and most crucially they more intuitively support the entailment estimating strategy described here. For the remainder of this thesis, we will focus on the sparse vectors.

2.3.3.3 Globalization

The advantage of thinking of a collection of entailment rules as a graph is that it gives us access to graph algorithms, allowing us to infer the existence of more entailment rules using the structural properties of the graph. This can be useful because the train-

[5] By placing a predicate in a dense space, it effectively benefits from the training data that supported the predicates around it

ing data may be too sparse to support a wide variety of edges, particularly for low-frequency predicates. We refer to improving a local Entailment Graph in this fashion as *globalization*. Note that we describe these steps here for completeness. Our experimentation focuses on local learning, leaving the interaction of globalization and our proposed methods to future work.

In their original proposal, Entailment Graphs were globalized using an exact Integer Linear Programming (ILP) solution (Berant et al., 2010). The primary constraints were to preserve transitivity of the graph (since entailment is a transitive relation — if $p \vDash q \land q \vDash r$, then $p \vDash r$), and to maximize the sum of graph edge weights. However, given that ILP is NP-complete, their method only scales to graphs of 50 nodes. A series of more efficient methods were developed in response to this limitation, aiming for an approximate solution and again capitalizing on various structural graph properties (Berant et al., 2011, 2012, 2015). This approach attained similar results to the original ILP solution and scales to 20K nodes.

Further scaling has been achieved by using soft constraints instead of hard constraints (Hosseini et al., 2018), making it possible to learn graphs containing hundreds of thousands of nodes. Hosseini et al. (2018) introduced constraints that operate both within a single type graph and between graphs of different types. A paraphrase constraint within graphs dictates that paraphrases should have similar entailments (e.g. if the local graph scores suggest that *win_against* is a paraphrase of *beat*, and *win_against* $\vDash$ *play*, the constraint dictates that the score $s_{beat,play}$ should also be high). The cross-graph constraint states that predicates that are similar between graphs should have similar entailments (e.g. if $win\_against(:organization, :organization) \vDash play(:organization, :organization)$, we can leverage this information in the $(:person, :person)$ graph if the respective *win_against* predicates are similar). Recently, more soft constraints such as soft transitivity have been implemented (Chen et al., 2022).

Existing Entailment Graphs can also be improved using methods trained on other tasks. For example, Hosseini et al. (2019) show how to obtain entailment scores from link predictions scores, showing an improvement over BInc by augmenting the data with additional predicted triples.

2.4 Temporality

2.4.1 Temporality in Language

We now turn to temporal linguistics, before describing the current state of temporal parsing. Broadly speaking, language users have a number of interconnected mechanisms at their disposal to communicate about time, including tense, temporal anaphora, temporal deixis and aspect. Tense and temporal anaphora encode how event times are temporally ordered compared to the utterance time and other times in the discourse. Temporal deixis can be used to instantiate and refer to reference times, accomplished using temporal adverbials in English. The aspectual system, finally, describes an event's temporal internal constituency (Hamm and Bott, 2018).

2.4.1.1 Tense

Tense allows us to indicate whether an eventuality occurs in the past, present or future. Reichenbach (1947) introduces the concepts of points of event (E), speech (S) and reference (R) to this end. Within this framework the purpose of tense becomes establishing the temporal order between these times points — the different tenses can be distinguished in terms of their different orderings.

(10) I see John

(11) I saw John

(12) I had seen John

(13) I will have seen John

For example, in example (10), all three points occur simultaneously (E, R, S), while in the simple past example in (11) the event and reference times occur before the utterance time (E, R—S), meaning that the speaker refers to the past[6]. Examples (12) and (13) illustrate how the introduction of a reference time allows us to construct more complex tenses such as respectively the past perfect (E—R—S) and future perfect (S—E—R).

Reichenbach proposed that tenses be categorized according firstly to their ordering of speech and reference time, and secondly to their ordering of event and reference

[6] In Reichenbach's notation, the comma indicates simultaneity and the em dash represents ordering in time (with elements on the left appearing before elements on the right).

time. As the ordering of speech time with respect to event time is usually irrelevant, this creates a system of nine fundamental tenses. In Reichenbach's terminology, the S,R ordering corresponds to 'past', 'present' and 'future', while E,R ordering corresponds to 'anterior', 'simple' and 'posterior'. This changes the traditional names of some English tenses (included in Table 2.2), for instance preferring 'anterior present' over 'present perfect' (E—S,R)[7]. His system suggested the existence of tenses that traditionally had no name, such as the posterior future (S—R—E) in example (14).

(14) I will be going to see John

A few alterations and expansions of this theory have been suggested. For instance, Prior (1967) points out that the distinction between reference times and speech times is unnecessary: speech time is simply the first reference time. Furthermore, there may be more reference times involved — the conditional past perfect in example (15) requires an ordering E before R_1 after R_2 before S (Declerck, 1986). Another proposed development distinguishes between absolute tenses and relative tenses, reserving the latter for cases where a reference time is needed, and doing away with reference times altogether for the former (Comrie, 1985). However, modeling multiple ordered times (such as E,R,S) has been a consistent feature of later works on tense.

(15) The others would have left by then.

English realizes tense using both auxiliary verbs and morphological inflections (Declerck et al., 2006). In terms of morphology English only distinguishes between present and past, and marks this once in a tensed clause - either on the main verb, or on an auxiliary verb if it is present. More complex tenses are built by adding auxiliary verbs, which may indicate future (e.g. *will*), conditionality (e.g. *would*), or perfectiveness (e.g. *has*). In referring to the future, English sometimes prefers a futurate use of the present tense instead of the future auxiliaries. The correct tense can be inferred by the hearer through context such as temporal adverbials, as seen in example (16). Note that the future tense distinguishes itself from the past tense by being partly temporal and partly modal (Lyons, 1977), since it contains an element of prediction or some other modal concept. For completeness, the traditional names of the tenses are listed in Table 2.2.

[7] We refer to the tenses by their traditional names.

Present Tense	I am in Amsterdam.
Past Tense	I was in Berlin.
Future Tense	I will be in Copenhagen.
Present Perfect	I have been in Dublin.
Past Perfect	I had been in Edinburgh.
Conditional Tense	I would be in Frankfurt.
Conditional Perfect	I would have been in Gdansk.

Table 2.2: Inspired by Declerck et al. (2006)

(16) The train leaves tomorrow.

2.4.1.2 Temporal Anaphora and Deixis

Later research highlighted the crucial role of tense in discourse. This anaphoric role was first recognized through the similarity of tense morphemes and definite pronouns (McCawley, 1971; Partee, 1973). The intuition is that a hearer maintains a discourse model with the previously mentioned event and reference times, and that times of newly introduced eventualities will occur relative in time to those previous points, just as pronouns refer to previous entities in the discourse model. For example, in (17) (example borrowed from Hinrichs (1986)), part of the anaphoric chain includes the time of *going to bed* being anaphoric on the time of *taking a shower*. Previous approaches assume that the reference time moves forward with the discourse narrative for simple past tense sentences, unless otherwise specified by temporal adverbials (Partee, 1984; Hinrichs, 1986; Jordan et al., 1994). Temporal discourse relations can also be indicated more explicitly using temporal subordinating conjunctions such as *until*, *while* and *since* (Hwang and Schubert, 1994).

(17) He took off his clothes, went into the bathroom, took a shower and went to bed.

(18) Sheila had a party last Friday and Sam got drunk.

(19) John went into the florist shop. He had promised Mary some flowers. She said she wouldn't forgive him if he forgot.

Examples such as (18) (Partee, 1973) and (19) (Webber, 1988) revealed that the picture is more complicated, however. In (18), *getting drunk* is a further description of *the party*, so the temporal relation is one of containment, and in (19), the time of *saying* is before the time of *going to the shop*. This led to a more formal account of tense as anaphora. Webber (1988) specifies rules for how the times of the utterance and discourse model can be ordered with one another, incorporating tense, aspect, the event time-reference time distinction and the internal tripartite structure of events (Moens and Steedman, 1988).

New reference times can also be introduced into the discourse through temporal deixis (Lyons, 1977). In English, this is achieved with temporal adverbials such as *now*, *before* or *soon*. Temporal adverbials can be used to specify a variety of temporal properties related to an eventuality (Hwang and Schubert, 1993). They can specify temporal locations (*now*, *yesterday*), durations (*for three weeks*, *forever*), time spans ("He ran the race *in three hours*") and repetitions (*frequently*, *every two years*). These adverbials can also interact compositionally, as in example (20).

(20) "John ran for half an hour every morning for a month"

2.4.1.3 Aspect

While tense refers to how a situation is ordered with respect to reference times, aspect describes the internal temporal constituency of those situations (Comrie, 1985). Within the category of aspect we can further distinguish between grammatical and lexical aspect.

Grammatical aspect refers to whether the situation is attended to as a whole or whether we attend to the internal structure of a situation. In English, this is realized morphologically (such as through the suffixes '-ed' and '-ing'), auxiliary verbs or other phrases (such as 'am' or 'used to'), and combinations between those categories (Declerck et al., 2006). In perfective aspect, we consider the situation in its entirety, as in example (21), where there is no internal structure of the situation to which the speaker can refer.

In contrast, consider example (22), an example of imperfective aspect. Here the situation is ongoing, and one of its subparts can be related temporally to some other situation (introduced through a subclause). Imperfective aspect can be categorized into ingressive, progressive and egressive aspect, which refer to respectively the beginning, middle and end of the situation (Declerck et al., 2006). In English, only progressive aspect can be realized grammatically (example (22)), and speakers do not have access

Type	Instant/Period	Telicity	Example
State	Instant	not unique/definite	She is happy.
Achievement	Instant	unique/definite	They reached the top.
Activity	Period	not unique/definite	He pushed a cart.
Accomplishment	Period	unique/definite	I ran a mile.

Table 2.3: Vendler's Categories of Lexical Aspect

to ingressive or egressive aspect. Instead, these are realized through full aspectual lexical verbs such as 'started' or 'stopped', as in examples (23) and (24).

Another category of grammatical aspect is habitual aspect, referring to situations repeated over time. Here English uses the semi-auxiliary 'used to', as in example (25). For both grammatical and lexical aspect further distinctions have been made that are beyond the scope of this thesis.

(21) I had dinner with Anna yesterday.

(22) I was having dinner with Bobby yesterday, when ... .

(23) I started having dinner yesterday, while ... .

(24) I stopped having dinner yesterday, when ... .

(25) I used to have dinner with Carlos every night.

Lexical aspect, instead of being realized grammatically, depends on lexical knowledge. This category was first introduced by Vendler (1957), who classified verbs into states, activities, accomplishments and achievements, depending on whether they pertain to time points or periods and whether they have a natural endpoint[8]. Examples are given in table 2.3. Although many extensions have since been given, these categories under varying names have been a mainstay in the literature. We use the term *eventuality* to refer to the general class containing all these types.

Being able to draw these distinctions is important for accurate formal semantics of events, since lexical aspect has an effect on truth values and entailment. For instance, the aspectual category has an impact on entailment between tenses of the same verb

[8] Note that the instant-period distinction was originally made by Vendler, but has since been reanalyzed to mean whether or not the situation is both dynamic and durative (Smith, 2013). Being an instant or a period essentially determines whether the situation is a process (and whether it appears felicitously with the progressive).

phrase, as shown in examples (26) and (27). These cases portray how entailment does not hold between the progressive and regular past tense of an *accomplishment*, while it does for an *activity*. In general, lexical aspect also informs us whether we are modeling points or durations in time, and whether some specific endpoint needs to be taken into account. Thus, to be able to formalize for which specific times a situation holds, lexical aspect needs to be understood.

(26) I was running. $\Rightarrow$ I ran.

(27) I was running a mile. $\not\Rightarrow$ I ran a mile.

This also illustrates the possibility of changing the aspectual class of a phrase by adding context, through a process called aspectual type coercion (Moens and Steedman, 1988). Various types of context can lead to aspectual class changes. For instance, adding auxiliaries and morphological inflection can coerce a *point* class into an *activity* (examples (28) to (29)). Involving temporal adverbials can change an accomplishment into an activity (examples (30) to (31)), and the same change can be achieved through adding specific arguments (examples (32) to (33)).

(28) Mary hiccupped.

(29) Mary was hiccupping.

(30) Tony played Canto Ostinato.

(31) Tony played Canto Ostinato for a few minutes.

(32) Pat drank.

(33) Pat drank a beer.

Often, world knowledge is required for reinterpretation of aspect. Consider example (34), which would normally be considered infelicitous, as it involves a *for* adverbial used with an accomplishment (Van Lambalgen and Hamm, 2008). However, this can be reinterpreted as an iterated activity, which does permit *for* adverbials. For that reinterpretation we require the knowledge that playing Opus 111 only takes a limited amount of time, and in this context probably refers to the event occurring repeatedly over a time span. This stands in contrast to (31), for which we require world knowledge to understand that a musical piece of indefinite length is being played just briefly.

(34) Pollini played Opus 111 for two weeks.

Moens and Steedman (1988) also introduce the event nucleus, conceiving of a situation as a tripartite structure containing a preparatory phase, a culmination and a consequent phase. The existence of these primitive phases acknowledges that temporal expressions in language are also about consequence, causation, preparation and planning, and not exclusively about time. Furthermore, they allow for temporal reference to specific parts of a situation, crucial to resolving the types of temporal discourse paradoxes described in Section 2.4.1.2. Again, both the idea of an event nucleus and of aspectual coercion are valuable to formal semantics, as they inform how representations can be built and altered.

2.4.1.4 Temporality and Entailment

Temporality influences the entailments of a predication in a multitude of ways. Perfect aspect (typically) describes events as a completed whole, and licenses inferences regarding the consequences of that event. In particular, the consequences of an event in the present perfect hold at the time of utterance, whereas events in the simple past or the past perfect do not (Comrie, 1985; Moens and Steedman, 1988; Depraetere, 1998). This is shown below. While the present perfect example (35) does license the inference to the present tense *is in*, both example (36) and example (37) do not.

(35) Elizabeth has gone to Meryton. $\models$ Elizabeth is in Meryton now.

(36) Elizabeth went to Meryton. $\not\models$ Elizabeth is in Meryton now.

(37) Elizabeth had gone to Meryton. $\not\models$ Elizabeth is in Meryton now.

This property can be explained through a Reichenbachian view of the present perfect, where the point of reference coincides with the point of speech, thereby indicating its current relevance (Reichenbach, 1947). On the other hand, events in the past simple or the past perfect license inferences for consequent states in the past, as examples (38) and (39) show.

(38) Elizabeth went to Meryton. $\models$ Elizabeth was in Meryton.

(39) Elizabeth had gone to Meryton $\models$ Elizabeth was in Meryton.

Progressive aspect describes ongoing events and therefore does not license inferences regarding their consequences as example (40) shows. It furthermore gives rise to the imperfective paradox (Dowty, 1979), which only licenses inferences for non-culminated processes (Moens and Steedman, 1988), as examples (41)-(43) below show.

(40) Mary is going to Netherfield now. $\not\models$ Mary has arrived / is in Netherfield.

(41) Catherine was walking in the woods. $\models$ Catherine walked in the woods.

(42) Jane was reaching London. $\not\models$ Jane reached / was in London.

(43) Jane was reaching London. $\not\models$ Jane was in London.

Alongside developments of linguistic theories of tense, aspect and time, theories were developed regarding how to logically represent natural language descriptions of situations. Examples include the introduction of Davidsonian event variables (Davidson, 1967), the situation calculus (McCarthy and Hayes, 1969), the event calculus (Kowalski and Sergot, 1989), the dynamic event calulus (Moens and Steedman, 1988) and contributions in the form of flattened formal semantics (Parsons, 1990). As these calculi essentially had as their goal to facilitate computation with events and time in language, they can be seen as the forerunners of modern temporal NLP.

2.4.2 Temporality in Natural Language Processing

Given a document of text, temporal reasoning in NLP has focused on predicting as much information as can be inferred about *when* the mentioned eventualities occur. We focus here on this task - determining the temporal extent of eventualities, as we use systems performing this task as subcomponents of the algorithms presented in Chapters 4 and 5. We briefly mention related temporal processing problems in Section 2.4.2.4.

Temporal relation extraction systems are expected both to recover explicit and implicit temporal information. In explicit cases, a unit of temporal information may be referred to directly, as in “They arrived on March 26th, 2022”, or “They visited every second Sunday of the month”. Implicit information will require further reasoning, such as connecting an eventuality to a reference time mentioned elsewhere in the document, as with the temporal adverbials “*A week after* they arrived, ...” or “*Two days before that*, ...”. Sometimes this reasoning is performed on the basis of typical temporal orderings between eventualities, without any temporal marker. For example, in “They arrived on March 26th, 2022. The journey was pleasant”, the reference time of *was* is before the *arrival*, while in “They arrived on March 26th, 2022. The visit was fantastic” the reference time of *was* would be after the *arrival*. Systems may also be expected to involve external world knowledge, such as "They arrived the weekend before Easter", or "They arrived after the 56th Superbowl."

2.4.2.1 Tasks and Annotation

These types of reasoning have been represented recently in NLP in the TempEval shared task series (Verhagen et al., 2007, 2010; UzZaman et al., 2013). Recent work has focused on the clinical domain with the Clinical TempEval series (Bethard et al., 2015, 2016, 2017), because of the immediate practical applicability of temporal understanding of doctors notes in conjunction with patient data (Olex and McInnes, 2021).

The temporal reasoning demanded in the TempEval series boils down to reproducing the annotations in the family of temporal specification languages based on TimeML (Pustejovsky et al., 2003a). TimeML annotates the eventualities, time expressions and temporal orderings in a document, and the TempEval subtasks correspond to these elements. Firstly, systems are expected to recognize the eventualities in the document, along with certain temporal attributes such as tense, grammatical aspect, polarity and modality. Secondly, the time expressions in the document are recognized and parsed into one of several structured temporal types: Time (for specific times of day such as "at half past twelve"), Date (Referring to parts of the calendar, e.g. "On Saturday April 2nd"), Duration (e.g. "for two months") or Set (for repeating elements such as "every day"). Lastly, TimeML distinguishes itself by annotating various types of relations between the identified events, the time expressions and the DCT.

These three tasks together allow a system to estimate when eventualities in a paragraph of text happened. The temporal algorithm work in this thesis (Chapters 4 and 5) stands to benefit from solutions to all three tasks when they help determine a specific time (rather than just an ordering). Task 2 of temporal expression recognition and parsing is particularly relevant, since we use the SUTime system that performs that task (using a dependency parse to link from time expressions to events (see Section 4.2.1)).

For example, take example sentence (44) from an article published on Saturday 09/02/2013, taken from the NEWSSPIKE corpus (Zhang and Weld, 2013). In this case, SUTime recognizes that *Friday* is a temporal expression, and returns an estimated beginning and end time grounded in calendar dates: [08/02/2013, 08/02/2013)]. The system is able to understand that, given the past tense of the sentence, the Friday referenced is most likely the one just before the article's publication date. In Chapter 4 we describe how these intervals are linked to relation triples and used in the learning algorithm.

(44) The Miami Heat dominated the middle two quarters on the way to an easy 111-89 win over the Los Angeles Clippers on Friday night.

For completeness we also discuss other facets of the task. Some systems also provide output describing relations between different expressions in an article. The majority of relation annotations are TLINKs, for temporally ordering eventualities and times. TimeML also annotates the subordinating SLINK, for connecting events that involve modality, evidentiality and factivity, such as connecting *said* and *arrived* in "He said they arrived". The aspectual ALINK connects aspectually related eventualities, such as *finished* and *racing* in "... when they had finished racing".

The TLINK orderings are {BEFORE, AFTER, INCLUDES, IS INCLUDED, DURING, DURING INV, SIMULTANEOUS, IAFTER, IBEFORE, BEGINS, ENDS, BEGUN BY, ENDED BY}, along with the IDENTITY link for coreferring eventualities. These correspond loosely to the 13 possible temporal relations between intervals specified in Allen's interval algebra (Allen, 1983), although the *overlap* and *overlapped by* relations are not represented (UzZaman and Allen, 2011). The algebra also defines how to infer new ordering relations by computing the transitive closure over a set of given relations. For example, if we know for intervals X, Y and Z that X $\{before\}$ Y and Y $\{before\}$ Z, by transitivity we conclude X $\{before\}$ Z. Inferred relations can also be ambiguous. For example, if X $\{finishes\}$ Y and Y $\{started\ by\}$ Z, we can infer a set of possible orderings: X $\{after, met\ by, overlapped\ by\}$ Z. Recent work has suggested moving to the more computationally efficient *point temporal algebra*, using just three possible point-level relations ($\{<,=,>\}$) between two sets of start and end points, instead of the thirteen relations over two intervals (Freksa, 1992; Leeuwenberg and Moens, 2019).

TimeML annotations have been refined into ISO-TimeML (Pustejovsky et al., 2010), THYME-TimeML (Styler IV et al., 2014) and, for time expressions specifically, SCATE (Bethard and Parker, 2016). These allow the annotations to better capture the potential complexity of the temporal reference system. For example, in ISO-TimeML, the MLINK was introduced to allow both the contiguous and non-contiguous interpretations of duration intervals, which are *measures* of the eventuality with which they are linked. The sentence "She trained for 4 hours today" is ambiguous between a contiguous reading in which the training happens in a single block, and one in which the 4-hour measure is spread out over the day. In SCATE, annotations become compositional, to better support the compositionality of temporal expressions. For example, SCATE improves over TimeML by supporting time expressions that contain unisons over simpler expressions (e.g. “every Tuesday and Thursday in March”) and that refer to multiple parts of the calendar (e.g. “the summers in 2015 and 2016”). It has been challenging to achieve high inter-annotator agreement for these tasks, and new annotation schemes such as MATRES have been proposed in response (Ning et al., 2018b).

2.4.2.2 Datasets

Numerous training and evaluation datasets have been annotated using these annotation styles. The original TempEval tasks focused on the news domain with TimeBank (Pustejovsky et al., 2003b), the first dataset in the TimeML specification. They also introduced the AQUAINT TimeML Corpus and the TempEval3-Silver, -Gold and -Platinum datasets (UzZaman et al., 2013), the latter having been annotated by experts for system comparison. Later, the medical-domain evaluation focused on the THYME dataset (Styler IV et al., 2014), which contains around 1,200 documents of clinical notes. These are temporally rich, as they typically contain descriptions both of a patient’s medical history, recent changes and test results, and future treatment plans.

Evaluation has also branched out to domains such as encyclopedic text, historical text and fictional text (Mazur and Dale, 2010; Strötgen et al., 2014; Rogers et al., 2019), as well as expanding to multilingual evaluation options (for example, the German KRAUTS (Strötgen et al., 2018) and German version of WikiWars, (Strötgen and Gertz, 2011), and French, Korean and Hindi versions of TimeBank, (Bittar et al., 2011; Jeong et al., 2016; Goel et al., 2020). A number of specific temporal features have been explored, often by reannotating or using additional annotations on TimeBank. Added features include temporally dense annotations that expand the number of allowed links per document (Cassidy et al., 2014), annotations of the durations of eventualities (Pan et al., 2006), and causal information (Mirza and Tonelli, 2016). Causality has also been explored in the CaTeRS dataset (Mostafazadeh et al., 2016). Annotations are particularly labor-intensive (with n eventualities and times, the number of potential links grows at n^2), and the datasets have been relatively small in consequence, normally containing fewer than 100,000 tokens.

2.4.2.3 Systems

The systems resulting from these shared tasks have been essential for temporal processing of language. For our purposes, they make it possible to automatically annotate a large corpus with temporal information, which can then be used as a signal in Entailment Graph learning.

A number of strategies have been attempted in solving these problems. Early state of the art systems such as HeidelTime (Strötgen and Gertz, 2010) and SUTime (Chang and Manning, 2012) performed rule-based time expression annotation. They used regex-based string matches and applied handcrafted rules to normalize the extracted strings. UW-Time (Lee et al., 2014) approaches the problem with semantic parsing, using a hand-engineered lexicon with the log-linear CCG of Clark and Curran (2007) to rank possible meanings for the mentions in the document.

Later approaches found success by combining various types of learning. For instance, CAEVO (Chambers et al., 2014) blends multiple rule-based and statistical learners in a cascaded sieve ranked by precision. It was later improved by implementing a *generalized* prediction ranking method to replace the coarse-grained sieve, with further improvements from word embeddings and semantic features (McDowell et al., 2017). CogCompTime (Ning et al., 2018c) uses a Beginning-Inside-Outside (BIO) chunking classifier and rule-based parsing for the time expression recognition task, combined with Integer Linear Programming to attain improvements in the temporal ordering task. Here they use the temporal transitivity constraints that are inherent in the graph of temporal relations between eventualities.

Finally, variants of neural networks and the contextualized word embeddings drawn from transformer architectures such as BERT (Devlin et al., 2019) have recently been applied to the problem. Dligach et al. (2017) were among the first to use neural networks, comparing the performance of Bidirectional LSTM and Convolutional Neural Network (CNN) architectures. Neural networks' limitation of requiring a large amount of high-quality training data have been mitigated using a self-training framework (Lin et al., 2018) over RNNs. They later improved on their method by combining it with BERT (Lin et al., 2019), focusing on the CONTAINS temporal relation. Embeddings from other larger transformer-based models have also been compared, showing that RoBERTa-large performed best amongst the selected models (Guan et al., 2021).

The CogCompTime system's ILP approach (Ning et al., 2018c) has also been improved using contextualized BERT embeddings (Ning et al., 2019). They also find that adding temporal commonsense knowledge of typical orderings from the TEMPROB knowledge base (Ning et al., 2018a) is useful, where the knowledge is generalized to unseen pairs using a Siamese network (Bromley et al., 1993).

2.4.2.4 Related Problems

Related problem formulations can build an even richer understanding of the temporal properties of a piece of text. Temporal commonsense reasoning may be useful in modeling more implicit temporal knowledge, such as the typical duration and typical frequency of eventualities (Zhou et al., 2020). For instance, their models predict that a *vacation* takes days or weeks whereas a *walk* takes minutes or hours. Another type of commonsense knowledge is that of implicit eventualities. Zhou et al. (2021) use a textual entailment-style setup to test knowledge of the temporal ordering between a given implicit eventuality and an eventuality mentioned in the text. For example, given a story about a *visit*, they might query the relative time of the *arrival*. Classifying the lexical aspect of eventualities has also been explored in its own right (Kober et al., 2020). Finally, event co-reference is important to temporal understanding, since event coreference chains can be used to propagate temporal ordering relations (if two event mentions co-refer, their relations should be identical). Jointly modeling event co-reference and temporal relation extraction has been shown to support both tasks (Teng et al., 2016).

2.5 Modality

2.5.1 Modality in Language

Finally we discuss the semantic phenomenon of modality. Modality contains a range of devices that speakers can use to refer to conceivable states of the world that might or might not occur, together with their attitude toward the propositional content of their utterance. Our main interest in modality in this project is its usefulness in determining the *factuality* of a predication: whether the eventuality is asserted as actually occurring in the real world. We can then take this into account in our downstream application of Entailment Graph induction.

Pyatkin et al. (2021b) point out that modality and factuality are two related but orthogonal concepts. Different kinds of modality can result in a positive factuality status (compare “Arsenal succeeded in winning.” and “John confirmed that Arsenal won”), while different kinds can also results in a negative factuality status (e.g. epistemic modality in "Arsenal might win tonight" and deontic modality in "Arsenal needs to win tonight."). We analyze modality because the type of modality resulting in non-factuality is still relevant to the semantics. Although we only use the factuality distinction in this project, the more fine-grained modal distinctions are likely to be relevant in future research, as these may have different implications for entailment learning. We use "modal" and "modalized" to refer to modalities that result in non-factual expressions.

We handle various semantic phenomena that can broadly be considered modal (e.g. conditionality, propositional attitude) along with negation. We first discuss the typical, more specific category of modality here.

2.5.1.1 Modality

The formal semantics of these notions was originally built on the logic of knowledge and belief (Hintikka, 1962) and the logic of possible world semantics (Kripke, 1963). In this quantificational view of modality, a modal expression quantifies over some coherent, restricted set of possible states of the world (with universal or existential quantification). For example, "Mary believes that it's raining." might be analyzed as "In *all* possible worlds that are compatible with Mary's beliefs, it is raining." (Lassiter, 2017).

Lewis (1973) and Kratzer (1981) refined the quantificational models for a broader variety of expressions. For example, Kratzer (1981) provides a formal analysis of modal degree modification (expressions such as *kann gut sein* (English: *can well be*) and *geringe Möglichkeit* (slight possibility)) — these require more fine-grained logical tools than just the quantifiers $\forall$ and $\exists$. Her solution depends on an ordering over possible worlds, defined according to how well each of the worlds satisfies the ordering-source propositions of a modal expression. Incidentally, this solution is also able to encode some entailment relations, for example *can well be* $\models$ *can be*.

Recently Lassiter (2011, 2017) has proposed moving past quantificational models, instead building a scalar basis of modal semantics. This would reflect more accurately the similarity of modal expressions to gradable expressions, such as *hot* and *heavy*. Lassiter (2011) argues that, like *temperature* or *weight*, modality pertains to concepts that admit of degrees, that can be ordered on a scale. He argues that modal adjectives (such as *likely*, *plausible*) are clearly scalar and that attitude verbs have grammatically similar properties to gradable expressions. Rethinking modal semantics from the ground up from a scalar perspective also allows analysis of the compositionality of modal expressions, which presented challenges in the approach taken by Kratzer (1981), and it is possible to extend the analysis even to expressions that do not immediately show evidence of gradability, like the modal auxiliaries (e.g. *must*, *may*). The analysis also permits inclusion of probability theory and Bayesian ideas from cognitive science (Lassiter, 2017), which have evidence of being (more) cognitively plausible (Lassiter and Goodman, 2015)

Typological definitions of modality focus on categorising the speaker's attitude, such as epistemic necessity (*That must be John.*), epistemic possibility (*It might rain tomorrow.*), deontic necessity (*You must go.*), and deontic possibility (*You may enter.*) (Van Der Auwera and Ammann, 2005). Categories such as *desire* can also be used to describe the speaker's attitude (Hacquard, 2006). Sometimes a lexical trigger of modality is ambiguous between categories; English *may*, for example, is ambiguous between an epistemic possibility reading (*It may rain tomorrow.*) and a deontic possibility reading (*You may enter.*)

In English, modality can be expressed in a variety of ways. The modal auxiliaries (e.g. *might, should, can*) are commonly used, but modality can be lexicalized in many other trigger words. Nouns (e.g. *possibility*), adjectives (e.g. *obligatory*), adverbs (e.g. *probably*) and verbs (e.g. *presume that*) can all indicate modality. In the long tail, speakers have access to vastly productive phrases that indicate their attitude. The following examples occurred naturally in the news domain (Zhang and Weld, 2013): *That's how close they were to ... ., I cannot come up with a scenario that has ... ., That's based on the world wide assumption that ... .*).

2.5.1.2 Conditionality

A conditional sentence is composed of a subordinate clause (which we will refer to as the antecedent) and a main clause (the consequent). The antecedent and consequent are connected by a conditional conjunction (which in English is often the word *if*) (Dancygier, 1998), for example, *if they attack there will be war*. Conditional sentences can have a variety of semantic interpretations, but the most commonly studied, the *hypothetical* conditional, expresses that the consequent (*there will be war*) will hold true when the antecedent (the *attack*) is satisfied (Athanasiadou and Dirven, 1997). For our purposes, the most important part of their semantics is that neither the antecedent nor the consequent are normally entailed by the sentence, so that the speaker is not committed to their truth.

2.5.1.3 Counterfactuality

In the counterfactual construction a more complicated semantic relation is established between antecedent and consequent, for example: *Had they protested, they would be content*. As with modality, this has been formalized more precisely with a possible world semantics (Lewis, 1973; Kratzer, 1981). With a counterfactual, the speaker communicates that in any world similar to the current one, differing only by the proposition in the antecedent, the consequent would hold true (Lewis, 1973). In the above example, if the world is altered by the *protest* in the antecedent, *they would be content* holds true. Again, the crucial semantic information for our work is that neither the antecedent nor the consequent are entailed.

2.5.1.4 Negation

Negation is a semantic category used to change the truth value of a proposition in order to convey that an eventuality does not hold (Horn, 1989). It may be expressed explicitly using various means, most notably closed-class function words such as *not, no, never, neither, nor, none* and *without*, but can also be expressed lexically in open grammatical categories such as nouns (e.g. *impossibility*), verbs (e.g. *decline*, *prevent*), and adjectives (e.g. *unsuccessful*). It may also be expressed implicitly, such as with combinations of certain verb types and tenses (e.g. *The polls were supposed to have closed at midnight*). In this work we consider only explicit cues of negation.

2.5.1.5 Propositional Attitude and Evidentiality

Propositional attitude allows speakers to indicate the cognitive relations that entities bear to a proposition (McKay and Nelson, 2000). For example, in *Republicans think that Trump has won*, the speaker expresses that *Republicans* hold certain beliefs. In English, such reports are often made using propositional attitude verbs such as *claim, warn* or *believe*. Normally only the entity's thoughts regarding the eventuality are entailed, not the eventuality itself. Propositional attitudes are often used as markers of *evidentiality* in English (Biber and Finegan, 1989; McCready and Ogata, 2007). These are important in Question Answering. For example when answering a question using the sentence *The Kremlin says protesters attacked the police* as evidence, mentioning the source (*The Kremlin*) might be particularly important.

2.5.2 Modality in Natural Language Processing

Understanding the phenomena in Section 2.5.1 is valuable to NLP, because many tasks require knowledge of the speaker's attitude towards an uttered proposition. Factuality can be particularly important, because tasks such as Information Extraction, Question Answering and Knowledge Base Population all depend on knowing whether described eventualities are asserted as occurring (Karttunen and Zaenen, 2005; Morante and Daelemans, 2012).

2.5.2.1 Annotation Schemes and Datasets

NLP investigates these concepts under various formulations, resulting in overlapping task definitions and naming conventions. For example, Karttunen and Zaenen (2005) talk of veridicity, while there is also work on uncertainty detection (Szarvas et al., 2012; Vincze, 2014), hedge detection (Medlock and Briscoe, 2007) and modality annotation (Saurı et al., 2006). Therefore, datasets that contain modal phenomena are associated with many different annotation schemes.

The terms veridicity, uncertainty detection and factuality refer broadly to the same phenomenon — whether the event in question actually occurs. The labeling schemes vary widely, but usually consist of three labels, corresponding to *asserted*, *uncertain*, and *negated*, where the middle category can often be split into more fine-grained notions of uncertainty. In the factuality datasest FactBank (Saurí and Pustejovsky, 2009), for example, eventualities are labeled as *certain*, *probable* or *possible*, along with their negated variants (**not** *certain, probable, possible*) and *uncommitted* options. In example (45) from the FactBank corpus, the verb *led* is annotated as *certain*; that is, it is certainly the case that the event happened according to the source of the sentence. Similar discrete labels are adopted in the more recent MegaVeridicality (White and Rawlins, 2018; White et al., 2018); annotators are asked the question "did that thing happen?" or "did that person do that thing?", with the possible labels *yes, maybe or maybe not, no*.

Other recent datasets in event factuality have adopted a [-3,3] annotation scale, including the UW event factuality dataset (Lee et al., 2015), Unified Factuality (Stanovsky et al., 2017), the Universal Decompositional Semantics "It Happened" dataset (UDS-IH1) (White et al., 2016) and its expansion, UDS-IH2 (Rudinger et al., 2018). For example, in sentence (46), the eventuality *said* receives an annotation of 3.0 (highly certain), while *trade* is annotated as -0.8.

(45) Scott Ritter **led** his team on a 10-hour tour of three suspected weapons sites classified as "sensitive" by the Iraqi authorities, U.N. spokesman Alan Dacey said.

(46) He also **said** of **trade** with Iraq: "There are no shipments at the moment."

These datasets encompass many genres. For example, FactBank (Saurí and Pustejovsky, 2009) is built on the temporal TimeBank (Pustejovsky et al., 2003b), consisting of newswire and broadcast news reports. CommitmentBank (De Marneffe et al., 2019) annotates events from the newswire, fiction and dialog genres. Kim et al. (2008) add event annotation to the Genia corpus of biological text, and annotations of multilingual data are available in the MEANTIME corpus (Minard et al., 2016).

Hedge detection can be seen as a specific kind of uncertainty detection, used to analyze uncertainty stemming specifically from authors qualifying their statements. For example, a scientist might qualify their result, as in "Our results *suggest* that XfK89 *might* inhibit Felin-9." (Medlock and Briscoe, 2007) . Hedge detection datasets usually focus on annotating cues, which can be connected to events. For example, the medical domain BioScope (Szarvas et al., 2008) is annotated for hedge cues, negation cues and their scope. This corresponds loosely to the three labels used in uncertainty detection, since an event that is not under scope of any cue can be considered *asserted.* Hedges also occur in the encyclopedia domain, explored in the WikiWeasel corpus of the CoNLL 2010 hedge detection shared task (Farkas et al., 2010).

Annotating modality presents a larger range of phenomena to analyze. Prabhakaran et al. (2012) propose five classes of modality: *ability*, *effort*, *intention*, *success*, and *want*, while Baker et al. (2010) use a more extensive set of classes that includes *requirement*, *permissive*, and *belief.* In a pilot alongside the CLEF 2011 QA shared task, Peñas et al. (2011) ask whether predications are *asserted*, *negated*, *uncertain*, or describe a *condition* or a *purpose* for another event. Saurı et al. (2006) enrich the TimeML specification language with yet other categories (e.g. evidentiality and conditionality).

2.5.2.2 Models

Early approaches to detecting modality focused on lexicon design (Szarvas, 2008; Kilicoglu and Bergler, 2008; Baker et al., 2010). The strategy employed is usually to construct a lexicon containing modal trigger words. These words are then recognized in the context of a sentence, after which their scope can be predicted, revealing which predicates are affected. Many early systems were purely rule-based (Lana-Serrano et al., 2012; Pakray et al., 2012), or combined rules with the output of a parser (Rosenberg et al., 2012).

Baker et al. (2010) employ two strategies for tagging modal triggers and their targets: 1) string and POS-tag matching between entries in a modality lexicon and the input sentence, 2) a structure-based method which applies rules derived from the lexicon to a flattened dependency tree, inserting tags for modality triggers and targets into the sentence. They use a set of eight modality tags, which we expand in Chapter 6 to cover a wider range of phenomena, including conditionality and propositional attitude. Our work is inspired by this lexicon and parse tree-based approach, in part due to the lack of a large, open-domain modality training dataset.

Modality tagging has also been cast as a supervised learning task, in which classifiers are trained using crowd-sourced annotated data (Morante and Daelemans, 2009; Rei and Briscoe, 2010; Prabhakaran et al., 2012). While performance is reasonably strong on in-domain data, out-of-domain data can still prove challenging for these models (Prabhakaran et al., 2012). The state of the art in uncertainty detection and modality annotation has experienced similar shifts as NLP tasks in entailment and temporality. Jean et al. (2016) show the benefits of building sentence-level vector representations, Adel and Schütze (2017) explore attention mechanisms in neural network-based solutions, and Rudinger et al. (2018) experiment with dependency tree LSTMs. Again, this has been extended to transformer architectures — models that incorporate RoBERTa achieve strong results (Pyatkin et al., 2021b).

Chapter 3

Temporal Entailment and Tensed Entailment Graphs

3.1 Introduction

Temporal entailments, such as the inference that if somebody *is visiting* a location, they *have arrived* there already, allow us to correctly answer a question like “Has Obama arrived in Hawaii?” from textual evidence like “Obama is visiting Hawaii”. Since they include time, they avoid incorrect answers that might be given with a more general rule *visit* $\vDash$ *arrive*, which might prompt the system to respond **True** even if the evidence is “Obama will visit Hawaii”.

Temporal inferences such as these were previously unexplored, so initial research centered around defining the problem and designing a new evaluation dataset, TEA (Section 3.2), published as (Kober et al., 2019).

We then experimented with representing these entailments in a model we call Tensed Entailment Graphs (TsEG), showing that they capture some entailment relations that regular EGs are unable to represent. This work is presented in Section 3.3 and has remained unpublished. We also discuss various challenges in both entailment dataset design and Entailment Graph induction.

3.2 Temporal Entailment

In order to make progress on modeling temporality in Entailment Graphs, we first need clearer definitions of what these models are trying to find, along with a dataset for evaluating the models.

3.2.1 Definition

For brevity, we will use the term entailment to refer to entailment between two typed binary predicates, as used by Hosseini et al. (2018). That is, we investigate predicates for which the types of the two arguments are specified, such as *buy*(*person*,*object*). We also interpret entailment as common-sense inference (Dagan et al., 2006) (see also Section 2.2.1), which has become the standard interpretation for evaluating whether systems are able to draw semantic inferences. In this context, a text T *entails* a hypothesis H if "typically, a human reading T would infer that H is most likely true."

Then, within the work in Sections 3.2 and 3.3, we define a temporal entailment task as one in which the texts in the dataset are centered on different morphosyntactically tensed (Deo, 2012) predicates. The tense interactions inform us about the temporal relationship between the two predicates. For example the pair "John has bought the laptop"-"John owns the laptop" requires some sort of entailment rule like *has_bought* $\models$ *owns*, which tells us that once somebody *buys* something they must *own* it afterwards[1]. We keep this definition broad, allowing pairs of predicates to be connected even if the tenses are identical. However, the cases in which tenses are different, such as the one above, are particularly interesting, since they inform us of the temporal ordering of the predicates and may include more complex lexical relations than atemporal troponomy.

In previous work, the question of how to represent an entailment relationship like *divorce* $\models$ *marry* was left unaddressed. This can create issues, since the knowledge of the atemporal entailment *divorce* $\models$ *marry* is not a sufficiently rich representation to distinguish between non-entailments such as *divorce* $\not\models$ *will marry* and entailments such as *divorce* $\models$ *was married.* If a model contains the rule *divorce* $\models$ *marry* and applies it blindly, it might incorrectly predict that there is an entailment between "They will divorce next week" and "They will marry next week". Conversely, if one avoids modeling the relationship at all, the model will incorrectly predict a non-entailment between "They divorced" and "They were married".

Models were not forced to represent this knowledge because previous datasets did not focus on these distinctions. Moving to datasets containing temporal entailment therefore encourages us to more accurately model the relationship between predicates.

[1] Note that temporal entailment is a specific case of the original entailment definition - there is nothing excluding these types of interactions from the original datasets, since those texts also contained tensed predicates. The main difference is that our dataset construction methods focus on the interactions between the tenses.

John is visiting London.	$\models$	John has arrived in London.
John will visit London.	$\not\models$	John has arrived in London.
John is visiting London.	$\not\models$	John has left London.
John is visiting London.	$\models$	John will leave London.
George has acquired the house.	$\models$	George owns the house.
George is acquiring the house.	$\not\models$	George owns the house.

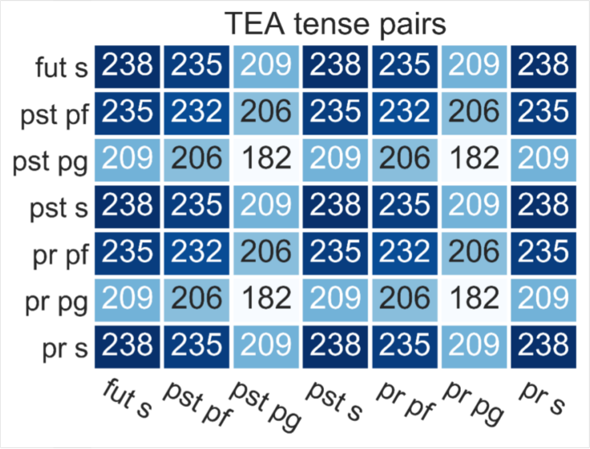
Table 3.1: Examples From TEA

Temporal entailments allow us to model, for instance, that *has divorced* entails *was married* while *is divorcing* entails *is married*. Note that there are other ways of temporally extending this definition, such as through lexical relations such as *consequence*, or through temporal orderings. We discuss these as future research possibilities in Section 8.2.

3.2.2 Dataset Creation

We developed TEA — the Temporal Entailment Assessment dataset — to evaluate models under the definition of entailment. We cast the problem as a natural language inference task, following a binary label annotation scheme (entailment vs. non-entailment). TEA contains pairs of short sentences with the same argument structure that differ in the tense and aspect of the main verb, such as the ones shown in Table 3.1.

There were previously no resources for specifically evaluating models that learn entailments of a temporal nature. The dataset of Levy and Dagan (2016), improved by Holt (2018), is commonly used for evaluating typed predicate Entailment Graphs, but contains virtually no temporality. The FraCas test suite (Cooper et al., 1996) contains a small section of temporal cases, but only a few examples are entailments between predicates. While MNLI (Williams et al., 2018) claims to contain some examples that involved temporal reasoning, it would be difficult to filter out the necessary cases as they are not explicitly annotated. This might be possible using temporal keywords such as *yesterday*, but it would not provide us with the temporally informative lexical entailments between predicates that we focus on. Other popular entailment datasets (see Section 2.2.3) also lack the desired properties.



	fut s	pst pf	pst pg	pst s	pr pf	pr pg	pr s
fut s	238	235	209	238	235	209	238
pst pf	235	232	206	235	232	206	235
pst pg	209	206	182	209	206	182	209
pst s	238	235	209	238	235	209	238
pr pf	235	232	206	235	232	206	235
pr pg	209	206	182	209	206	182	209
pr s	238	235	209	238	235	209	238

Figure 3.1: Frequencies of Tense Pairs in TEA; pst = past, pr = present, fut = future; s = simple, pg = progressive, pf = perfect.

3.2.2.1 Data Collection

We sampled candidate pairs from the before-after category of VerbOcean (Chklovski and Pantel, 2004), the WordNet verb entailment graph (Miller, 1993), the entailment datasets of Weisman et al. (2012) and Vulić et al. (2017), and the relation inference dataset of Levy and Dagan (2016). Subsequently, we manually filtered the list, and discarded candidate verb pairs without any temporal relation to each other. For each pair we chose nouns as arguments to form full sentences. The arguments further served the purpose of reducing ambiguity, for instance by avoiding habitual readings. For example, the sentence "The farmer harvests crops" allows for a habitual reading, which is made less salient with a definite pronoun in "The farmer harvests the crops". Choices like this enable our dataset to concentrate on specific eventualities instead of a series of of eventualities.

TEA covers entailments between an all-by-all combination of the present simple, present progressive, present perfect, past simple, past progressive, past perfect and the modal future, covering perfect and progressive aspect. The dataset contains 11,138 sentence pairs with a class distribution of 22 : 78 (entailment : non-entailment). Figure 3.1 presents a heatmap of different tense pair frequencies in the dataset.

3.2.2.2 Data Annotation

We interpreted entailment as common-sense inference (Dagan et al., 2006), see also Section 2.2.1. We decided against crowd-sourced annotation of TEA, as our aim was to maximize the consistency of fine-grained entailment decisions. Therefore, TEA was labelled by two annotators[2], where the first round of annotation resulted in just under 20% disagreement across the whole dataset (measured as a raw percentage of cases disagreed upon). The relatively high level of disagreement suggests that even for annotators who (more or less) know what they are looking for, assessing whether an entailment holds between two temporal predications is a very challenging task.

Disagreements in TEA were resolved on a case-by-case basis and all sentence pairs with an initial disagreement have been resolved and included in the dataset. We found that with temporality involved, many entailment pairs became uncertain. For example, in the pair "Airbus is producing the engine" $\vDash$ "Airbus will ship the engine" (labeled **True**), there are readings where perhaps Airbus first puts the engine on an airplane before any shipping occurs, or where a different company does the shipping. We resolved the disagreements stemming from this uncertainty by first discussing which of several possible readings is the strongest, and whether that reading is sufficiently more likely than any other possible reading. Subsequently we discussed whether the strong reading is above the "most likely" common-sense entailment threshold.

3.2.3 Results

As mentioned in Chapter 1, the corresponding publication to this chapter (Kober et al., 2019) also contains experimentation with various embedding methods to see whether they capture temporal entailment. However, the experimental work was carried out by Thomas Kober, so results will be presented here for completeness, and only briefly. The models under examination were `word2vec` (Mikolov et al., 2013), fastText (Bojanowski et al., 2017), ELMo (Peters et al., 2018), BERT (Devlin et al., 2019) and Anchored Packed Trees (APTs) (Weir et al., 2016). APTs are a sparse, high-dimensional semantic vector space model in which the features are derived from typed dependency trees, and which provides a method of distributional semantic composition. We also pre-trained two bi-LSTM encoders (Hochreiter and Schmidhuber, 1997) on the SNLI (Bowman et al., 2015) and the DNC (Poliak et al., 2018) datasets.

3.2.3.1 Preliminary Tasks

We performed a preliminary investigation on the first five models to confirm that they encode basic morphosyntactic information. This provides a sanity check that the mod-

[2] Two of the authors on the paper Thomas Kober and Sander Bijl de Vroe.

els are able to represent this kind of information at all, at least at the more shallow morphosyntactic level rather than the deeper semantic levels subsequently probed using TEA. We tested the models with an auxiliary-verb agreement task, in which the model has to decide whether the inflections of auxiliary verb and main verb match (*is walking* matches, for example, while *had walking* does not). We also investigated whether it was possible to learn consistent translation operations from the lemma embedding to inflected word forms (for example, learning an offset vector for *gerunds* and fastText, such that it can translate from lemmas like *walk* to the respective inflections, like *walking*). Results from those experiments showed that morphosyntactic information relating to tense and aspect is encoded in the different embedding spaces.

3.2.3.2 Performance on TEA

For evaluation we measure precision and recall over varying thresholds and report performance in terms of average precision. We also cast TEA as a binary classification task, and report accuracy and macro-averaged F1-score for the two pre-trained biLSTM models[3]. Table 3.2 shows the average precision scores for the models and the accuracy and F1-scores for the two pre-trained biLSTMs in comparison to a majority class baseline and a baseline predicting the majority class per tense pair. The results show that neither of the models are able to outperform the majority class / tense baseline. This highlights that despite the use of short and simple sentences in the dataset, the latent nature of tense and aspect make TEA a very challenging problem.

Our analysis indicates that although the embedding models appear to extract knowledge about tense and aspect in the contextualization procedure, the signal is not strong enough to reliably draw temporal entailments. A key issue is that these models are primarily governed by distributional similarity, which often does not correlate with deeper semantic concepts such as tense and aspect. See (Kober et al., 2019) for more details.

3.2.4 Challenges in designing TEA

Designing TEA brought with it many lessons surrounding the construction of (temporal) entailment datasets. One issue with entailments between binary predicates is

[3] While the pre-trained models are designed specifically for binary classification of entailment, the others are more general semantic models, so we do not pick a specific threshold and only evaluate them over the whole precision-recall curve.

Model	**Avg. Precision**	**Accuracy**	**F1-Score**
word2vec	0.31	-	-
APT	0.28	-	-
fastText	0.30	-	-
ELMo	0.21	-	-
BERT	0.27	-	-
biLSTM-DNC	0.22	0.58	0.49
biLSTM-SNLI	0.21	0.51	0.47
Maj. class	0.22	0.78	0.44
Maj. class / tense pair	**0.35**	**0.80**	**0.66**

Table 3.2: TEA results. All model results are significantly worse at the $p < 0.01$ level w.r.t. the majority class / tense pair baseline, using a randomized bootstrap test (Efron and Tibshirani, 1994).

that the entailment only really holds for particular arguments. Consider examples (1) to (5):

(1) The Chamber is also selling T-shirts. The Chamber had also shipped T-shirts.

(2) The Chief of Police is prosecuting the thief. The Chief of Police has arrested the thief.

(3) John is publishing a documentary. John has filmed a documentary.

(4) Thomas is eating soup. Thomas has cooked soup.

(5) The Chamber is also selling the T-shirts. The Chamber also designed the T-shirts

All of these examples seem acceptable, and there is a relationship between the predicates, but they are not strictly entailments. If the Chamber sells something, it could be a different company that shipped it. The prosecutor of a thief is not necessarily the one that made the arrest. In the last three examples the entailed predicates seemed to arise from the fact that the object exists at all. All we really know is that *somebody* filmed the documentary, so this entailment can never be captured between binary predicates with the same arguments. This challenge sparked interest in the construction of Multivalent Entailment Graphs, in which we amended the DIH to apply between predicates of different valencies (for example, *kill(A,B)* ⊨ *die(B)*). The work was published under (McKenna et al., 2021) and has not been made part of his thesis.

Another issue is that different annotators have different intuitions about whether an entailed proposition is likely enough to pass the commonsense threshold, especially when dealing with future tense. For example:

(6) The farmer is planting the crops. The farmer will harvest the crops.

(7) Mary booked a ticket to Birmingham. Mary will travel to Birmingham.

In both these cases the entailed event could fail to transpire, although again some lexical relationship should be established between the predicates.

A uniquely temporal problem arose from an interaction between the tenses of events and the ordering of the events' nuclei. Examples (8) and (9) illustrate the point. For instance, the first pair is ambiguous between two readings, a "will-divorce-and-is-already-married" reading on one hand, and a "will-divorce-and-is-yet-to-marry" reading on the other. The issue here is that the ordering between *divorce* and *marry* (*after*) is the same as the ordering between *divorce* and *now* (*after*), so that the ordering between *marry* and *now* cannot be inferred. Without this, we cannot assign a tense to *marry*. Although with pragmatics and world knowledge we might conclude that they are already married, for many predicates the cases are equally likely in the absence of any further context. Part of the issue is also that the entailed statement is too simple - what is really entailed by the premise in example (8) is something like "Bob is married to Jane at some point before the divorce". Often such complex sentences seemed necessary to avoid ambiguity, but in this case we decided to follow previous datasets in having similar sentence structure between the premise and hypothesis.

This leaves us with a conundrum in labeling these cases. One option is to label both as true, which might have happened in a crowd-sourced annotation where annotators don't see all sentence pairs. However, it is probably undesirable to have a dataset that contains these pragmatically contradictory hypotheses. Another option is to label both as false, but being so conservative leaves very few true labels in the dataset, and leaves many predicate pairs for which there should be some entailment disconnected. We mostly chose the third option, in which only one of a pair is labeled as true according to which hypothesis is determined by commonsense knowledge as most likely. In this case, (8) is labeled as True, and (9) as False.

(8) Bob will divorce Jane. Bob has married Jane.

(9) Bob will divorce Jane. Bob will marry Jane.

A similar issue arises from the question of how long a state can be said to hold, combined with a sort of temporal implicature. Consider examples (10) and (11).

(10) Arsenal is winning against Manchester United. Arsenal was playing against Manchester United.

(11) Arsenal is winning against Manchester United. Arsenal is playing against Manchester United.

Technically one could argue that Arsenal was also playing a few minutes ago, or by world knowledge could argue that they were probably playing in the last few months. However, the statement *was playing* doesn't seem to be pragmatically licensed when *is playing* holds true, even though it is logically true. It seems correct to label the first example as false, but this does raise interesting questions about how an entailment task should treat implicatures arising from the hypothesis, and whether a premise and hypothesis should be interpreted as existing in the same discourse, or each treated on their own terms.

Then there is a rather glaring issue: tenses are ambiguous in English. This could be solved in future datasets by adding context in the form of temporal adverbials, but it is still a problem in TEA. For example, *Mary is traveling to London* is ambiguous between a true present tense where travel is ongoing, and a futurate use of the present tense, where travel is still to happen. This is similar to the issue with the ambiguity of the present tense described earlier. We tried to disambiguate habituals in those cases by using definite articles, as in example (12).

(12) James is chewing on the sandwich. James eats the sandwich.

Another problem was generating examples from all tense-aspect combinations. This created issues because the pairs of sentences didn't always share a clear reference time. For example, it is odd to think of a predicate in the past perfect to be entailed by anything other than another past perfect predicate, see example (13). Generating many simple present tenses also resulted in many cases where a habitual reading was available.

(13) James ate the sandwich. James had chewed on the sandwich.

Another source of possible annotation disagreements was a clash between the imperfective paradox and commonsense knowledge. Take example (14). On the one hand, the past progressive *was producing* (as opposed to the perfect *has produced*) should not entail the past simple *owned* because the progressive does not guarantee completion of the event. On the other hand, commonsense knowledge often makes it reasonable to assume that events will be completed - most people would assume that Veloretti would succeed in production.

(14) Veloretti was producing the bicycle. Veloretti owned the bicycle.

A similar disagreement occurs with the semantics of the perfect. As mentioned in Section 2.4.1.4, the consequences of an eventuality expressed in the present perfect are still in force at the utterance time, as opposed to the consequences of eventualities in the simple past (Moens and Steedman, 1988). It is clear then that (15), in the present perfect, should express an entailment relation. However, there is a tension with the commonsense definition of entailment in (16): reasoned semantically, there is no entailment, and yet many other datasets might label this as *most likely* entailed.

(15) George has acquired the house, George owns the house

(16) George acquired the house, George owns the house

Perhaps many of these are also simply issues with the RTE paradigm, which are more evident when we deal with entailments between events in time. It is clear that lexical entailment between events is quite complex, and that these inferences are mostly drawn in the presence of a rich context. Example (16) could leave the annotator with questions that would normally be disambiguated by information in the discourse: Does this house still stand? Is George still alive? Was he acting on behalf of an organization that now owns the house? How far back in the past is the reference time?

The way we design our datasets and our models should take this complexity into account. Another possible solution would be to add more textual context to the examples, avoiding much of the ambiguity mentioned above. This might include adding temporal adverbials, or even a description of the situation, so that a discourse is established and people are more informed of the referents. This would require models to have a strong contextualization mechanism, and could lead to interesting future work in Entailment Graphs where its method of contextualization is extended beyond argument typing.

The commonsense definition stated in Section 3.2.1 acknowledges these ambiguities, but can also make it challenging to settle on a clear distinction between entailment and non-entailment. “Most likely” will mean something different to different annotators in different contexts. Recent work has shown that the disagreement may be inherent in the task (Pavlick and Kwiatkowski, 2019). The multimodality of the label distribution is not resolved by adding more annotators (which would be expected if the disagreements were simply noise). Rather, the distribution remains bimodal or multimodal when more annotators are added, indicating that annotators are gravitating towards separate available readings of the sample. Counterintuitively, they also agreed less as more context around the sample was shown. This may seem discouraging, but if we can find an automatic way of assigning context targeted at the types of events we are dealing with we might arrive at a more well-defined task. In any case, it seems that just tense and aspect are not sufficient phenomena to disambiguate the available readings.

3.3 Tensed Entailment Graphs

We now shift our attention to modeling. An initial project was to add temporality to the graphs by changing the nodes, so that they represent tensed predicates instead of the lemmatized predicates used by Hosseini et al. (2018). This allows us to model the relationship between the different morphosyntactic surface forms of predicates, for example modeling that *is visiting* entails *has arrived* as in Figure 3.2 We refer to this model as Tensed Entailment Graphs.

3.3.1 Model

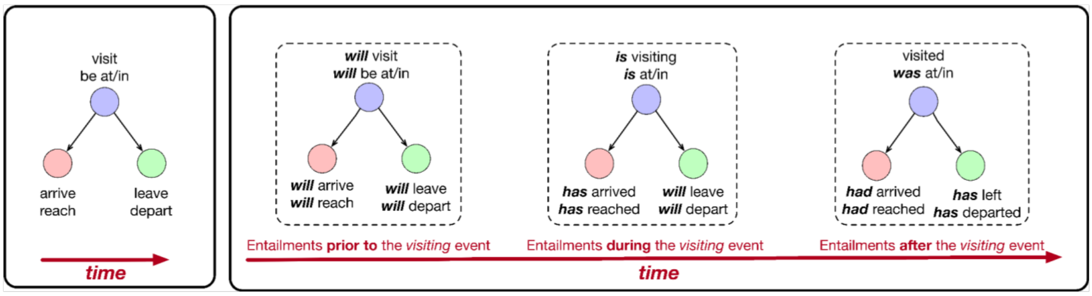


Figure 3.2: A comparison of Entailment Graphs (left) and Tensed Entailment Graphs (right)

3.3.1.1 Definitions

We define temporal entailment as in Section 3.2.1. Our goal is to mine $\mathcal{G}$, a set of typed Entailment Graphs $\mathcal{G}_{t_1,t_2}$, consisting of a set of nodes $P(t_1,t_2) \subseteq \mathcal{P}_{\mathcal{T}}$, where $\mathcal{P}_{\mathcal{T}}$ is a vocabulary of tensed predicates, and a set of directed edges E indicating an entailment relation. $\mathcal{P}_{\mathcal{T}}$ is $\mathcal{P} \times \mathcal{T}$, the cross product of the vocabulary of predicates with the vocabulary of tenses, containing predicates such as *is_visiting(:person,:location)* and *was_visiting(person,location)*. It is here that temporality is introduced into the structure. As before if a node p has an outgoing edge e to a vertex q, then $p \vDash q$. An example of the difference between regular and tensed graphs can be seen in Figure 3.2.

3.3.1.2 Learning Method

We build Tensed Entailment Graphs by modifying the existing entailment graph mining system (Hosseini et al., 2018)[4]. As described in Section 2.3.3, the system first extracts binary relations from text, assigns fine-grained types to the entities and computes directional similarity measures based on the DIH. We use the local graphs, leaving an investigation of the tensed graphs' interaction with globalization to future research.

In order to build TsEGs, it is necessary to add functionality to the relation extraction system. In particular, Hosseini et al. (2018) parse sentences using GraphParser (Reddy et al., 2014), which uses an open-domain syntactic CCG parser (Clark and Curran, 2007) to parse sentences[5]. The syntactic parses are mapped to graphs, from which binary relations with their arguments can be extracted by following all possible paths from entity to entity. Graphparser was originally designed for question answering over Freebase (Bollacker et al., 2008) and didn't require an analysis of tense. To add morphosyntactic tense and arrive at TsEGs we make a number of changes to the relation extraction steps.

We create a lexicon of special cases that takes into account the possible contexts of auxiliary verbs. For instance, the past auxiliary *have* carries the CCG category $(S[dcl]\backslash NP)/(S[pt]\backslash NP)$, while the copular auxiliary *be* for creating progressive aspect requires the category $(S[dcl]\backslash NP)/(S[pg]\backslash NP)$. In addition, unlike Hosseini et al. (2018), we do not lemmatize the verbs so that the extracted relations can distinguish between the simple present and past. We also choose the modal verb *will* as the only modal verb to keep attached to extracted relations, since it almost always marks future

[4] Accessed from https://github.com/mjhosseini/entGraph

[5] Note that the graphs presented in later Chapters uses a different relation extraction system.

tense in English, and other modal verbs (e.g. shall, must, ought to) do not do so as unambiguously.

Through these changes our parser is able to distinguish between 12 morphosyntactic (Deo, 2012) tenses: $\{$*past*, *present*, *future*$\}$ × $\{$*simple*, *perfect*, *progressive*, *perfect progressive*$\}$. For example, where our parser previously returned the relation *sell(person,company)*, it can now return *has_sold(person,company)*. Note that only the seven tenses that use one auxiliary are tested in the evaluation set. For example, the dataset does not investigate the entailments of *will have visited*.

After further learning steps, then, we are able to model entailment edges between different tenses of predicates (as illustrated in Figure 3.2). Whereas before a subset of the <*person*,*object*> graph might look like the graph on the left, adding temporality to the nodes allows for more complex interactions to be captured. As in the subgraph on the right, it can model that *is visiting* entails *has arrived* and *will leave*, while modeling that it does not entail *will arrive* and *has left*.

3.3.2 Experimental Setup

Our methods are evaluated on TEA. As described above, each example contains contextualized predicates in varying tenses, and asks models to make a binary decision regarding whether a premise entails a given hypothesis. The dataset is small, encouraging generalizable models that learn their knowledge from other datasets.

3.3.2.1 Model Implementation

Our graph models are trained using the NEWSSPIKE corpus (Zhang and Weld, 2013), a collection of 500K news articles published over a span of a few months in early 2013. The articles come from parallel news streams, so that there is ample room for the same situations to be expressed in different ways (allowing for more paraphrases and entailments to be found). Furthermore, each of the news stories is annotated with a document creation date, allowing for other the temporal extensions in Chapters 4 and 5.

As mentioned, We use GraphParser (Reddy et al., 2014) for extracting relations. The arguments of the relations are mapped to Wikipedia entities using AIDA-Light (Nguyen et al., 2014). Only relations involving at least one named entity are retained. The argument types of the Wikipedia entities are determined by mapping to their Freebase entry (Bollacker et al., 2008), and then matching this with the first level of the

FIGER-type hierarchy (Ling and Weld, 2012). For example, given a mention of *Google* in a piece of text, we link it to the Google Wikipedia entity, which we map to *organization*. Following experimentation by Hosseini et al. (2018) we calculate directional similarities between predicates using the BInc score.

3.3.2.2 Models

We compare the performance of 5 models. The first three models are based on EG methods, while the others rely on dense representations. Our first model is the entailment graph method as presented by Hosseini et al. (2018). We build TsEGs as presented in Section 3.3.1.

Then, we present a combined model that highlights the strengths of these two methods. When the tenses in two sentences are identical, we use the regular EGs, and if the tenses differ the temporal graphs are used. We refer to this model as EG-TsEG.

As will be further discussed in Section 3.3.3, the nature of the TEA dataset makes it so that the EG model is effectively forced to make mistakes. We therefore provide another baseline, EG-0, to compare TsEG to a model that makes fewer mistakes. This model again uses the regular EG when tenses in the two sentences are identical, but predicts 0 when tenses are different.

Finally, we present results on two neural embedding methods, applied as in (Kober et al., 2019). We evaluate `word2vec` (Mikolov et al., 2013) and BERT (Devlin et al., 2019), two strong embedding models that have performed well over a wide range of NLP tasks. This work was submitted in 2019, precluding the use of stronger modern representations (RoBERTa, GPT-3, etc.). We also wanted a comparison to a classical, non-contextualized dense representation, which is why we included the evaluation of `word2vec`. While BERT can be contextualized by nature of the model, we contextualize `word2vec` by averaging vectors as proposed by Kober et al. (2019). In spite of its symmetric nature, cosine similarity scores performed best for these methods, so we use this as our entailment score.

3.3.3 Results and Discussion

Figure 3.3 shows the precision-recall curves for each of the models on the **TEA** dataset, and Table 3.3 shows the AUC scores at four different recall thresholds. We use different thresholds so that the graph models receive a fair comparison for their respective recall ranges, which differ substantially. We choose thresholds of 0.1, 0.2, 0.25, and 0.75 for

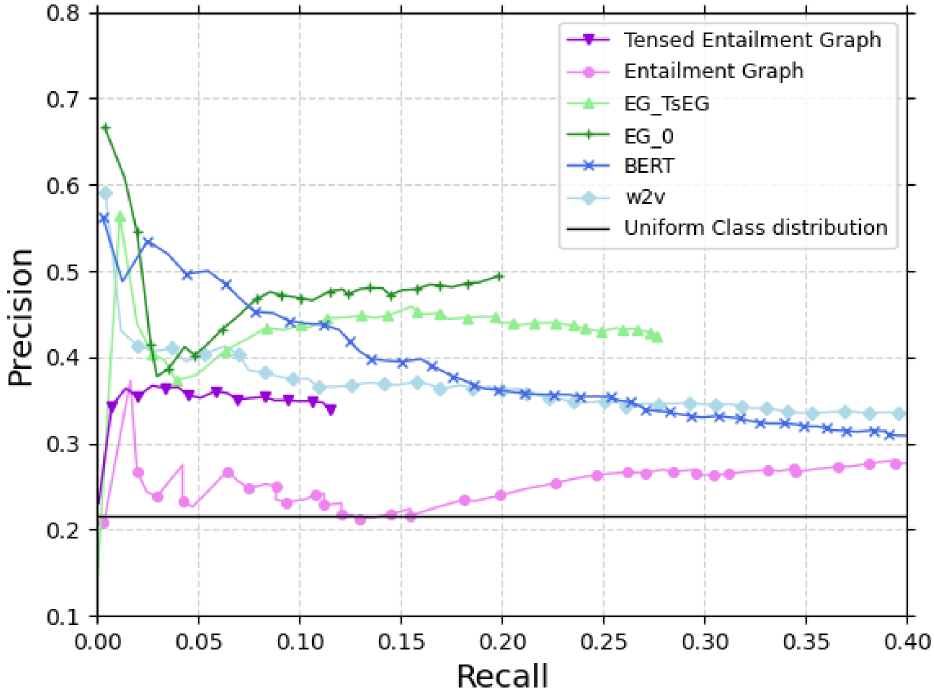


Figure 3.3: Precision-Recall curves for each model

TsEG, EG-0, EG-TsEG and EG, respectively. The distributional models can compute a similarity score between any pair of predicates in their vocabulary, and since their vocabulary covers the Levy predicates they reach a recall of 1. Regular Entailment Graphs, conversely, depend on observing these predicates with the same argument pairs, limiting their maximum recall to 0.78 (note that Figure 3.3 is limited to 0.4 recall). The other models have a much lower maximum recall due to the sparsity of tensed predicates in the training corpus.

In terms of precision all models fair quite poorly. While they do all manage to surpass the uniform class distribution (0.22), they are unable to maintain high levels of precision. Untensed Entailment Graphs have especially low precision, because the structure of the dataset effectively forces them to make mistakes. The dataset focuses on the semantic intricacies of different tense interactions, which conflicts with the model, capable of predicting either 1 (entailment) or 0 (non-entailment) for an entire group of tense pair interactions for any given predicate pair. Thus, even when it understands the predicates are related and has learned the correct directionality, it will predict **True** for all tense interactions of that ordered predicate pair, resulting in much lower precision due to forced false positives. The only entirely correct block of pre-

Model	**AUC scores**				**Max Recall**
	Rec. < 0.1	< 0.2	< 0.25	< 0.75	**Reached**
TsEG	0.035	0.04	0.04	0.04	0.115
EG	0.024	0.048	0.06	0.2	0.783
EG_TsEG	0.042	0.087	**0.109**	0.12	0.276
EG_0	0.047	**0.094**	0.094	0.094	0.198
BERT	**0.049**	0.088	0.106	**0.259**	**1**
w2v	0.041	0.077	0.095	0.258	**1**

Table 3.3: Performance of the distributional and graph-based models for different recall thresholds

dictions it can make is correctly predicting 0s for all the non-entailment pairs due to directionality (e.g. the tensed versions of *own*$\models$*buy*). These true negatives aren't taken into account in calculating precision and recall, however.

The Tensed Entailment Graphs do improve over the regular graphs in terms of precision at the low recall range, which is reflected in their higher AUC score at recall <0.1. Still, their precision is low. Part of the issue is that they are faced with learning a larger number of potential links from the same amount of data. The tensed graphs need to apportion the data of one lemmatized predicate between its 12 morphosyntactically tensed versions, resulting in less reliable distributions that are estimated from a smaller amount of data. Further, the data distributes unevenly over these tenses, magnifying the sparsity for edges between rare tenses, such as *past perfect-past perfect*.

EG-TsEG is the strongest model for its most favorable threshold of 0.25. We expect that its success compared to EG and TsEG in isolation is due to playing to each model's strengths and weaknesses. TsEGs are able to distinguish between different tenses, so they thrive in those cases, while regular EGs suffer less from sparsity, so they perform better than TsEGs when the tenses are identical. The combined model is thus allowed to make the most confident decision for identical tenses and different tenses.

EG-0 focuses on the regular graph's most reliable predictions, simply predicting 0 when the tenses are different. At its most favorable recall range, it achieves the highest AUC of any model. This improvement over EG-TsEG at lower recall indicates that the predictions made by EG are more reliable than those made by TsEG. Still, EG-TsEG achieves higher recall, indicating that the TsEGs correctly retrieve some of the challenging different-tense examples in the dataset.

Premise	**Hypothesis**
True Positive TsEG; False Negative EG	
Liverpool has defeated Arsenal	Liverpool challenged Arsenal
William has acquired the desk	William has the desk
John is visiting London	John was traveling to London
Lee has killed the president	Lee had wounded the president
Kim has the degree	Kim obtained the degree
True Negative TsEG; False Positive EG	
Phil divorced Jane	Phil will marry Jane
Max bought the flat	Max will own the flat

Table 3.4: Examples where TsEG predicts correctly and EG incorrectly on the **TEA** dataset. Additionally, for the true negatives TsEGs still predict true positives on other tense pairs for these predicates, for example finding that *Phil divorced Jane* → *Phil had married Jane* and *Max bought the flat* → *Max has owned the flat*

Table 3.4 shows some example temporal entailments that the TsEGs are able to capture. Although the model is sparse, it is clear that some interesting temporal relations between predicates are being learned from the corpus. For example, *having* something is a consequence of *acquiring* it, and *wounding* is understood to occur before *killing*; these are the types of commonsense knowledge we set out to find. These examples were found in graphs produced with a score threshold corresponding to 10% recall (high precision being more important for these problems).

That distributional models are consistently among the strongest, but also struggle to surpass 0.5 precision. Since they are learned on vast quantities of data and do not depend on argument pair co-occurrences for their learning signal, they are able to produce relatively reliable distributions while covering the entire recall range.

Overall, it is clear that this is a particularly challenging task. With precision values of mostly less than 0.5, all models struggle to recover the deep semantic interactions between tense, aspect and entailment. Note that this may be in part due to possible flaws in the dataset (cf. Section 3.2.4). Still, Tensed Entailment Graphs do manage to correctly predict some of the complex entailment relations in the dataset, of which Table 3.4 shows some examples.

3.3.4 Challenges in Entailment Graph Induction

3.3.4.1 Tensed Entailment Graphs

Although our models are able to capture relations that are difficult to mine from unsupervised data, it is also clear that the evaluation dataset poses a serious challenge. For EG-based methods it is clear that sparsity is a serious issue in spite of the fairly large number of articles in our training corpus. Sparsity issues become even more significant for TsEGs in particular. This is in part due to our choices in the representation — the strategy to separate atemporal nodes into multiple tensed nodes requires even more data, since far more edges need to be learned. When nodes are separated by tense the number of possible edges grows from $n^2 - n$ to $n^2t^2 - nt$ with number of predicates n and number of tenses t.

In terms of learning, it is unclear whether it is principled to apply the Distributional Inclusion Hypothesis to tensed predicates. Here we are assuming that if a predicate in a particular tense entails a predicate in another tense, its context set will be subsumed by the other's context set. In retrospect this seems tenuous because predicates in different tenses mentioned at different times can be used to refer to the same eventuality (*Obama has arrived*, *Obama is arriving*, *Obama will arrive*, all referring to the same arrival, but expressed at different speech times). If the eventualities are mentioned at those different times, it will be challenging to extract the correct tensed entailments because there will be evidence for each of the tenses — *has arrived*, *is arriving* and *will arrive* would all be similarly probable entailments of *is visiting*. Some of the benefits we see may therefore be due to biases in how the eventualities are reported — perhaps *arrive* is often used in the perfect, which would allow the model to predict more easily that its present perfect version is entailed. It is certainly clear that the theoretical grounding of the learning signal interacting with tense warrants further investigation, if other variants of this model are attempted.

3.3.4.2 Entailment Graphs

At this point it is also worth mentioning a few drawbacks to Entailment Graphs more generally. One issue is that their only learning signal comes from the DIH. Again, this states that if a predicate p entails a predicate q, we expect to see the contexts in which p occurs included in the contexts in which q occurs, and not vice versa. In other words, the hyponym's context set will be a subset of the hypernym's context set. This would hold under perfect information: if we somehow observe all possible predications of every occurring eventuality, then the context surrounding entailing predicates p will always also be seen around the entailed predicate q. However, in real world discourse this relationship can be easily skewed. One reason is that news is biased to report interesting occurrences, so that rare eventualities will appear to be entailed more than they should. For example, *taking a breath* will almost never be reported, while *murder* might seem like a relatively common eventuality, making it more difficult to recover the entailments of *taking a breath*.

Relatedly, word frequency in general will interact with these patterns, even when the semantics is constant. For instance, the context sets and frequencies across argument pairs for *somnambulate* and *sleepwalk* will be very different. Since *sleepwalk* will apply to many more argument pairs, *somnambulate* might seem to entail *sleepwalk*, even though they are paraphrastic. This also relates to the frequency effects of cognitively useful *basic-level categories* (Rosch and Mervis, 1975; Rosch et al., 1976) (e.g. *dog* as opposed to *animal*), the names of which are used more often in language than both more general and more specific alternatives. This frequency effect could theoretically make it difficult to recover the correct directionality of entailment between a basic-level category and the more general category it entails. Since the basic-level category applies to more argument pairs (due to reportability and frequency), the model is biased to consider it as a hypernym, even though it should be the hyponym. Although previous research has shown the DIH to be a useful signal in discovering the entailments of nouns, its application to verbs deserves further study, particularly in context with these effects.

Another drawback of Entailment Graphs is their method for contextualization in the representation. Previous approaches (Berant et al., 2011; Hosseini et al., 2018) have achieved this by learning a separate graph for every pair of argument types. We have already discussed that entailment is a highly contextualized problem, with a pair of types failing to supply sufficient context to determine that an entailment will always hold. Temporal entailment contains clear examples, and it is easy to find others within the more generic types such as *organization*, which, within the more fine-grained levels of the FIGER-type scheme, contains types as diverse as *organization/sports_team*, */political_party* and */terrorist_organization*. When *attack*, for instance, is predicated of arguments of these types, the semantics and the available entailments will be very different.

The most generic *thing* type, which forms a bucket for entities that are not recognized by Named Entity Recognition and Linking, poses an extreme version of this problem. For example, the predicate *serves(person,thing)* needs to cover many meanings. By using this fairly coarse typing, the more particular entailments of relations like a *person serves food*, *serves the army*, *serves time* and *serves the ball* cannot be represented in the model — argument types alone are an insufficiently powerful feature to perform WSD. Note that the NER system included in Stanford CoreNLP (Manning et al., 2014) and the NEL system AIDA-Light (Nguyen et al., 2014) (see also Section 2.3.3.1) have not been re-evaluated for this project specifically. They are no longer state of the art, but were sufficiently strong for the research performed by Hosseini et al. (2018).

Improvements can be made by developing the typing systems and the typing scheme, but there is an unavoidable trade-off within this paradigm: any time we increase the specificity of the typing scheme, we increase the number of graphs we need to learn, which introduces sparsity issues and makes each graph significantly harder to induce. On the other hand, a typing scheme with more general categories will reduce the disambiguation power of the representation, allowing us to model fewer entailments.

Finally, a practical challenge with Entailment Graphs is that they require a relatively extensive pipeline to be mined. Even when each component has high accuracy, errors are bound to accumulate. One particularly challenging component (currently not part of the pipeline) is coreference resolution, especially for the news genre. This arises in part because reporters often switch between different nominal references in the newswire style. For example, a single article about Beyoncé in NEWSSPIKE refers to her by name only a handful of times. Besides pronominal references, the article also refers to Beyoncé with the "'Love on Top' songstress", "Queen Bey", "the 31-year old", "the R&B Diva" and "the mother of one". This poses a real issue, since these references are extremely challenging for modern co-reference systems, while they may simultaneously contain essential data for the learning algorithm, which thrives on having varied, co-occurring predicates and argument pairs.

3.4 Conclusion

This chapter has presented work on defining the problem of temporal entailment, which was previously largely unexplored in NLI in spite of its semantic importance. We designed the first evaluation dataset to focus on entailments between predicates of different tenses and aspects. We then alter the relation extraction pipeline to mine unsupervised Tensed Entailment Graphs and compare them against various distributional and graph-based baselines. Although these graphs are able to recover some of the challenging entailment relations in the dataset, they also suffer from sparsity issues. In Section 8.2 we discuss modeling options to alleviate this.

Once these initial conclusions had been reached we decided to shift our attention to other research directions. Rather than introducing temporality into the Entailment Graph representation, we decided instead to use temporal information in the mining procedure to induce more accurate Entailment Graphs. That work is described in Chapter 4.

Chapter 4

Temporality in Entailment Graph Induction

4.1 Introduction

A different kind of possible error in Entailment Graph induction is the prediction of spurious entailments between antonyms, in particular similar but temporally distinct eventualities that occur with the same argument pairs. For example, both the predicates *win against* and *lose against* will apply to sports team argument pairs such as (*Arsenal*, *Manchester*), but will do so at different times. This is likely to mislead the current atemporal methods into incorrectly assigning an entailment relation between those predicates.

Instead of approaching this problem in the representation (as in Chapter 3), we approach it in the learning algorithm, incorporating the temporal location of eventualities, with the aim of removing spurious entailments from the graphs. Temporal information can be used to disentangle these clusters of highly correlated predicates, because although they will share argument pairs, they will never occur at the same time. Consider Figure 4.1, in which Arsenal and Manchester played each other three times in 2019, with three different outcomes, expressed with the predicates *win against*, *lost against* and *tied with*. Previous methods that use the DIH to learn predicate entailments have used a formulation in which the context set refers to argument pairs (Berant et al., 2011; Hosseini et al., 2018)). Therefore they mistakenly take the examples in Figure 4.1 as evidence of entailments or paraphrases between the three antonymous outcome predicates (*win*, *lose*, and *tie*), depending on the distributions found in the data. Our method enriches this context to include time interval information, thereby filtering out combinations that are not temporally near each other. This constitutes a kind of temporal reformulation of the DIH, which should avoid learning that *win against* ⊨ *lost against*, while still learning that *win against* ⊨ *play*.

Arsenal-played, *lost against-Manchester* 1-3 (25/01/19)
Arsenal-played, *won against-Manchester* 2-0 (10/03/19)
Arsenal-played, *tied with-Manchester* 1-1 (30/09/19)

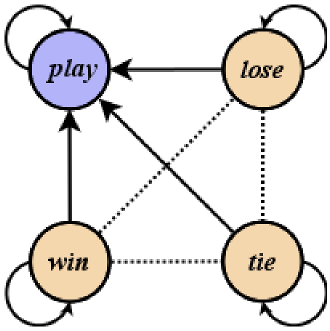


Figure 4.1: Example triples (left) and their resulting entailment/non-entailment graph (right). Arrows indicate entailment, dotted lines indicate non-entailments that previous methods were biased to spuriously learn.

As an initial test domain, we focus on the sports news genre, using extracted triples that involve two sports teams. We design a semi-automatic dataset construction method based on entailments between paraphrase clusters. Applying this to clusters built around the sports predicates *win*, *lose*, *tie* and *play*, we produce a dataset of 1,312 entailment pairs, which we use to evaluate our graphs. Our goal is to recover the structure of the graph in Figure 4.1 in an unsupervised way, separating each of the highly correlated, yet antonymous predicates *win*, *lose* and *tie*, while predicting that they all entail *play*. The news domain is particularly useful for leveraging temporal information, because which each article has a known publication date and temporal expressions are commonly used.

The contributions of this work are: 1) a model for incorporating triple-level time intervals into an Entailment Graph induction procedure, outperforming atemporal models, 2) a manually constructed evaluation dataset of sports domain predicates, and 3) results showing that temporality is indeed a useful signal for entailment learning. To our knowledge this is the first attempt to incorporate temporal information into Entailment Graph induction.

4.2 Method

4.2.1 Relation Extraction

We use a pipeline based on a Combinatory Categorial Grammar (CCG) (Steedman, 2000) parser to extract triples with time intervals. These triples are used to construct typed Entailment Graphs using the unsupervised method of Hosseini et al. (2018), adapted to compare only pairs of triples that are temporally near each other.

We use an updated version of the relation extraction pipeline used in Chapter 3, reimplemented together with Liane Guillou. This replaces Graphparser (Reddy et al., 2014) (which depends on the C&C CCG parser (Clark and Curran, 2007)) with a Rotating CCG parser (Stanojević and Steedman, 2019). The Rotating CCG parser is stronger than the C&C parser (an F1 score of 90.5 compared to 85.2 on the labeled dependencies of CCGbank (Hockenmaier and Steedman, 2007a)), in part because it applies many of the recent advances in neural NLP models, such as ELMo embeddings (Peters et al., 2018) and bi-LSTMs (Graves et al., 2005). It is also more incremental, and thus more cognitively plausible, than previous methods. We later upgraded this system to include modality tagging capabilities (presented in Chapter 6).

The output's form remains as described in Section 2.3.3.1; we extract triples of the form *predicate(arg1,arg2)* (e.g. *win_against(Arsenal, Manchester)*. Triples are extracted from the NEWSSPIKE corpus (Zhang and Weld, 2013) of news articles collected from multiple sources over a period of approximately six weeks.

The Rotating CCG parser (Stanojević and Steedman, 2019) generates a syntactic parse. For example, the sentence "Johnson doubts that Labour will win the election." is parsed as shown in the CCG parse tree in Figure 4.2. Using a method similar to Clark et al. (2002), we then convert the parse into a CCG dependency graph, which can be traversed to produce our relation triples. An example dependency graph is shown in Figure 4.3.

Then, we traverse the dependency graphs starting from verb and preposition nodes, until we reach an argument leaf node. In the example in Figure 4.3, the paths are highlighted in orange. The traversed nodes are used to form (lemmatized) predicate strings, and arguments are classified as either a named entity (extracted by the CoreNLP (Manning et al., 2014) Named Entity Recognizer), or a general entity (all other nouns and noun phrases). Predicate strings may include (non-auxiliary) verbs, verb particles, adjectives, and prepositions. Negation nodes are detected via string match ("not", "n't", and "never"), and are included in the predicate if there is a path between the negation

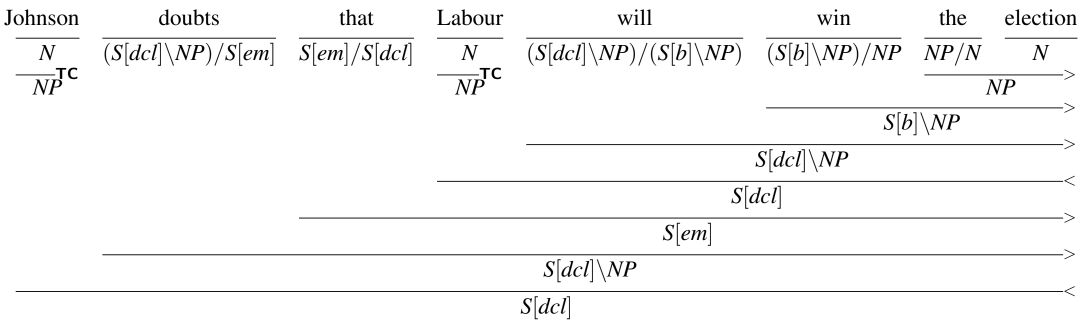


Figure 4.2: A CCG syntactic parse tree of the sentence "Johnson doubts that Labour will win the election."

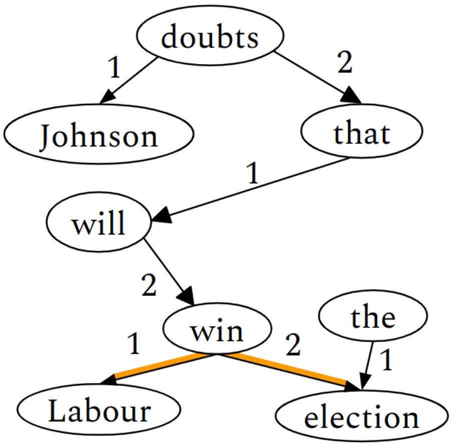


Figure 4.3: A CCG dependency graph converted from a syntactic parse

node and a node in the predicate. We map passive predicates to active ones. Modifiers such as "managed to" as in the example "Arsenal managed to win against Manchester" are also extracted and included in the predicate. As the modifiers may be rather sparse, we extract the relation both with and without the modifier.

We extract and resolve time expressions in the document text using SUTime (Chang and Manning, 2012), available via CoreNLP. If there is a path in the CCG dependency graph between the time expression and a node in the predicate, the triple is assigned a time interval. A number of reasons motivated our choice of SUTime. It was fastest to incorporate, given that the information was already available with CoreNLP, which was already included in our pipeline. It is also a relatively fast system compared to heavier systems like CAEVO (which consists of a ranked sieve of multiple classifiers) and UWTime (which is built using the C&C syntactic CCG parser combined with a temporal semantic parser), see Section 2.4.2.3 for more information. Finally, including just the time expression representation in the parse (rather than using other systems for linking to eventualities as well) allowed the possibility of investigating logical, discourse-level approaches for temporal reference in one of our intended future research directions. However, this system does have the drawback of being older, thus potentially having lower precision and recall than more modern systems. Given that we extracted a time interval for approximately 19% of triples, and that its rule-based approach should yield fairly high precision, we were content using the system.

As in the previous pipeline, arguments are mapped to types by linking to their Freebase (Bollacker et al., 2008) IDs using AIDA-Light (Nguyen et al., 2014), and subsequently mapping these IDs to their fine-grained FIGER types (Ling and Weld, 2012).

We restrict our investigation to the sports domain, which provides many advantages. As a primary reason, the sports domain contains arguments that interact with many other arguments, and do so repeatedly over time (this holds for sports teams in particular, but also players, leagues, cities, stadiums, etc.). Since the DIH depends on argument pair interactions, this provides a rich ground for investigating the learning signal. Intuitively speaking there should also be less missing data, because many news outlets report at least briefly on every game played in a league. Sports data is also easy to mine from the web, and is common in NEWSSPIKE (constituting over a third of the data, see Section 6.5), and in our experience sports entities have reliable Named Entity linking. Due to all these reasons sports provides a strong exploration ground for entailment learning in general.

For the purpose of this project we focus on sports teams. This simplifies the representation to a single type-pair graph, and also provides the straightforward *win-lose-tie* outcome predicate set for investigating the conflation of antonyms and entailment. We limit our data by filtering the total set of output triples, accepting only those involving two arguments of the fine-grained FIGER type *organization/sports_team*. This results in a set of 78,439 triples extracted from 24,147 articles, of which 14,664 triples have time intervals derived from SUTime. In Chapter 5 we apply this method to other argument type pairs in an attempt to generalize our conclusions beyond the sports domain.

4.2.2 Graph Construction

The input to the graph construction step is the set of typed triples with their time intervals, $p(a_1{:}t_1, a_2{:}t_2, [t_s, t_e])$, where t_s and t_e are the start and end of the time interval, are both calendar days, and $t_s \preceq t_e$. An instantiated example is $beat(Arsenal{:}organization, Man\_United{:}organization, [13/5/2013, 13/5/2013])^1$. Since we focus on eventualities that involve two sports teams, the output is a single graph $G_{organization-organization}$, rather than the typical set of graphs for every pair of types. Note that these graphs contain only locally learned entailments.

In the original method for computing local entailment scores, Hosseini et al. (2018) extract a feature vector for each typed predicate (e.g. *play* with type pair organization-organization). As described in Section 2.3.3.2, The argument pairs (e.g. (*Arsenal*, *Man_United*)) are used as the features, and either the count or pointwise mutual information (PMI) between the predicate and the argument pair is the value. These feature vectors are then used to compute local similarity scores.

We extend this method to take into account the time intervals for each of the triples, with the goal of comparing only those eventualities that are temporally near each other. To achieve this, we filter the counts of predicate q according to whether each triple's time interval overlaps with any of p's. In other words, a triple in q is retained if it is temporally close enough to any triple in p. A temporal similarity score s_t is computed by replacing the atemporal feature values $v(p, f)$ by a temporally filtered version $v_t(p, f)$, keeping the computation otherwise identical. Again, $v_t(p, f)$ can be either a temporally filtered count $N_t(p, f)$ or $PMI_t(p, f)$, see also the formulae in Section 2.3.3.2. For example, when we substitute temporally filtered PMI for the value $v(p, f)$ in Weed's Precision we simply get:

$$\text{Temporal Weed's Precision}(p,q) = \frac{\sum_{f \in F_p \cap F_q} PMI_t(p,f)}{\sum_{f \in F_p} PMI_t(p,f)},$$

Algorithm 1 describes the process of filtering counts using time intervals. The process uses a set of edges $\mathcal{E}$ between predicate nodes to store filtered count information. We loop through each argument pair *ap* and get the list of predicates that occur with that argument pair (line 4). Then, for each pair of predicates, we instantiate *edgeObjects* (line 7) between predicates p and q (in both directions), to store the filtered count information. We also retrieve p and q's *timeObjects*, containing a list of the time intervals at which the predicate and argument pair co-occurred (lines 8–9). For each pair of time intervals we compute whether there is an overlap (lines 12–19). The filtered count is the total number of triples in predicate p that temporally overlap with any triple in predicate q. The count is stored in the *edgeObject* $edge_{p,q}$. Once all counts have been collected, they are used to compute the similarity measures.

[1] Since we are dealing with sports events, the start point and end point of the interval will normally be the same day.

Play	Arsenal	Manchester	18/1/2021
Beat	Arsenal	Manchester	18/1/2021
Play	Arsenal	Manchester	12/2/2021
Lose to	Arsenal	Manchester	12/2/2021

Table 4.1: The extractions resulting from four descriptions of two sports matches between the same teams.

Edge	Regular Count	Filtered Count
$play \vDash beat$	2	1
$beat \vDash play$	1	1
$play \vDash lose$	2	1
$lose \vDash play$	1	1
$beat \vDash lose$	1	0
$lose \vDash beat$	1	0

Table 4.2: Comparing regular and temporally filtered counts for each edge.

Since our loop considers the argument pair, predicate p and predicate q, generalizing the PMI score to our case would mean precomputing and storing each Conditional PMI(p, ap, q). However, this is computationally expensive (if not infeasible) given the existing graph construction framework, so we instead opt to scale the original PMI scores using the filtered counts. We apply two strategies: 1) *Ratio:* the temporally filtered PMI_t = $PMI \cdot (c_t/c)$, i.e. the original PMI multiplied by the ratio of filtered counts (c_t) to regular counts (c), and 2) *Binary:* PMI_t = PMI if c_t>0; otherwise 0 — using the original PMI score if any of the triples in predicate p overlap with any triple in predicate q, otherwise setting the score to zero. The intuition in both cases is that the PMI score will be reduced if the predicates are temporally separate.

For example, suppose two football matches are held between Arsenal and Manchester, one described as happening on 8th January 2021 where "Arsenal *played* and *beat* Manchester.", and another on 12th February 2021 where "Arsenal *played* and *lost to* Manchester" (see Table 4.1). The algorithm computes a *filtered count* for each argument pair for the pair *p*-*q*: the total number of eventualities of predicate p with a time interval that temporally overlaps with the time interval of any eventuality of predicate q, and vice versa. In this case the filtered count for $play \vDash beat$ = 1 and $play \vDash lose\_to$

= 1 as there is a temporal overlap for the *play* and *beat* events in the first match and the *play* and *lose to* events in the second. Crucially, the filtered count for *beat* ⊨ *lose_to* becomes zero as there is no temporal overlap between the *beat* and *lose to* events. See Figure 4.2 for an illustrated example. In this way, the values of temporally separate predicate pairs are lowered, leading to discounted similarity scores. We use the filtered counts to compute the temporal similarity measures described in Section 5.2.4. The regular counts are used to compute their (standard) atemporal counterparts.

We consider three possible sources of time intervals: 1) the resolved time expressions extracted from raw text using SUTime, 2) the document creation date (provided as metadata in the NEWSSPIKE corpus), and 3) a combination of the two – using resolved time expressions where these are available, and backing off to the document creation date where they are not. The intuition behind using time expressions extracted from the article text is that these ought to more accurately pinpoint the time interval of the eventualities. However, as such expressions may be sparse, we also investigate the use of the document creation date, under the assumption that sports news is likely to be reported very close the day of the eventuality.

We also consider a temporal window to extend the time intervals by N days either side. This would mitigate the problem of sports events being reported several days later, especially when we fall back to the document creation date. For sports matches we would expect to see a benefit in using a small window of a few days, and a detrimental effect as that window grows increasingly larger. Specifically, we expect that larger windows would render temporal information useless, preventing our model from being able to distinguish between two different matches involving the same pair of teams. Time interval source and window size are parameters that we experiment with in Section 4.4.1.

4.3 Evaluation

4.3.1 Dataset Construction

We propose a semi-automatic method to construct a small evaluation dataset based on manually constructed paraphrase clusters. We start with a small set of predicates for which we know the entailment pattern, in our case $\{win, play, lose$ and $tie\}$. We restrict the dataset to include only those triples that involve two sports teams, by filtering on the fine-grained FIGER (Ling and Weld, 2012) type *organization/sports_team*. We then

Algorithm 1 Temporal filtering in local graph computation

```
procedure TemporalFilter($argument\_pairs, predicates$)
    $\mathcal{E} \leftarrow initializeAllEdgeObjects(predicates)$        ▷ Initialize set of edges
    for ap in argument_pairs do
        $predicates_{ap} \leftarrow getPredicatesForEntityPair(predicates, ap)$
        for $p \leftarrow 0$ to $length(predicates_{ap})$ do
            for $q \leftarrow p + 1$ to $length(predicates_{ap})$ do
                $edge_{p,q}, edge_{q,p} \leftarrow getEdgeObjects(\mathcal{E}, p, q)$
                $time\_objects_{ap,p} \leftarrow getTimeObjects(ap, p)$
                $time\_objects_{ap,q} \leftarrow getTimeObjects(ap, q)$
                $overlap_p \leftarrow initializeVectorOfZeros(length(time\_objects_{ap,p}))$
                $overlap_q \leftarrow initializeVectorOfZeros(length(time\_objects_{ap,q}))$
                for $i \leftarrow 0$ to $length(time\_objects_{ap,p})$ do
                    for $j \leftarrow 0$ to $length(time\_objects_{ap,q})$ do
                        if $intervalOverlap(time\_objects_{ap,p}[i], time\_objects_{ap,q}[j]) = 1$ then
                            $overlap_p.set(i, 1)$
                            $overlap_q.set(j, 1)$
                        end if
                    end for
                end for
                $edge_{p,q}.addTCounts(sum(overlap_p))$        ▷ Rels $p$ that temporally overlap with any $q$
                $edge_{q,p}.addTCounts(sum(overlap_q))$
            end for
            $\mathcal{E}.update(edge_{p,q}, edge_{q,p})$
        end for
    end for
    return $\mathcal{E}$
end procedure
```

order the predicates by their frequency in the corpus, and manually select paraphrases of our small set with a count of at least 20 (the 235 most frequent predicates). This results in four clusters of paraphrases, with sizes of 26, 8, 3 and 5 respectively for *win*, *lose*, *tie* and *play*. We then automatically generate entailment pairs (1,312 in total), labeling them according to the pattern in Table 4.3, with premises in the rows and hypotheses in the columns.

This results in a LIiC-style dataset of 1,312 pairs with *entailment* and *non-entailment* labels. The task is fairly challenging compared to other LIiC datasets because its non-entailments are particularly challenging. Models are expected both to predict the correct directionality of entailments (otherwise penalized with the directional 0 category) and not to predict spurious entailments between antonyms. Note that this task has similarities to antonym detection (see Section 2.2.3.4), and the public version of the dataset contains the full label set so that it can be adapted for this purpose.

We include the *paraphrase* category for completeness, although we are more interested in the effect of separating the antonymous, temporally disjoint sports match outcomes. The paraphrase category contains predicates of varying gradation, such as *crush* suggesting a strong victory or *eliminate* indicating that a team is knocked out of a tournament. We wish to avoid specific predicates such as *eliminate* entailing non-specific predicates like *win against*. To avoid this issue we manually annotated the predicates for specificity, and for the *paraphrase* entailments subset we only generate pairs for non-specific predicates. More generally, a set of paraphrase clusters with a total of n predicates yields $n^2 - n$ pairs (not taking into account the paraphrase subsets reduction)[2].

4.3.2 Similarity Measures

We compute both symmetric and directional similarity measures to learn entailments, making use of the temporally filtered counts and PMI scores described in Section 4.2.2. Specifically, we adapted Lin's similarity measure (Lin, 1998), Weeds' precision and recall measures (Weeds and Weir, 2003), and BInc (Szpektor and Dagan, 2008). This leads to a number of adaptations. **Temporal count-based measures** using the temporally filtered counts: Weeds' precision, recall, and similarity (harmonic average of precision and recall); Lin's similarity; BInc using Weed's precision and count-based

[2] The subtracted term comes from duplicate pairs like *defeat-defeat*

(a) 1 = entailment, 0 = non-entailment. Blue = base (directional entailments, and non-entailments from temporally disjoint antonyms), orange = directional non-entailment, green = paraphrases

	win	lose	tie	play
win	1	0	0	1
lose	0	1	0	1
tie	0	0	1	1
play	0	0	0	1

(b) Examples from the evaluation dataset

Category	Examples	Size
directional 1	defeat ⊨ vs crush ⊨ face	272
antonym 0	beat ⊨ fall to outscore ⊨ lose	446
directional 0	play ⊨ win go against ⊨ tie	272
paraphrase 1	top ⊨ knock off defeat ⊨ outplay	322

Table 4.3: Entailment pairs evaluation dataset

Lin's similarity. **Temporal PMI-based measures** using both Ratio and Binary PMI: Weeds' precision, recall, and similarity; Lin's similarity; BInc using Weed's PMI precision and Lin's similarity. **Temporal hybrid BInc measures:** Computed using count-based Weeds' precision and PMI-based Lin's similarity, using the temporally filtered counts, Ratio and Binary PMI. We also ensure that for every temporal measure, its atemporal counterpart is also included, and we include cosine similarity as a symmetric baseline[3].

4.4 Experiments, Results and Analysis

4.4.1 Experimental Settings

As described in Section 4.2.1 we extract all possible triples from the NEWSSPIKE corpus and map their arguments to types using the FIGER hierarchy. We construct a typed Entailment Graph for the *organization-organization* type pair using the subset of these triples where both arguments are sports teams. We compute entailment scores using the set of 29 similarity measures described in Section 4.3.2. We highlight results for the strongest measures: BInc based on counts and PMI values and Weed's precision, along with their temporal counterparts.

We experimented with different values for the time information source and temporal window described in Section 4.2.2. We constructed typed Entailment Graphs using

[3] In total, we experiment with 29 similarity measures. However, to focus the discussion we present only the strongest scores. The hybrid scores did not exhibit interesting differences.

only time expressions (timexOnly), only the document creation date (docDateOnly), and using time expressions where available, otherwise backing off to the document creation date (*timexAndDocDate*). For each of the time interval sources, we also applied windows of 1, 2, 3, 4, 5, 6, 7, 30 and 3,650 days, as well as *no window*.

We used the evaluation dataset described in Section 4.3.1 and Table 4.3 to evaluate entailments captured under each of these experimental settings. We evaluate on three different configurations of the dataset: **Base** (directional 1 and antonym 0), **Directional** (directional 1 and directional 0), and **All** (directional 1, antonym 0, directional 0 and paraphrase 1).

Note that the results are different to those reported by Guillou et al. (2020) due to updated relation extraction, correction of an error in the method for applying temporal windows, and thresholds as applied by Hosseini et al. (2018) (predForArgPair[4] and argPairForPred = 4, edgeThreshold[5] = 0.01). We consider these the definitive results. In the timexOnly setting we use an edgeThreshold of 0.00, since this sparse model benefits from using the full set of edges.

4.4.2 Temporality and Temporal Information Source

Table 4.4 contains area under the curve (AUC) scores for a range of temporal (T.) and atemporal similarity measures, for each of the three temporal information sources. To evaluate the similarity measures fairly we calculate AUC under a recall threshold (a recall of 0.75 is reached by all non-timexOnly measures).

Results confirm that temporality is a useful signal. Firstly, *timexAndDocDate* is the strongest temporal information source — it is most effective to use the temporal expressions when available and use the document date otherwise. Comparing the temporal and atemporal scores for that information source, we find that the temporal scores are consistently higher. The precision-recall curves in Figure 4.4 also illustrate this. We tested statistical significance of the differences between temporal and atemporal scores on the *timexAndDocDate* setting using bootstrap resampling (10K samples) (Efron and Tibshirani, 1985; Koehn, 2004). The p-values are .065, .091 and .074 for BInc (PMI), BInc (Count) and Weed's Precision respectively, so that for all these scores the difference is significant at the < 0.1 level.

Example subgraphs from the atemporal BInc and temporal Ratio BInc Entailment

[4] A hyperparameter for reducing noise in the input data. A value of 4 means an argument pair is kept in the data if it co-occurs with at least 4 predicates (and vice versa for argPairForPred).

[5] Another noise reduction hyperparameter that removes all edges in the graph below the threshold.

Similarity measure	*timexAndDocDate*	*docDateOnly*	*timexOnly*	
	rec < 0.75	rec < 0.75	rec < 0.75	rec < 0.1
T. BInc (Count)	0.481	0.473	0.112	0.082
BInc (Count)	0.462	0.462	**0.460**	0.069
T. BInc (Ratio PMI)	**0.495**	**0.493**	0.119	**0.092**
T. BInc (Binary PMI)	0.491	0.489	0.116	0.089
BInc	0.471	0.471	0.459	0.072
T. Weed's Pr (Count)	0.455	0.449	0.103	0.074
Weed's Pr (Count)	0.440	0.440	0.434	0.061

Table 4.4: Temporal information source: AUC scores for the base evaluation dataset, and a temporal window size of 5 days

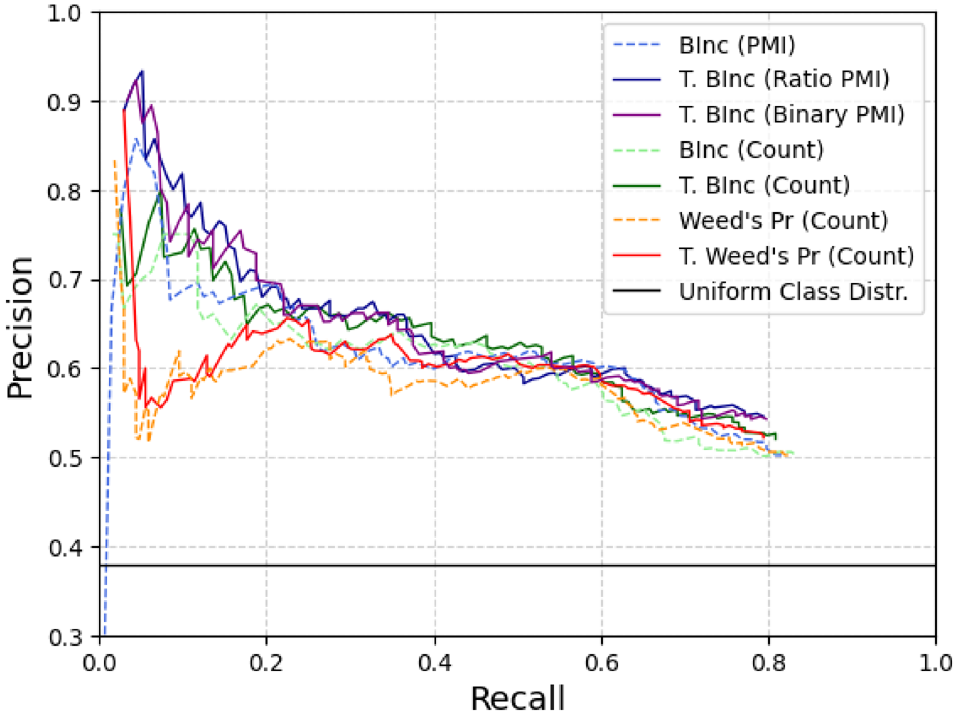


Figure 4.4: Results on the base evaluation dataset with the *timexAndDocDate* time information source and a temporal window size of 5 days. Dotted lines are atemporal scores, complete lines of similar color are the temporal counterparts.

Graphs are shown in Figure 4.5. Entailment scores between graphs are not necessarily directly comparable, since a graph trained with sparser data or fewer comparisons might exhibit lower scores across the board. Therefore we also present the rank of the entailment edge within the evaluation dataset (in bold next to the score) — entailments should have a low rank, whereas non-entailments should have a high rank. We can see that in these example subgraphs the temporal signal has worked as intended. The edges going into *play* have a lower rank in the temporal case, whereas the edges between the outcome predicates *win against* and *lose to* both have higher ranks in the temporal case. We can also observe that entailment scores in the atemporal graph tend to be higher overall.

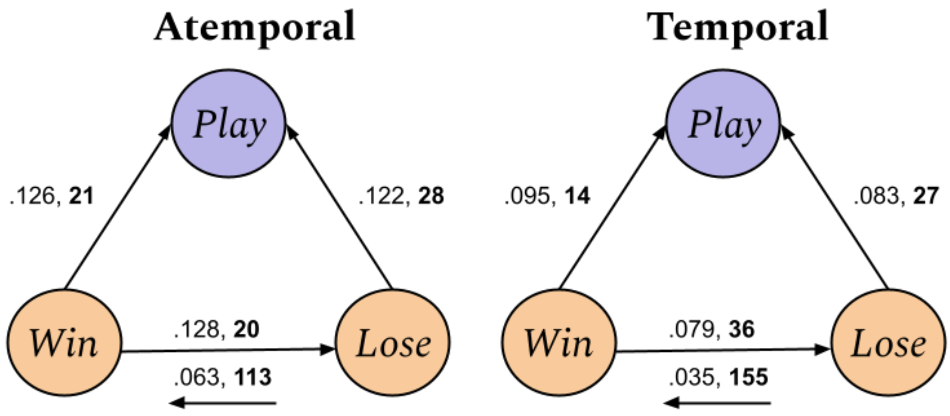


Figure 4.5: Edge values and dataset rank on *base*, for the *play* - *win against* - *lose to* subgraph. On the left is the BInc score graph, and on the right is the Temporal Ratio BInc score graph trained with *timexAndDocDate* information and a 5 day window.

Although temporality overall provides a useful effect, these results raise the question of whether the time expressions provided by SUTime are a benefit to the system as a whole. We can evaluate this in two ways, firstly through the comparison of the *timexAndDocDate* and *docDateOnly* sources, and secondly through the comparison of the temporal and atemporal scores within the *timexOnly* source.

The difference in AUC derived from adding temporal expressions alongside temporal document date data (*timexAndDocDate* compared to *docDateOnly*) is positive and consistent across scores, but very slight. The largest absolute difference is only 0.008 for the T. Binc (Count) score, while the strongest score, T. BInc (Ratio PMI), gains only 0.002 AUC. Thus while time expressions improve the score, their contribution is not essential to the strongest system. The result indicates that it would be worth improving the precision (and recall) of the temporal understanding systems, discussed in more detail in Sections 4.5 and 8.3.5. We discuss in Section 4.5 how this slight difference may be due to the noise of document dates drowning out the signal of time expressions.

The *timexOnly* experiments paint a revealing and more promising picture. We note first that the *timexOnly* source reaches much lower recall, most likely due to having access to a sparser set of triples (with only 19% of triples linked to a time expression in the text). We therefore also include AUC scores below a 0.1 recall threshold (Table 4.4). Performance of timexOnly temporal scores evaluated under 0.75 recall is significantly worse than atemporal scores, but the results flip when we focus on the low recall range. Here we find that every temporal measure outperforms their atemporal counterpart (for example, the strongest score is T. BInc (Ratio PMI), exceeding the BInc state-of-the-art). This result indicates that when focused on just those triples for which we have accurate time interval information, it is more useful to include temporal expression information than to exclude it, which is promising for temporal expressions as a learning signal. The result is also visualized in Figure 4.6 — the temporal measures are able to start at a much higher precision, but reach very low recall.

With the current temporal parsing system SUTime, the benefit of temporal expressions is only very slight to the system overall, and temporal expressions alone are clearly not enough to be practically useful. However, we do believe these results show that temporal expressions show clear potential in the future. Higher recall temporal parsing systems will certainly be necessary to extract more specific information from the text, and alongside further development of the algorithm they may yet prove their value. Note that this discussion pertains only to time expressions specifically — temporality in general (as with the document date information) is still clearly useful.

4.4.3 Temporal Window Size

Figures 4.7 and 4.8 show the performance of graphs with different window sizes. Every point represents the AUC for a precision-recall curve with a particular window size. The graphs presented here use the same results as in (Guillou et al., 2020). Since they are particularly expensive to compute (with each data point requiring a separate Entailment Graph) we did not update these results with the newer models.

In both figures we can see that there is a sharp improvement in AUC score for all

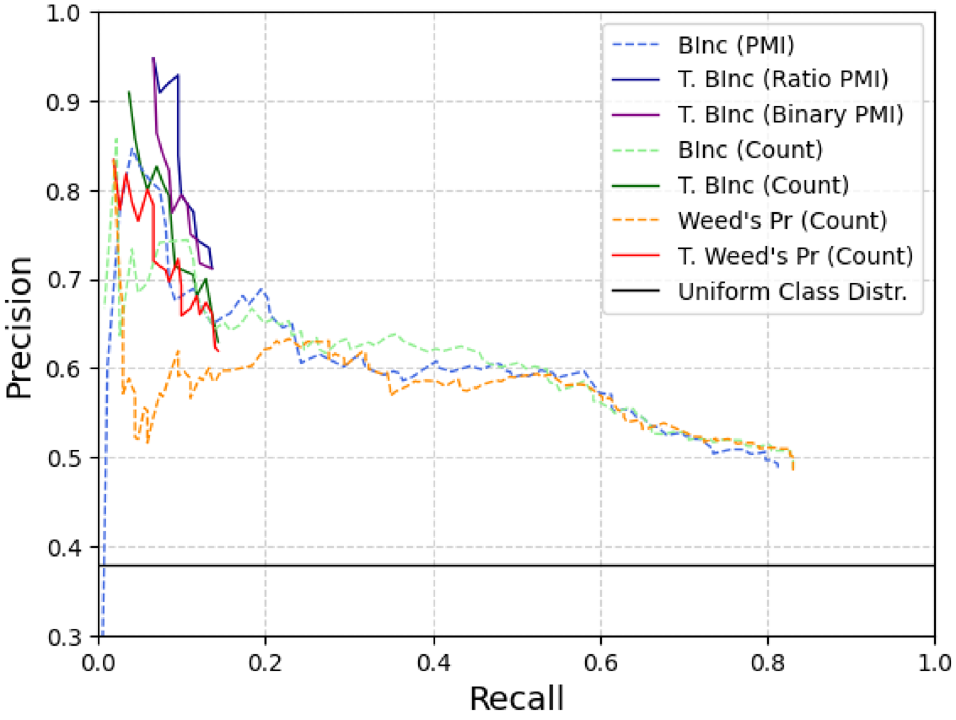


Figure 4.6: Results on the base evaluation dataset with the timexOnly time information source, temporal window size of 5 days.

of the temporally-informed similarity measures when a window of one day is applied[6]. This is likely due to data sparsity and because sports articles report on the same event on different days. In some cases (e.g. T. BInc (PMI) in Figure 4.7 and all scores in the *timexOnly* case in Figure 4.8) the temporal score then rises above the atemporal equivalent represented by the horizontal lines, before returning to the atemporal baseline with very large window sizes[7]. The aim in future research can be seen as maximizing the size of this gap.

For the *timexAndDocDate* time source (Figure 4.7) there are two peaks for most similarity measures, at 5 days and at 10 days. This may be due to different window sizes being effective for different sports: if two teams play each other in consecutive weeks with possibly differing outcomes then the window side should be shorter, but if teams only play each other once a month then it is safe to increase the window size. For this class of predicates a window size of 5 seems suitable, as it avoids conflating games that happen on consecutive weekends, while giving some leeway. We explore

[6] With no window, the temporally informed similarity measures perform poorly (between 0.24 and 0.26)

[7] At window size $w = \infty$, every temporal score $s_t = s$

the possibility of using a dynamic window in Chapter 5.

For the *timexOnly* source described in Figure 4.8, a positive window size almost always allows the temporal score to surpass their atemporal counterpart. Here we see a peak around 4-5 days, and a similar second peak at ten days, with the PMI-based BInc performing best. This highlights that using a window around the temporal expression time is necessary for good performance, and that the temporal expressions can be a valuable learning signal for predicate entailment.

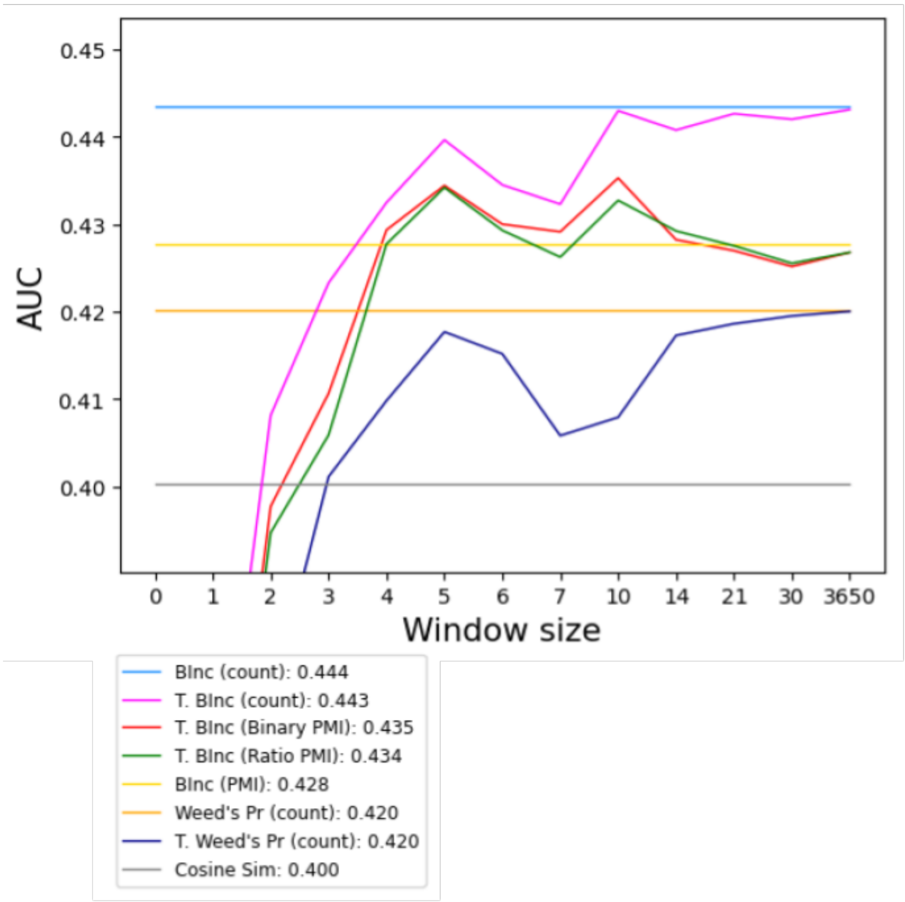


Figure 4.7: Effects of window size for the *timexAndDocDate* temporal information source

4.4.4 Data Subsets

Our main interest is in performance on the *base* dataset, but for completeness and comparison to previous work, which has investigated directionality, we also evaluate on the *directional* and *all* sets (see Table 4.5).

To investigate the challenge of directionality in entailment we consider the set of

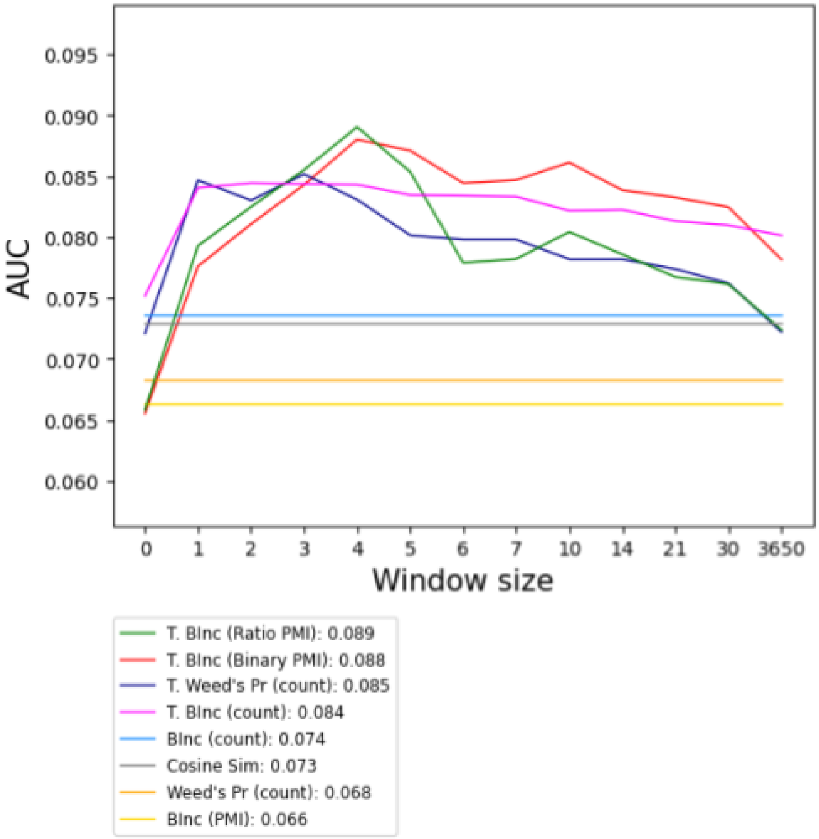


Figure 4.8: Effects of window size for the *timexOnly* temporal information source

entailments and their reverse, e.g. *win* ⊨ *play* and *play* ⊨ *win*, "directional 1" and "directional 0" in Table 4.3(b)). We find that in general the temporal similarity measures still perform well, although the difference is smaller than for Base. Every score outperforms its atemporal counterpart, and *T. Weeds' precision*, the only purely directional measure in the set, is the strongest score on the *directional* subset.

We also evaluate on the complete dataset (*all*), which includes paraphrases ("paraphrase 1" in Table 4.3(b)). Here we find that *Weeds' precision* is the best measure by a small margin. Its performance may again be due to correctly identifying the directional entailments in the dataset. *BInc* also performs reasonably well on this subset, showing that atemporal measures remain competitive when multiple phenomena are tested. The temporal measures do not exhibit the same improvement as on the other data subsets, likely because the temporal method will reduce the available evidence for paraphrases. In this particular distribution of entailment categories, it seems that that benefits for antonyms and directionality are approximately balanced to the costs

Similarity measure	**Base**	**Dir**	**All**
T. BInc (Count)	0.481	0.430	0.426
BInc (Count)	0.462	0.419	0.424
T. BInc (Ratio PMI)	**0.495**	0.437	0.426
T. BInc (Binary PMI)	0.491	0.435	0.425
BInc	0.471	0.432	0.427
T. Weed's Pr (Count)	0.455	**0.472**	0.429
Weed's Pr (Count)	0.440	0.460	**0.431**

Table 4.5: AUC scores for different subsets of the evaluation dataset. Temporal information source is *timexAndDocDate*, with a temporal window size of 5 days, with a recall threshold of 0.75

for paraphrases. Future research may investigate ways of mitigating this cost while retaining the benefits for antonymy and directional entailment.

4.5 Discussion and Future Work

We take these results to indicate that temporality is a useful signal for accurately learning entailments and non-entailments between predicates, which can be taken into account in future research and applications. The temporal measures are stronger than atemporal ones with the strongest temporal information source *timexAndDocDate*. *T. BInc Ratio/Binary PMI* outperform *BInc*, the state-of-the-art atemporal similarity measure for predicate entailment employed by Berant et al. (2011) and Hosseini et al. (2018). It is also promising that scores with the *timexOnly* information source achieve much higher precision for the short recall range they cover. Furthermore, the benefits of temporality extend to two challenging task setups: distinguishing between directional entailments and antonyms, and between directional entailments and non-entailments. Still, they are not necessarily helpful for all types of entailment, particularly paraphrase.

One reason that the *timexOnly* results do not extend into higher recall ranges could be the quality of SUTime, which takes a rudimentary rule-based approach. Currently, we also only connect time expressions to eventualities through CCG dependencies in the same sentence, performing no further reasoning to deduce more event times at the paragraph or document level. A stronger temporal resolution system could thus greatly improve the *timexOnly* performance and might allow temporal measures informed by the *timexAndDocDate* soure to reach even higher AUC scores. Relatedly, the NEWSSPIKE corpus covers a period of only approximately six weeks, and the outcomes of two matches between two teams within this period may be similar (with few changes to the teams management, players, etc.). In future work it should be useful to move to a corpus covering a longer time period, as well as a larger corpus in general, for which we would expect to observe a greater effect. We discuss related future work strategies in Chapter 8, including a number of options for gathering higher quality relation extraction data.

Whilst we can observe a positive effect when using temporal information, the effect is modest. Upon closer inspection we found that this was due to relatively few triples being filtered. An analysis of a subset of sentences revealed that triples were being extracted spuriously due to various linguistic phenomena. Issues were caused by conditionals (e.g. "if Arsenal win"), modals ("I still expect Arsenal…"), incorrect future predictions ("Arsenal will win") and counterfactuals ("had Arsenal won, ...")[8]. These types of predictions appear to be especially common in the sports domain. Another issue arose due to an incorrect application of passive to active conversion (*lost_to(Arsenal, Manchester)* from "Manchester lost to Arsenal") resulting from incorrect verb feature labels in the CCG parses. Addressing these issues should lead to a larger effect from using temporal information, because it would reduce overlaps and allow more filtering.

A similar issue was that the data sometimes contained underspecified time expressions (e.g. "They played last month"). SUTime provides the full time interval for these expressions (e.g. $[01/01/12, 31/01/12]$), which again may lead to spurious overlaps. We incorporated a simple fix to this issue after the publication of (Guillou et al., 2020), and the results presented here are updated accordingly. Our quick solution is simply to exclude any triples longer than a week, but there are certainly further potential gains here in future research. For one, it would be possible to distinguish between eventualities that actually have a longer duration (e.g. $plays\_for(:person, :sports\_team)$) and eventualities of a shorter duration that simply have an underspecified time expression. The duration prediction strategy used in Chapter 5 could serve this purpose.

Later investigations (beyond (Guillou et al., 2020)) led to the discovery of a potential drawback of the approach described here. One potentially undesirable feature

[8] This observation motivated the work in Chapters 6 and Chapters 7.

is that the more eventualities there are, the more challenging it becomes to filter out eventualities. This is a result of conditions we set on temporal filtering. In the algorithm, a particular eventuality time i of $time\_objects_{ap,p}$ (i.e. a single count of a triple $v(p,ap)$) is filtered out when it does not temporally overlap with *any* time interval of $time\_objects_{ap,q}$. This means that as the length of $time\_objects_{ap,q}$ increases, each time interval $time\_objects_{ap,p}[i]$ needs to pass more comparisons in order to be filtered out. Simply put, the more data there is, the higher the chance that there are overlaps. This feature is not necessarily incorrect, but it does mean the system's robustness to noise is limited. With a noisy pipeline and a large amount of data, the chance greatly increases that at least one spurious overlap occurs for a pair $(time\_objects_{ap,p}[i], time\_objects_{ap,q})$ that should be separated (for instance, overlapping due to a modalized predication or incorrect time interval allocation).

The solution may involve a metric of temporal overlap between a time interval and a vector of time intervals that is less dependent upon the vector's length. One simple approach could be to base the condition on a tuneable percentage of overlaps. For example, keep the count if $time\_objects_{ap,p}[i]$ overlaps with 10% or more of the elements of $time\_objects_{ap,q}$ (instead of just 1). These options are worth exploring in the future.

Incidentally, the effect of more data leading to less filtering may help explain a counterintuitive result. We might expect the timexOnly condition to have the most accurate temporal data, which might lead us to expect a significantly stronger temporal effect in *timexAndDocDate* than in *docDateOnly*. However, the difference between these conditions is small. It is possible, then, that this small effect is due to the amount of data in the *timexAndDocDate* condition (and the noise that accrues with it). Perhaps the more precise temporal signal comes through more strongly in the relatively sparse set of *timexOnly* triples, whereas this signal gets drowned out in the volume (and added *DocDate* noise) of the *timexAndDocDate* triples. Still, it is also possible that the temporal expressions are similarly noisy to the document dates, but the result certainly warrants further investigation.

An interesting potential feature of our algorithm is that it could add a directional signal using temporality, allowing us to better distinguish between $beat \vDash play$ and $play \nvDash beat$ in the count-based case. Unfortunately this opportunity was missed in the original implementation. The effect is easiest to understand by example.

Imagine we again have a toy dataset resulting in the counts in Table 4.6, like the one in Table 4.1. This time we also highlight the crucial role of normalization: in Weed's

Edge	N	$\frac{N}{N}$	N_t	$\frac{N_t}{N_t}$	$\frac{N_t}{N}$
play $\models$ *beat*	2	$\frac{2}{2}$	1	$\frac{1}{1}$	$\frac{1}{2}$
beat $\models$ *play*	1	$\frac{1}{1}$	1	$\frac{1}{1}$	$\frac{1}{1}$
play $\models$ *lose*	2	$\frac{2}{2}$	1	$\frac{1}{1}$	$\frac{1}{2}$
lose $\models$ *play*	1	$\frac{1}{1}$	1	$\frac{1}{1}$	$\frac{1}{1}$
beat $\models$ *lose*	1	$\frac{1}{1}$	0	$\frac{0}{0}$	$\frac{0}{0}$
lose $\models$ *beat*	1	$\frac{1}{1}$	0	$\frac{0}{0}$	$\frac{0}{0}$

Table 4.6: Comparing regular (N) and temporally filtered counts (N_t) for each edge. This time we include regular normalized counts ($\frac{N}{N}$), temporal counts with temporal normalization ($\frac{N_t}{N_t}$) and temporal counts with atemporal normalization ($\frac{N_t}{N}$)

precision, the feature value is added both to the numerator and the denominator[9].

When moving from N to N_t it may initially seem like we are are inducing *play* and *beat* to be more paraphrastic (since for N the counts are different and for N_t they are the same). However, what matters to the entailment score is not the raw count, but the count and its normalization. Here we observe that the behavior is actually similar between the temporal and atemporal score: the edge for *play* $\models$ *beat* and *beat* $\models$ *play* receive the same normalized information. We correctly separate the antonyms ($1 \rightarrow 0$) and leave the other behavior relatively constant.

However, the final column describing temporal counts with atemporal normalization reveals that we missed an important choice in normalization. We get the desired difference between the *play* $\models$ *beat* and *beat* $\models$ *play* edges when we normalize the temporal counts with the atemporal counts. In this toy dataset, that would lead to the desirable effect of increasing the average score of *beat* $\models$ *play* towards 1, while driving the *play* $\models$ *beat* score to a lower value of $\frac{1}{2}$. A more effective formula might then be:

$$\text{Temporal Weed's Precision}(p,q) = \frac{\sum_{f \in F_p \cap F_q} N_t(p,f)}{\sum_{f \in F_p} N(p,f)},$$

It should be intuitive that temporality can provide an additional directional entailment signal here. After all, the reason we know that *beat* $\models$ *play* and *play* $\not\models$ *beat* is that playing leads to beating *some of the time*. Those other times occur with other predicates like *lose*, and we can reflect this in normalization.

Note that discourse effects may work against this. If *beat* is more salient to the

[9] For convenience we notate the fraction in the table — in the computation the sum occurs separately over the numerator and denominator, not over the fractions.

reader, and described more often, then the data might support the other direction, even though under perfect information the correct distribution would be learned.

The effect of triple-level information being biased towards paraphrase, which was also present in the atemporal scores, has not been discussed in previous research as far as we know. Perhaps this is because previous research has focused on distributional inclusion *across* argument pairs, whereas this effect describes distributional inclusion within the occurrences of a single argument pair. Future work should investigate these options, along with the complication of using PMI scores instead. As will be discussed in Chapter 8, it may be worth improving aspects of the pipeline before delving deeper into such algorithmic choices, since it can be challenging to attain reliable experimental answers with the remaining noise in relation extraction, Named Entity Recognition, Linking and Typing, and temporal expression parsing.

More generally, our method could incorporate in its filtering any function of the contextualized eventuality to determine whether their co-occurrence should contribute to an entailment score. In this project a binary decision is made based on time interval overlap and argument pair overlap, but one might use features such as (lexical) aspect, tense, the presence of other entities, etc. Previous work was limited to using the presence of two arguments as a proxy for entailment relevance; with our refinements we could expand to involving not only time but also other features of the contextualized eventualities.

One of the most pressing avenues of future research was to generalize these results beyond the sports domain, which eventually led to the work in Chapter 5. We anticipated that this might require us to set the window dynamically. In the setup presented here, eventualities stay relevant for a similar amount of time, but the varied predicates in the general domain should allow comparison for different granularities of time. For example, the window around a person *being president* should be larger than a person *visiting* a location, because the consequences of *being president* may hold for longer. We had initially planned to learn a different window size per predicate (for example by taking into account average predicate duration and granularity), but eventually settled on dynamic windowing per contextualized eventuality using modern duration prediction methods (Zhou et al., 2020).

4.6 Conclusions

We injected temporal information into the local Entailment Graph construction framework of Hosseini et al. (2018), with the goal of comparing only those eventualities that are temporally near each other. This is achieved by filtering the counts of predicate p according to whether its triples' time intervals overlap with the those of predicate q. We considered a range of new local similarity scores based on both temporally filtered counts and scaled PMI scores, which we evaluate on a semi-automatically constructed dataset, based on manually constructed paraphrase clusters.

Our temporal versions of PMI-based BInc outperform the atemporal version, the previous state-of-the-art measure for predicate entailment. We also show that adding a temporal window around the time intervals of the triple is essential. The performance of the temporal similarity measures over the atemporal measures is particularly strong at the low recall range, and is also useful when only time expressions from the text are used. This suggests that temporality is a useful signal for learning entailment, and points at recall-improving avenues of future research such as the development of sophisticated temporal resolution systems that link more eventualities to time intervals. In the next chapter we explore another avenue of research, investigating the versatility of our method by applying it to the general news domain.

Chapter 5

Temporality in General-Domain Entailment Graphs

5.1 Introduction

In Chapter 4 we proposed an algorithm that prevents Entailment Graphs from learning spurious entailment edges, by taking into account the temporal overlap of eventualities, alongside their occurrence with argument pairs. This effectively reinterprets the DIH's context set as containing both argument pairs and time. The focus of these experiments was the outcome predicates in the sports news domain.

In this chapter, we extend the sports domain work by applying the method to a corpus of general news domain text. In acknowledgment of the temporally more diverse set of predicates in the general domain we propose setting different size temporal comparison windows. We dynamically assign a window for each eventuality in the corpus using a temporally-aware language model (Zhou et al., 2020) that predicts the expected duration of the eventuality, and show initial experimentation comparing this to uniform windows.

The Sports Entailment Dataset (Chapter 4) is unsuitable for evaluating graphs built on the general news domain, motivating the construction of the general-domain ANT dataset — a novel dataset derived from WordNet (Miller, 1993) antonyms.

We find that refining the DIH's context to include time (in addition to argument pairs) is beneficial for the sports news domain, but that this does not extend to the general news domain. We do, however, identify predicates in legal news as another possible area in which temporal information may be useful for learning Entailment Graphs. The contributions of this Chapter are: 1) the application of a temporally informed Entailment Graph learning method to the general news domain, 2) ANT, a novel general-domain entailment dataset based on WordNet antonyms and 3) an analysis that investigates possible reasons why the temporal DIH is effective in some domains, but not others. For a background on temporality, antonym detection and (evaluation of) Entailment Graphs, please consult Chapter 2.

5.2 Method

5.2.1 Relation Extraction

We start by extracting triples from a corpus of news articles. As in Chapter 4 we use MONTEE (Bijl de Vroe et al., 2021), an open-domain system that uses the RotatingCCG parser (Stanojević and Steedman, 2019) and extracts triples consisting of predicates and their arguments by traversing the resulting CCG dependency graph. For each sentence we extract all potential triples of the form *predicate*(*arg1*, *arg2*) (e.g. *beat*(*Arsenal*, *Man_United*))[1]. Arguments, which may be either named entities or general entities (all other nouns and noun phrases), are mapped to their fine-grained FIGER types (Ling and Weld, 2012) (e.g. *person*, *disease*, etc.). For more detail on relation extraction, see Section 2.3.3.1.

We also use the temporally extended MONTEE as before, adding temporal intervals to triples where there is a path in the dependency graph between the predicate and a temporal expression in the text. The temporal intervals consist of the start and end date of the eventuality, and are derived using SUTime (Chang and Manning, 2012).

5.2.2 Graph Learning with Temporal Filtering

To learn Entailment Graphs we build on the temporal filtering method described in Chapter 4 (Guillou et al., 2020) which extends the graph learning framework of Hosseini et al. (2018). As before, the input is the set of typed triples paired with their time intervals, $p(a_1{:}t_1, a_2{:}t_2, [t_s, t_e])$. The output is a set $\mathcal{G}$ of graphs G_{t_1,t_2}, one for each pair of FIGER types found in the set of triples. We focus on locally learned entailments, leaving an investigation of the interaction between temporality and globalization to future work.

[1] As we are not concerned with the intersection of temporality and modality, we do not tag triples for modality.

In Chapter 4, we update the method of Hosseini et al. (2018) to also observe temporality: an eventuality in p's feature vector is retained (and counted) if it is temporally close enough to any eventuality in q's feature vector. The goal of this process is to separate out different instances of antonymous predicates (such as *win* and *lose*) that recur frequently with the same argument pairs. Given a subgraph of correlated predicates *win, lose to* and *play*, we aim to learn the following entailment relations following completion of the temporal filtering process: *beat* $\vDash$ *play*, *lose to* $\vDash$ *play*, and *lose to* $\nvDash$ *beat* (and its reverse). Without temporal filtering a spurious entailment relation between *beat* and *lose to* (and vice versa), which occur within a similar context (i.e. they share the same argument pair), would be learned.

5.2.3 Dynamic Temporal Window

Although a uniform temporal window is suitable for sports matches, which are typically concluded within a single day, it may be less suitable for other eventualities. We follow the recommendation in Chapter 4 and apply a dynamic window on a per-eventuality basis to reflect that different eventualities remain relevant for different lengths of time. For example, the window around information stating that a person *is president* should be larger than a report of a person *visiting a location*.

We incorporate a temporally-aware language model, *TacoLM* (Zhou et al., 2020), and use it as the basis for per-predicate dynamic windowing. *TacoLM* augments language model pre-training to improve their understanding of several important temporal phenomena, including duration. Firstly, training data is generated cheaply from the Gigaword corpus (Napoles et al., 2012), by observing the contexts in which temporal cues occur. For example, the sentence "Jack rested **for 2 hours** before the speech" can be used to generate a training instance in which *resting* takes *hours* within this particular sentential context. Then, a joint training objective is designed by including these labels as part of the input sequence, following Huang et al. (2019). The masked prediction objective can be used for classification.

As such, TacoLM predicts the expected duration of a triple using the context provided by the sentence in which the relevant eventuality mention occurs. For each eventuality in a sentence it assigns a duration label from the set $\{seconds, minutes, hours, days, weeks, months, years, decades, centuries\}$.

We incorporate TacoLM into our corpus preprocessing, associating a duration prediction with each triple in the corpus. In a small number of cases the model is unable to make a prediction, indicated by the *no_prediction* label[2].

Subsequently TacoLM can be used to determine the size of the temporal window. Recall that in the uniform window model each eventuality e is assigned a temporal interval $e_t = [t_{start} - w, t_{end} + w]$, where t_{start} and t_{end} are predicted using SUTime(e), and w is the model's fixed window size. Conversely, we define the window using TacoLM in the dynamic window model. We instead assign $e_t = [t_{start} - map(TLM(e)), t_{end} + map(TLM(e))]$. Here $map(TLM(e))$ is TacoLM's prediction mapped to a concrete duration value: $weeks \mapsto 15$, $months \mapsto 30$, $years \mapsto 365$, $decades \mapsto 3,650$, $centuries \mapsto 36,500$ and $TLM(e) \mapsto 5$ if $TLM(e) \in \{seconds, minutes, hours, days\}$. That is, for shorter durations we maintain the uniform window of 5 days, extending it only for eventualities with longer durations.

5.2.4 Similarity Measures

We compute both a symmetric and a directional temporally-informed similarity measure to learn entailments, making use of the temporally filtered counts and PMI scores described in Section 5.2.2. We adapted BInc (Szpektor and Dagan, 2008) and Weeds' precision (Weeds and Weir, 2003).

We compute **Temporal Weed's precision** using the temporally-filtered counts. For **Temporal BInc-based** measures, we focus on the temporal PMI scores. As in Chapter 4, we refrain from computing conditional PMI between an argument pair, predicate p, and predicate q due to space and time complexity issues. Instead, we scale the PMI scores used to compute BInc. The temporally filtered $PMI_t = PMI \cdot (c_t/c)$, i.e. the original PMI multiplied by the ratio of filtered counts (c_t) to regular counts (c). We refer to this measure as T. Binc (Ratio PMI)[3].

5.3 Evaluation

We evaluate the Entailment Graphs using two different entailment datasets. 1) the Sports Entailment Dataset (Guillou et al., 2020) which contains 1,312 entailment pairs, focusing on events that occur between two sports teams. 2) ANT, a novel dataset based on WordNet antonym pairs. ANT addresses the need for a *general-domain*, LIiC-style dataset containing antonyms.

[2] 249,262 [0.61%] eventuality mentions in the NEWSSPIKE corpus

[3] We limit the set of similarity scores again for a more concise story. For example, Binary PMI is not presented since it does not differ significantly.

5.3.1 ANT Dataset Construction Overview

ANT contains entailment pair examples of the form *premise, hypothesis, label*. The premise and hypothesis take the form of natural English sentences containing a subject, predicate, and object. The label denotes one of four types of entailment relation: 1) *Antonym*: non-entailments between antonymous predicates (e.g. *acquit - convict*), 2) *Directional Entailments* between an antonymous predicate and a related third predicate (e.g. *acquit* ⊨ *indict*), 3) *Directional Non-Entailments*, the reverse of each *Directional Entailment* (e.g. *indict* ⊭ *acquit*), and 4) *Paraphrases* of each predicate in the antonym pair (e.g. *acquit - absolve*). For a standard entailment evaluation setup, we map: $(Antonyms, Dir.Non\text{-}Entailments) \mapsto 0$ and $(Paraphrases, Dir.Entailments) \mapsto 1$. Our released dataset contains the original four labels as these may be useful in future research. For instance, it may be useful to instead map *antonym* to a separate *contradiction* label (as in NLI), or to keep labels to distinguish between *paraphrases* and *directional entailments*. ANT can also easily be adapted for the evaluation of antonym detection systems.

Dataset construction was semi-automatic. The manual steps were carried out by Liane Guillou and myself (a native and fluent English speaker respectively). Our dataset generation method uses the entailment relations between manually annotated predicate *clusters* to generate entailment pairs. By ensuring that most of the annotation occurs at the *predicate* level, rather than the *predicate-pair* or *sentence-pair* level, we are able to generate thousands of high quality entailment pairs from hundreds of annotated predicates. This is in contrast with the construction processes of the Levy (Levy and Dagan, 2016) and SherLIiC (Schmitt and Schütze, 2019) datasets, which involved generating large numbers of candidate entailment pairs of varying quality, prior to manual annotation by crowd-source workers. Our method also avoids the issue of selection bias present in Zeichner et al. (2012) and SherLIiC, that arises from using a similarity measure to automatically pre-select candidate entailments.

5.3.2 Antonym Pair Selection

We started by automatically collecting a list of 477 lemmatized verb antonym pairs from WordNet (Miller, 1993) and propose these as possible conflicting predicate pairs. Although WordNet's antonym set is not large, the high quality of its annotations makes WordNet a reliable starting point.

Interestingly, we found that WordNet's antonym list contained many temporal entailments (e.g. *fall asleep*-*wake up*; you will always *fall asleep* before you can *wake up*). Although this set may be useful in work continuing the research program set out in Chapter 3, we exclude it here. The pairs contained within are antonymous only when interpreted as simultaneous eventualities (i.e. you cannot *fall asleep* and *wake up* at the same time). If one of the two human annotators marked the antonym pair as having a possible temporal entailment between the predicates, we removed it from the set. This step resulted in 283 remaining antonym pairs.

We also removed pairs that were highly specific (e.g. *dehydrogenate-hydrogenate*) as these are likely to be infrequent in the general domain, pairs resulting from simple alternation of prepositions or morphemes (*scale up*-*scale down*; *deceive*-*undeceive*), and duplicate pairs in the British spelling.[4] We were left with 114 antonym pairs.

5.3.3 Entailment Cluster Construction

For each antonym pair, we identified possible paraphrases and third predicates that are entailed by both. We used the online Merriam-Webster Thesaurus (Merriam-Webster, 2021), which includes both (near) synonyms and antonyms, and the Relatedwords website (RelatedWords, 2021) – an online tool for finding related words beyond synonyms that combines a number of NLP resources including word embedding spaces, ConceptNet and WordNet. This often suggested entailed predicates and helped us find less typical paraphrases.

For each antonym pair we created an *entailment cluster* $C = (A_1, A_2, E)$, where A_1 and A_2 are the sets of predicates containing the *first* and *second* predicate in the seed antonym pair respectively, plus their paraphrases, and E is a set of predicates entailed by all elements in $\cup(A_1, A_2)$.

Each cluster was then manually annotated with a set of argument type pairs (designed for this project and distinct from the FIGER types for named entities), which were later used for instantiating simple sentences. For example, the cluster for the antonym seed pair *refresh-tire* receives a set containing a single argument type pair, *activity#generic_person*. We also allowed predicates with a specific word sense to be assigned a specific set of types. For example, for the *enjoy-suffer through* pair, the entailed predicate *see* is assigned the set containing just the type *generic_person#entertainment_watch*, to avoid it being paired with arguments from the *entertainment_read* type. This also enabled us to specify argument order, allowing

[4] We prefer American English spellings (e.g. *colonize*) over British English spellings (*colonise*) as the training corpus contains mostly American English news articles.

a predicate pair like *refresh(activity#generic_person) - do(generic_person#activity).*

5.3.4 Entailment Pair Generation

The aim of the generation step is to automatically convert the entailment clusters into the dataset format required for evaluation: premise, hypothesis, and a label denoting the type of entailment relation that holds between them.

To generate entailment pairs we take the cross product of different sets in the cluster. *Directional Entailments* are generated by $\cup(A_1 \times E, A_2 \times E)$, *Antonyms* by $\cup(A_1 \times A_2, A_2 \times A_1)$, *Directional Non-Entailments* by $\cup(E \times A_1, E \times A_2)$ and *Paraphrases* by $\cup(A_1 \times A_1, A_2 \times A_2)$, excluding duplicate predicates. We exclude an entailment pair if no intersection is found in the sets of its argument types, or if it already occurs as part of another antonym pair's cluster.

To generate a sentence for a predicate we need to populate its subject and object arguments. We therefore manually created argument strings for each argument type, ensuring they combine effectively with all predicates in the cluster. For example, the argument type *politician* maps to arguments like *Hillary Clinton*, used to instantiate sentences for predicates like *govern*. We used the Relatedwords website (RelatedWords, 2021) for inspiration. We then sampled an argument type pair from the intersection of the type pair sets of both predicates in the entailment pair. For each argument type we sampled non-identical argument strings. This produces an entailment example of the form (*arg1, predicate1, arg2. arg1, predicate2, arg2. label*). For example, (*The school, admitted, Jean. The school, evaluated, Jean. 1*) represents the directional entailment *admit* $\vDash$ *evaluate*. Finally, both annotators made a single pass over the dataset to identify errors, and corrected the clusters accordingly. For example, we encountered unforeseen predicate-argument mismatches stemming from word sense ambiguity. Whilst this refinement method may be repeated indefinitely, we found that after a single manual pass the quality of the generated sentence pairs was very high.

The test portion[5] of ANT (based on 100 WordNet antonym pairs) contains 6,300 entailment pairs: 1,800 Antonyms, 1,465 Directional Entailments, 1,465 Directional Non-Entailments, and 1,570 Paraphrases. For the purpose of evaluation we used the following data subsets: 1) **Base**: *Antonyms* and *Directional Entailments*, and 2) **Directional**: *Antonyms* and *Directional Non-Entailments*.

[5] ANT also contains a small development set (based on 14 antonym pairs) for use with supervised learning techniques

5.3.5 Error Analysis

To verify the dataset's quality we conducted an error analysis on 200 examples, with 50 examples per label sampled randomly from the test set. We found 82.5% (165 /200) examples to be correct, confirming that the dataset is of high quality. Of the 35 incorrect examples we labelled five as a syntactic error, 18 as a semantic error, and 12 as unnatural/disfluent. The syntactic errors were attributed to wrong verb tense or a missing auxiliary verb in the predication. Sometimes semantic errors resulted from the introduction of subtle meaning change, such as for the directional non-entailment "Morgan *changed* the server" - "Morgan upgraded the server" (here *changed* might be interpreted as *replaced*). They also arose due to predicate pairs that were overlooked in cluster construction, e.g. *look down on* is an antonym of *like* but not necessarily a paraphrase of *dislike* - you can *dislike* (a person) without *looking down on* (them). Unnatural sentences were often the result of odd argument-predicate combinations, e.g. "Gale *expended* gas".

5.4 Experimental Setup

We used the NEWSSPIKE corpus of multi-source news text (Zhang and Weld, 2013) for our experiments. NEWSSPIKE comprises approx. Using The relation extraction system (Section 5.2.1) we extracted 40,669,470 triples from NEWSSPIKE. Of these 8,107,944 (19.94%) triples are extracted with a temporal interval resolved by SUTime (Chang and Manning, 2012) from a temporal expression in the text. As the temporal filtering method relies on the information contained in the time intervals to compute the temporal overlap of two eventualities, the sparseness of temporal expressions in the text raises a problem. To address this we employ the strategy described in Chapter 4, using the SUTime temporal interval if it is available and backing off to the document publication date if not. Furthermore we use TacoLM to associate a duration with every triple.

We used the *entGraph*[6] framework (Hosseini et al., 2018) with the extension of temporal filtering by Guillou et al. (2020) to train the Entailment Graphs. We used the default values for all other parameters, except the infrequent predicate and argument cutoffs. With the increased complexity (resulting from the additional inner loops) we found that the algorithm became prohibitively slow for large graphs such

[6] https://github.com/mjhosseini/entGraph

as *thing-thing*. We therefore separately raised its cutoffs to [minPredForArgPair=6] and argument pairs [minArgPairForPred=6], using cutoffs of [minPredForArgPair=4] and [minArgPairForPred=4] for all other type-pair graphs. Although there is room for these hyperparameters to be increased further, it would be useful to find improvements in the algorithm complexity of the method presented in Chapter 4, if the training data's scale is increased further.

With the following exceptions we used MONTEE's default settings to extract triples. We enable the SUTime component [includeTemporal=True]. We disabled unary relation extraction [writeUnaryRels=False], and restricted triples to only those that include at least one named entity [acceptGGBinary=False].

All of the experiments were conducted on a single server which has two Intel Xeon E5-2697 v4 2.3GHz CPUs (each with 18 cores) and 330GB RAM. The computational cost of training a single Entailment Graph is approximately one day (this is the typical time for all steps, with some variation depending on other jobs running on the server) and 160GB RAM (the stable maximum memory usage reached during the local learning step). Evaluation of both the Levy/Holt and ANT datasets using the *entGraph* evaluation scripts takes approximately 6 hours per graph.

We conducted experiments using two main settings. For the sports domain we apply a uniform window of 5 days on either side of the temporal intervals, as suggested by experiments in Chapter 4. We also chose this setting because the evaluation predicates all refer to sports matches. Since these have a short duration and occur frequently between different pairs of teams, the window for which a match stays relevant to the readers, and for which the preconditions and consequences of the eventuality hold, is typically short.

For the general domain the duration of eventualities is highly variable, ranging from minutes or hours, to years, decades, or even centuries. These eventualities may also remain relevant for much longer than the sports matches. We therefore apply a dynamic window around each time interval using TacoLM information (see Section 5.2.3 for details). We carry over the default five day window from the sports setting for shorter durations. In Section 5.5.1 we first show results comparing the uniform and dynamic window strategies. We then compare performance of the temporal method on the general domain ANT dataset compared to *Sports* (Section 5.5.2), before briefly presenting results on the Levy/Holt dataset (Section 5.5.3).

	ANT Base			ANT Directional		
Window Method	Uniform	Dynamic	Atemporal	Uniform	Dynamic	Atemporal
Similarity measures:						
Weed's Pr (Count)			0.181			0.199
T. Weed's Pr (Count)	0.164	0.180		0.177	0.198	
BInc (PMI)			0.161			0.178
T. BInc (Ratio PMI)	0.144	0.161		0.157	0.178	
BInc (Count)			0.159			0.167
T. BInc (Count)	0.144	0.160		0.148	0.167	

Table 5.1: AUC scores for the **Base** and **Dir**ectional subsets of the Sports Entailment and ANT datasets.

5.5 Results

5.5.1 Comparing Uniform and Dynamic Windowing

Results of the temporal scores with a uniform and a dynamic window, alongside the atemporal scores, are presented in Table 5.1. The methods here were all evaluated on the ANT dataset subsets. Note that the atemporal scores are consistently higher than the temporal uniform scores — this effect will be discussed in more detail in the following section.

Comparing uniform and dynamic windows, we find that the dynamic windows consistently achieve higher AUC scores. This seems initially hopeful, because it suggests that the duration information from the TacoLM model is useful and that a dynamic window is more effective. However, we note that all dynamic scores are very close to the atemporal scores. These effects hold for both the base and the directional portions of ANT.

One possible reason for this may again be spurious overlaps — the larger windows result in fewer occasions of filtering, which makes the dynamic scores equal to the atemporal scores. Thus although a window set dynamically per eventuality is a theoretically appealing idea, we leave it to future research to analyze how best to apply this in practice and develop a more effective implementation. For now we therefore take the uniform score to be more representative of the temporal signal, and focus on this for the remainder of the results and analysis.

	Sports		ANT	
Data subset	Base	Dir.	Base	Dir.
Recall < *threshold*	0.75	0.75	0.3	0.3
Similarity measure:				
Weed's Pr (Count)	0.440	0.460	**0.181**	**0.199**
T. Weed's Pr (Count)	0.455	**0.472**	0.164	0.177
BInc (PMI)	0.471	0.432	0.161	0.178
T. BInc (Ratio PMI)	**0.495**	0.437	0.144	0.157
BInc (Count)	0.462	0.419	0.159	0.167
T. BInc (Count)	0.481	0.430	0.144	0.148

Table 5.2: AUC scores for various temporal and atemporal similarity scores on the Base and Directional subsets of the ANT dataset. All scores use a uniform window.

5.5.2 Comparing Domains

Table 5.2 contains AUC scores for the Base and Directional subsets of the *Sports* and ANT datasets. As presented in Chapter 4, temporal measures are consistently higher than their atemporal counterparts for the *Sports* base and directional subsets. For the Base and Directional subsets of ANT, however, the performance of temporal measures is consistently lower than that of the atemporal counterparts. This difference suggests that the atemporal formulation of the DIH by Dagan et al. (1999) and Geffet and Dagan (2005) is appropriate for the general domain, while the temporal formulation is more applicable in the sports domain.

Figure 5.2 contains the precision-recall curves for the *Sports Entailment* and ANT datasets. To provide a fair comparison between similarity scores that have different recall ranges, we compute AUC under a recall threshold, chosen separately for each dataset (See threshold values in Table 5.2). For the *Sports Entailment Dataset* we observe higher precision for the temporal measures compared with their atemporal counterparts at the lower recall ranges.

For the ANT dataset we make two observations. Firstly, recall is very low. This is due to the absence of many of the entailment pairs in the Entailment Graphs. Secondly, in contrast to the Sports Entailment Dataset, the temporal curves are consistently below the atemporal curves, even at the low recall level. It seems that the temporal distributions for eventualities in the more general domain are such that temporal filtering has a counterproductive effect.

The example subgraph in Figure 5.1 sheds some light. For both the Weed's Precision and Temporal Weed's Precision measures, the system outputs a score of zero for all edges marked on this graph. The scores may be partially due to sparse coverage of the predicates in ANT in general, but can also be due to these antonyms being uncorrelated with each other in the data to begin with (as one might initially expect for antonyms). In that case, applying the temporal algorithm to further filter down the scores will have no effect, and thus the temporal algorithm only adds sparsity in cases where the signal is necessary. We investigate related explanations in the analysis in Section 5.6.

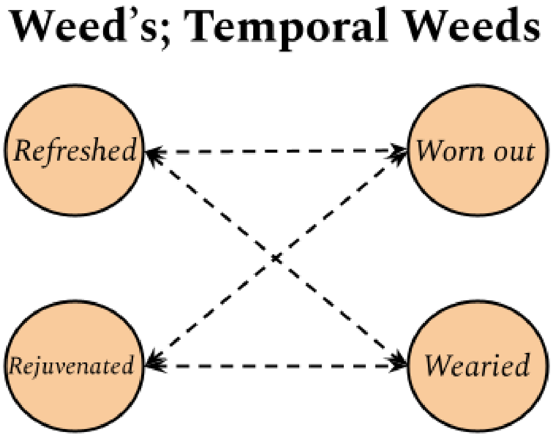


Figure 5.1: Subgraph between the predicates *refreshed by* and *rejuvenated by*, and their antonyms *worn out by* and *wearied by* from the ANT dataset. For both Weed's Precision and Temporal Weed's Precision, all scores are 0.

5.5.3 The Levy/Holt Dataset

For completeness we briefly report results on the Levy/Holt (Levy and Dagan, 2016; Holt, 2018) dataset, following previous work (Hosseini et al., 2018, 2019, 2021; McKenna et al., 2021). This dataset is more general-domain than the *Sports Entailment Dataset*, but it is not designed for evaluating performance on the task of temporally separating contradictory eventualities. We use the same dev/test split proposed by (Hosseini et al., 2018): 5,486 pairs for dev and 12,921 pairs for test.

AUC scores are provided in Table 5.3. We again use the uniform temporal measures rather than the dynamic ones. Overall, these results further support the idea that the

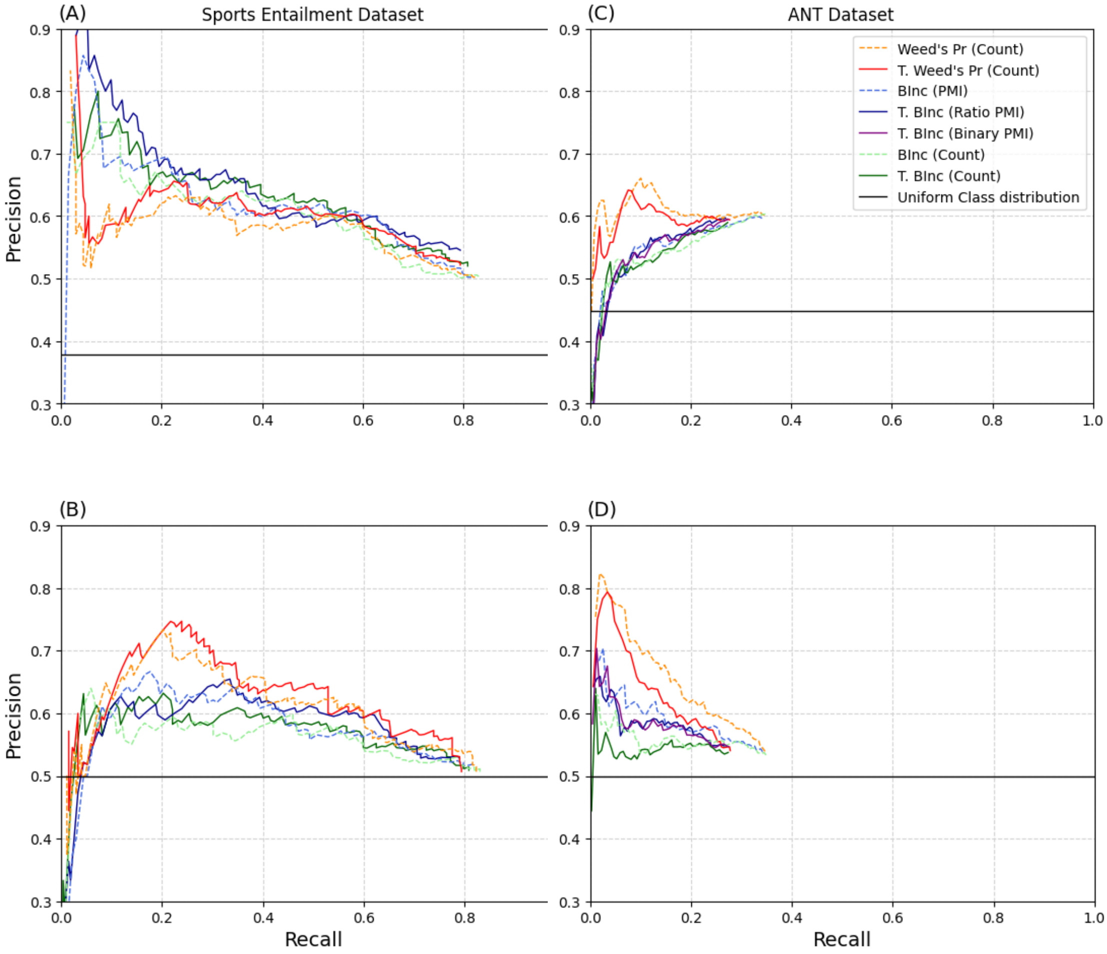


Figure 5.2: Precision-recall plots for the Sports Entailment Dataset (A) base and (B) directional subsets, and the ANT dataset (C) base and (D) directional subsets

temporal distributions for the eventualities in the general domain are such that temporal filtering is not useful — the temporal scores are worse across the board, except for close performance in the complete Dev set[7]. However, the results on ANT are more relevant to this claim, because the *Levy* effect can also be explained by its lack of temporally separable antonyms.

5.6 Analysis and Discussion

Table 5.4 contains statistics of temporal separation for the Base subset of the Sports Entailment and ANT datasets. *% Scaled_down* is the percentage of PMI scores (for

[7] For dynamic windows we see the same effect as in *Sports*. Dynamic window scores converge to very similar levels as the atemporal scores.

	Dev		Test	
Data subset	All	Dir.	All	Dir.
Recall $<$ *threshold*	0.45	0.5	0.45	0.5
Similarity measure:				
Weed's Pr (Count)	0.215	**0.217**	0.207	**0.220**
T. Weed's Pr (Count)	0.215	0.183	0.193	0.194
BInc (PMI)	0.221	0.203	**0.212**	0.203
T. BInc (Ratio PMI)	**0.224**	0.164	0.196	0.176
BInc (Count)	0.217	0.208	0.205	0.201
T. BInc (Count)	0.218	0.173	0.191	0.174

Table 5.3: AUC scores for the Levy/Holt datasets: **All** examples and **Dir**ectional only examples for the dev and test sets. Settings: dynamic window, 5 day default.

		True	False	$\delta(T-F)$
Sports	% Scaled_down	31.5	35.8	-4.2
	% Overlap	72.8	65.8	7.1
ANT	% Scaled_down	53.0	51.8	1.2
	% Overlap	50.4	50.4	0.0

Table 5.4: Analysing the difference in effect of temporal filtering between the Sports and ANT base datasets.

each co-occurrence p,q,ap of triples (p,ap) where p and q are predicates, and ap is an argument pair) that are scaled down by the temporal filtering method. *% Overlap* is the percentage of eventuality comparisons (e_p, e_q) that result in a temporal overlap. When the method is effective, we expect *% Scaled* to be higher for false predicate pairs than true predicate pairs (as scores of antonymous predicate pairs should be scaled down). Scaling should be inversely related to the average *Overlap*, which we expect to be higher for true predicate pairs than false predicate pairs.

We indeed find that *% Scaled* is higher for false predicates pairs in the Sports Entailment Dataset, whereas there is a small difference in the wrong direction for the ANT dataset. This helps explain the differences observed in the Base dataset precision-recall graphs A (Sports Entailment Dataset) and C (ANT dataset) in Figure 5.2. Furthermore, *% Overlap* has the expected correlation, showing that our method works for the temporal distribution of the sports domain data, but not for the general-domain data. That is, it can be applied successfully when there are antonymous predicate pairs that are found applying to the same argument pairs in the data, with argument co-occurrences of the antonym pairs being temporally disjoint more often than the argument co-occurrences of entailing predicate pairs. In our training corpus, this distribution holds for sports predicate pairs but not for general domain predicate pairs. We expect that the decrease in performance on the general domain is due to the scaling and overlap being (fairly) uniformly distributed over True and False predicate pairs — in that case the method simply reduces the amount of data available, while providing no added accuracy.

Breaking down the *% Scaled* statistic per predicate pair in the ANT dataset, we do find antonyms for which many scores are scaled down, indicating that there may still be specific predicates in the general domain where temporality is a useful signal. For example, the antonymous predicate pairs that are scaled most include *violate-respect*, *convict-acquit*, *allow-prohibit* and *(thing) kills (person)-(person) survives (thing)*, suggesting that predicates in *crime* news are worth exploring.

Examples found in the corpus also support this idea for other predicate pairs. We find *"Cameron, who ... , leaves London today ... ."* and *"Cameron will instead stay in London ... ."*, referring to dates a month apart. The atemporal baseline models use this data to erroneously support that *leave* $\models$ *stay in*, whereas our method successfully disentangles the evidence. Future research could further investigate which other subdomains or predicates stand to benefit from temporal information. As mentioned above, the crime news domain is a potential candidate. In general, it seems likely that the method becomes useful in domains where there are antonymous predicates frequently repeating with the same actors, such as with crime news (e.g. *acquit* - *convict*) or political news (e.g. *vote for* - *vote against*) This could inform models that are able to decide whether to apply temporal filtering for particular predicate pairs.

5.7 Conclusion

We applied the temporal filtering method of Chapter 4 Guillou et al. (2020) to the construction of Entailment Graphs for the general news domain, and compared performance across different domains. The results on the *Sports Entailment Dataset* suggest that a reformulation of the Distributional Inclusion Hypothesis that incorporates time could be beneficial for the sports domain. In contrast, the results on the general-domain ANT dataset suggest that the *atemporal* formulation is appropriate for the general domain, although there may still be specific predicates for which the temporal formulation is effective. Our analysis shows that the temporal formulation of the DIH is best applied to domains where the predicates have particular temporal properties — there must be false predicate pairs for which the scores are scaled down (this happens when they co-occur with the same argument pairs, but at different times) while the true predicate pairs should overlap more often.

Chapter 6

CCG-Based Modality Tagging

6.1 Introduction

As introduced in Chapter 1, linguistic modality also has potential as a learning signal for Entailment Graph Induction. We investigate this usefulness in Chapter 7. In order to perform that investigation, we required a modality tagger that can judge whether each eventuality in a corpus is asserted as actually occurring. This Chapter will describe our implementation of a CCG-based modality tagger for that purpose.

As described in Chapter 2, linguistic modality is frequently used in natural language to express uncertainty regarding the occurrence of eventualities. Downstream NLP tasks that depend on knowing whether an eventuality actually occurred, such as Knowledge Graph construction, Fact-checking and Question Answering can benefit from understanding modality. Modal information is crucial in the medical domain, for instance, where it facilitates more accurate Information Extraction and search for radiology reports (Wu et al., 2011; Peng et al., 2018).

Similarly, if a Question Answering system receives the query *Did the protesters attack the police?*, the answer will be different depending on the evidence observed: *Protesters attacked the police* (**True**) or *Protesters are unlikely to have attacked the police* (**Uncertain**)[1]. These challenges are exacerbated by the prevalence of the phenomenon. In a multi-domain uncertainty corpus (Szarvas et al., 2012), sentences containing uncertainty cues are significantly more common in newswire text (18%) compared to encyclopedic text (13%). Modality is also commonly observed in editorials (Bonyadi, 2011).

[1] Assuming trustworthy source text

Category	Example
∅	Protesters attacked the police
Negation	Protesters did **not** attack the police
Lexical negation	Protesters **refrained** from attacking the police
Modal operator	Protesters **may** have attacked the police
Conditional	**If** protesters attack the police ... .
Counterfactual	**Had** protesters attacked the police ... .
Propositional attitude	Journalists **said** that protesters attacked the police

Table 6.1: Modality and negation categories

We present MONTEE[2], an open-domain system for **Mo**dality and **N**egation **T**agging in **E**vent **E**xtraction, built on top of the existing relation extraction pipeline. Tagging these phenomena allows us to distinguish between eventualities that took place (e.g. *Protesters attacked the police*), those that did not take place (*Had protesters attacked the police ... .*), or are uncertain at the time that a document is written (*Protesters may have attacked the police*). We also show that within the news genre, modality is common in the politics and sports domains, where experts often make predictions and state their opinions on the possible outcomes of eventualities such as elections or sports matches, and analyse alternative outcomes where situations unfold differently.

The extracted relations, as in previous chapters, consist of a predicate and one or two arguments, for example: $attack(protesters, police)$ (from the sentence *Protesters attacked the police*). The predicates are analysed according to the following semantic phenomena: negation, lexical negation, modal operators, conditionality, counterfactuality and propositional attitude. See Table 6.1 for examples of each category. Our tagger depends on a novel modality lexicon contributed as part of this project. The lexicon contains words and phrases that trigger modality, and was compiled from a number of different resources. It is unique in its breadth, to our knowledge capturing more phenomena than the lexicons from which it is composed. Finally, we present a corpus study comparing different domains of a large corpus of news text.

[2] https://gitlab.com/lianeg/montee

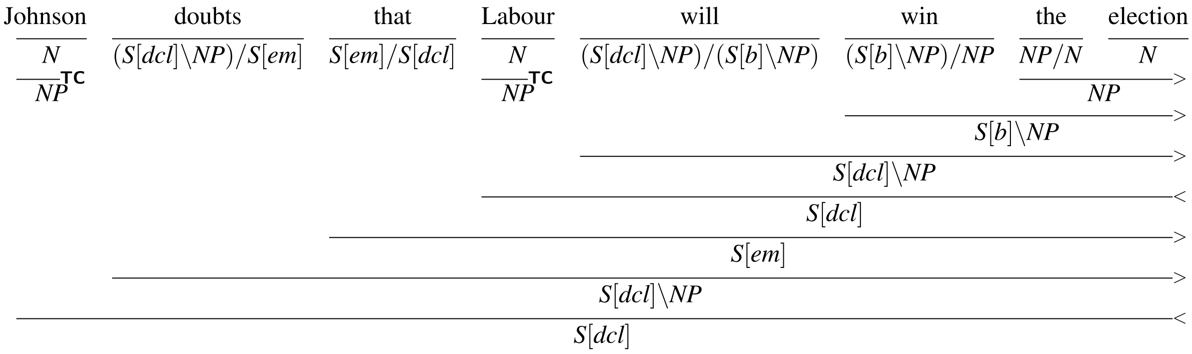


Figure 6.1: CCG parse tree for *Johnson doubts that Labour will win the election*

6.2 Modality Tagger

6.2.1 Relation Extraction System Overview

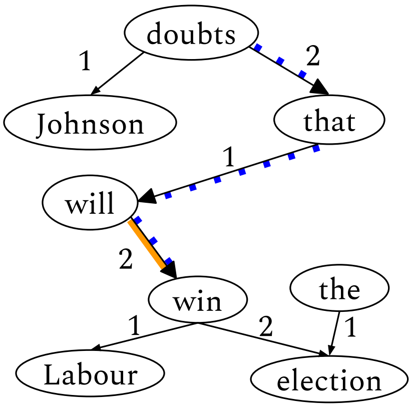


Figure 6.2: CCG dependency graph for *Johnson doubts that Labour will win the election*; marked paths from *doubts* (blue, dotted) and *will* (orange, solid) to *win*.

In Section 2.5 we discuss the linguistic side of the modal phenomena exemplified in Table 6.1, as well as some of the NLP problem formulations and approaches that have been applied to them. Although there are many existing open-domain relation extraction systems, none capture the full range of phenomena described here. For example, neither OpenIE nor OLLIE handle some of the phenomena we are interested in (in particular counterfactuals and lexical negation), and they fail to extract unary relations (see Section 6.3 for a comparison of our system with OpenIE and OLLIE)[3]. Since we believe it necessary to handle these phenomena in order to perform the investigation in Chapter 7, we therefore expand the relation extraction system used in

[3] Note that expanding the system to handle unary relations is also useful for the work in (McKenna et al., 2021).

Chapters 4 and 5 to take the phenomena into account[4].

Our system takes as input a text document, and for each sentence outputs a set of tuples, as in Section 2.3.3.1. A relation tuple consists of a predicate and one or two arguments ($p(a)$ or $p(a_1,a_2)$, e.g. $ended(the\_protest)$, $addressed(Angela\_Merkel,$ $NPD\_protesters))$. Typing will be omitted in the notation for succinctness.

Each sentence in the document is parsed using the *RotatingCCG* parser (Stanojević and Steedman, 2019) over which we construct a CCG dependency graph using a method similar to the one proposed by Clark et al. (2002). (See Figure 6.2 for an example of a dependency graph and Figure 6.1 for the CCG parse tree from which it was extracted). CCG dependency graphs are more expressive than standard dependency trees, allowing them the model phenomena that may interact with modality, such as long-range dependencies (e.g. "The government is ... , they believed") and coordination ("Merkel might *win* and *celebrate* next week").

As before, we traverse the dependency graph, starting from verb and preposition nodes, until an argument node is reached. The traversed nodes, which are used to form the predicate strings, may include (non-auxiliary) verbs, verb particles, adjectives, and prepositions. The CCG argument slot position, corresponding to the grammatical case of the argument (e.g. 1 for nominative, 2 for accusative), is appended to the predicate.

Our focus is on the extraction of binary and unary relations. Binary relations may be extracted from dependency paths between two entities. Extraction of unary relations, which have only one such endpoint, poses a unique challenge (Szpektor and Dagan, 2008) – we must decide whether they are truly a unary relation, or form part of a binary relation. Therefore linguistic knowledge must be carefully applied to extract meaningful unary relations (similar to the rules for binary relations described in Section 2.3.3.1). We extract unary relations for the following cases: verbs with a single argument including intransitives (*bombs exploded*) and passivized transitives (*protests were held*), and copular constructions (*Greta Thunberg is a climate activist*).

In addition to binary and unary relations we also extract n-ary relations which combine two binary relations via prepositional attachment. These are of the form: $predicate\_arg2\_preposition(arg1,arg3)$, and are constructed by combining the two binary relations $predicate(arg1,arg2)$ and $preposition(arg2,arg3)$. For example *marched_on(protesters,Parliament_Square)* and $in(Parliament\_Square, London)$ combine to form the new relation $marched\_on\_Parliament\_Square\_in(protesters,London)$

[4] Note that (Bijl de Vroe et al., 2021) was the first mention of the pipeline as as a standalone system, presented separately from Entailment Graph experimentation.

Lemma	Category	POS-tag	Strength
succeed	MOD	VB	4
shall	MOD	MD	3
conceivably	MOD	RB	2
impossible	MOD	JJ	0
as long as	COND	RB	2
concede	ATT_SAY	VB	4
reckon	ATT_THINK	VB	2

Table 6.2: Example lexicon entries

(from the sentence: *Protesters marched on Parliament Square in London*).

The type system is identical to that described in Section 2.3.3.1, and may be leveraged to identify eventualities belonging to specific domains. This supports our corpus analysis in Section 6.5, allowing us to identify and track, for instance, political eventualities such as elections, debates and protests, according to their occurrence with political entities.

6.2.2 Lexicon

Since many of the phenomena we capture involve lexical trigger items, we opt for a lexicon-based approach. Triggers identified using the lexicon can then be linked to predicate nodes in the CCG dependency graph. Entries in the lexicon cover modality, lexical negation, propositional attitude, and conditionality, with counterfactuality handled separately. Each entry contains the lemma, the categories that it covers, the POS-tag and an estimate of the epistemic strength that the word would often indicate. A few examples are included in Table 6.2.

Our lexicon is constructed by pooling together various lexical resources. The majority of the entries derive from the modality lexicon presented by Baker et al. (2010), who use it for a similar rule-based tagging approach. Their lexicon contains just under a thousand instances, but includes multiple forms for each verb inflection. Using only infinitival forms, we add approximately 200 of the modal entries to our own lexicon.

For modeling propositional attitude (largely ignored in Baker et al. (2010)), we include a list of reporting verbs found in (Fay, 1990). This expands the resource by another 120 phrases. The entries are separated by attitudes that are stated ("attitude say", tag ATT_SAY, e.g. *say, state*) and attitudes of thought (tag ATT_THINK, e.g.

suspect, assume).

More phrases expressing uncertainty are found in the analysis described in Chapter 4 of news domain sentences describing conflicting eventualities. Sentences describing simultaneous *win* and *loss* events, for example, often contained modal descriptions of eventualities that didn't actually happen. Yet more related words were found by generating each entry's WordNet synonyms and antonyms (Miller, 1993). We filtered and annotated these manually to obtain just under another 200 phrases, and added these to the lexicon. We also took inspiration from Somasundaran et al. (2007), especially for conditionals. In aggregate, this work resulted in a resource of 530 phrases.

We also annotated each phrase with a modal category. Our lexicon contains the categories *deontic*, *intention* and *desire*, and for the remaining phrases lists a kind of epistemic strength, with values 4 (*definitely*), 3 (*probably*), 2 (*possibly*), 1 (*probably not*) and 0 (*definitely not*). The latter correspond to lexical negation. These epistemic strength values may optionally be used to use specific subsets of predicates under modal scope, such as those marked with *probable* modals.

6.2.3 Tagger

We use the CCG-based relation extraction system (Section 6.2.1) and the expanded modality lexicon (Section 6.2.2) in tandem to assign modal categories to relations. The procedure is described in Algorithm 2. The focus of the tagger is to identify the bulk of uncertain relations: we prioritize recall over precision, so that we can expect relations without a tag to have actually happened.

The relation extractor produces a CCG dependency graph $\mathcal{G}$ that contains a node n for each word in the sentence (line 2 of the algorithm). We then decide which of these nodes is a trigger (lines 4-7). For modality, negation, lexical negation, propositional attitude and conditionals, we tag these nodes if the node's lemma is present in the lexicon (*check_lexicon* function, line 5). The loop in the algorithm covers the simple case of single token modal triggers (such as *possible*); in the implementation we extend it to multi token triggers (e.g. *shoot for*).

Counterfactual nodes are identified separately. The *check_cf* function (line 5) finds instances of the token "had" that are assigned one of two indicative CCG supertags: $(((S\backslash NP)\backslash(S\backslash NP))/(S[pt]\backslash NP))/NP$ or $((S/S)/(S[pt]\backslash NP))/NP$. For example in the sentence *The protesters would have been arrested, had they attacked the police*, the token "had" would be assigned the CCG supertag $(((S\backslash NP)\backslash(S\backslash NP))/(S[pt]\backslash NP))/NP$

Algorithm 2 Tagging Modal Relations

```
 1: procedure TAGMODALEVENTS(sentence s, predicates p, lexicon l)
 2:     $\mathcal{G}$, pred_nodes ← CCG_dep_parse(s, p)
 3:     trigger_nodes ← [ ]
 4:     for n in $\mathcal{G}$ do
 5:         if check_lexicon(n,l) or check_cf(n,$\mathcal{G}$) then
 6:             trigger_nodes.add(n)
 7:         end if
 8:     end for
 9:     for p_n in pred_nodes do
10:         for t_n in trigger_nodes do
11:             if path_between(p_n, t_n) then
12:                 p_n ← update(p_n,t_n.tag)
13:             end if
14:         end for
15:         p_n.tag ← tag_precedence(p_n)
16:         pred_nodes.update(p_n)
17:     end for
18:     return pred_nodes
19: end procedure
```

and is therefore recognized as an instance of counterfactual had. Additionally, any instance of "if" that syntactically governs an instance of "had", is labelled as counterfactual. Upon realising that even this common counterfactual pattern was rare in the corpus, we decided not to engineer further counterfactual patterns.

We can then decide whether a predicate node should be tagged, by checking whether there is a path in the dependency graph from the trigger nodes to the predicate node (lines 9-12). Figure 6.2 illustrates the intuition behind walking the dependency graph. The graph shows a path from both *doubt* and *will* to *win*. This strategy is effective because the existence of a path between a trigger node and an predicate node corresponds to the trigger node taking syntactic scope over the predicate node. The semantic phenomena we handle all rely heavily on this syntactic process (for example negation, see McKenna and Steedman (2020)).

A single predicate node may be connected to multiple triggers (e.g. in *might not play*, *play* is connected to both the triggers *might* and *not*). We therefore choose the final tag on line 15. Since our primary concern is whether the eventuality happened,

MONTEE	OpenIE	OLLIE
The guerrillas are ready to talk with the Soviets, if Moscow is willing.		
MOD_(guerrillas; talk; Soviets)	(guerrillas; are; ready)	(Moscow; is; willing)
COND_(Moscow; be willing)	(guerrillas; talk with; Soviets)	
	(guerrillas; talk; if Moscow is willing)	
	(guerrillas; talk; willing)	
	(Moscow; is; if Moscow is willing)	
	(Moscow; is; willing)	
Had Trump won the election, Cummings would still be in Downing Street.		
COUNT_(Trump; win; election)	(Trump; Had Trump won; election)	(Trump; Had won; the election)
MOD_(Cummings; be in; D.St.)	(Cummings; would; would still be in D.St.)	(Cummings; would still be in; D.St.)
Protesters did not attack the Police.		
NEG_(Protesters; attack; police)	∅	(Protesters; did not attack; the police)
Parliament failed to investigate the Kremlin.		
(Parliament; failed to investigate; Kremlin)	(Parliament; investigate; Kremlin)	(Parliament; failed to investigate; the Kremlin)
LNEG_(Parliament.; investigate; Kremlin)		(Parliament; to investigate; the Kremlin)
Ed Miliband says the government betrayed Yorkshire.		
ATT_SAY_(government; betray; Yorkshire)	∅	(the government; betrayed; Yorkshire)
(Ed-Miliband; say)		[attrib=Ed Miliband says]

Table 6.3: Comparison of our system with OpenIE and OLLIE. We borrow their notation for clarity, using *argument - predicate - argument*.

we do not combine tags and instead assign a single tag based on the following order of precedence: MOD > ATT_SAY > ATT_THINK > COND > COUNT > LNEG > NEG. The negation categories need to be ordered last because an relation that is negated and modal is still uncertain (e.g. *might not play* shouldn't result in NEG_play), but the ordering is otherwise arbitrary.

6.3 Comparison with Existing Relation Extraction Systems

We highlight the capabilities of our system on five example sentences, comparing with two existing open-domain relation extraction systems: OpenIE (Angeli et al., 2015) and OLLIE (Mausam et al., 2012). See Table 6.3 for a comparison of the relations extracted by our system, OpenIE and OLLIE. The examples are all naturally occurring sentences from the news domain, obtained by a web search targeted to the modality categories discussed in this chapter. To enable a fair comparison, we focus on the extraction of binary relations, as neither OpenIE nor OLLIE was designed to extract unary relations.

While Stanford OpenIE (Angeli et al., 2015), OLLIE (Mausam et al., 2012), and

OLLIE's predecessor REVERB (Fader et al., 2011) may be used to extract binary relations for eventualities, they do not explicitly mark eventualities for modality or negation. Stanford OpenIE (Angeli et al., 2015) typically includes modals as part of the predicate (for example: *(Protesters; may have attacked; police)*), but ignores the other categories of linguistic modality described in Section 6.2.3. In particular, it does not extract relations for sentences involving negation or propositional attitude, omits lexical negations, and is easily confused by sentences involving conditionals or counterfactuals.

OLLIE (Mausam et al., 2012) handles the phenomena in more detail. It identifies conditionals by detecting markers such as *if* and *when*, and labels the *enabling condition* for extracted relations that are governed by a conditional[5]. It typically includes modals and negation as part of the predicate, and captures propositional attitude in its handling of attribution (e.g. *Ed Miliband says ...* .). Like OpenIE, OLLIE is not designed to handle counterfactuals. In terms of lexical negations, OLLIE extracts the predicate both with and without the negation cue (e.g. *failed to investigate* and *to investigate*), which is undesirable if the downstream NLP application needs to be able to distinguish between eventualities that took place and those that did not.

6.4 Tagger Evaluation

We conduct an intrinsic evaluation of our modality-aware relation extraction system[6], measuring performance on a set of 100 extracted relations with manually annotated tags.

We identified the set of articles from the NEWSSPIKE corpus (Zhang and Weld, 2013) for which at least 20% of the relations contain tags, and from these we randomly selected five articles. We then processed the articles using our system to extract relations. From the extraction we selected 100 relations for inclusion in our evaluation set. We excluded those for which the predicate contains only a preposition as these have little meaning unless they form part of a high-order n-ary relation. At the sentence-level we ensured that we include only one relation for each predicate node in the dependency graph, since all relations with the same predicate node will be assigned the same modality. The set of 100 relations was manually annotated by Liane Guillou

[5] The labeling of *conditional* is not applied in the first example in Table 6.3 as no relation is extracted for the consequent.

[6] We exclude both OLLIE and OpenIE from this evaluation as neither system is designed to handle the modality or negation phenomena (c.f. Section 6.3)

	Precision	**Recall**	**F1**
Micro-average	0.81	0.81	0.81
Macro-average	0.72	0.88	0.76

Table 6.4: Intrinsic evaluation results

(a native English speaker) and myself (fluent). For each example, we asked the annotators to answer the question *Does the text entail that the eventuality definitely happens?* using the following scale: the eventuality happened (2), is uncertain (1), didn't happen (0). Inter-annotator agreement over the set of 100 examples was measured using Cohen's Kappa (Cohen, 1960). The agreement score was 0.77, indicating *substantial agreement*, and the annotations differed for only 16 examples.

Following the completion of the annotation task, the two annotators resolved the disagreements. This reconciled annotation was then used as the gold standard against which system performance was evaluated. System-assigned modal and negation tags were mapped to the scale used in the manual evaluation, with LNEG and NEG tags mapped to 0 (didn't happen), empty tags mapped to 2 (happened), and all other tags mapped to 1 (uncertain). In Table 6.4 we report the micro- and macro-averaged precision, recall and F1 scores. As the number of examples per tag type is too small for a meaningful error analysis over types, we provide aggregated scores. The distribution of labels is also uneven, with few negations marked in the gold standard. We therefore take the micro-averaged F1 score of 0.81 to be the definitive result.

We performed an error analysis of the 17 errors made by our system. Parsing was a common issue, with five errors attributed to general dependency parsing mistakes, and five errors due to missing dependency links between reporting verbs and predicates in quoted text (e.g. *"Police were attacked", they said*). Two mistakes were due to human error, as the annotators also missed these reporting verbs in longer sentences. Then, three errors arose from shortcomings of the lexicon. Two of these stemmed from lack of coverage: our lexicon does not handle temporal displacement, as in *We won't act **until** the white house gives more information*. The other was caused by incorrect application of a lexical entry, which would need to be disambiguated by context. Finally, two errors could also have been avoided by treating linguistic aspect, as in *They began the process to ...* .. Future research could thus focus on expanding the lexicon by these final categories of displacement, and take context into account when linking a words in the sentence to entries in the lexicon.

6.5 Corpus Analysis

We conducted a corpus analysis of extracted relations over the NEWSSPIKE corpus (Zhang and Weld, 2013). NEWSSPIKE contains approximately 540K multi-source news articles (approximately 20M sentences) collected within a period of six weeks. We report on the distributions of tagged phenomena over the set of binary relations extracted from news articles in the complete corpus (general domain), and for the subsets of articles related to the politics and sports domains.

The NEWSSPIKE corpus does not include topic or domain information in the article-level metadata. Therefore to identify articles belonging to the politics and sports domains we leveraged the Named Entity Linker AIDA-Light (Nguyen et al., 2014) and the FIGER (Ling and Weld, 2012) type system. We first identified the set of fine-grained FIGER types related to each sub-domain, and then obtained the set of entities belonging to each type. Next we used the output of AIDA-Light to identify the set of articles for which more than 40% of the entities found by the linker belonged to the politics domain, with at least two political entities. We repeated this process for the sports domain, with a lowered threshold of 25%, as the sports topic is less likely to overlap with other topics.

The distribution of relation tags over the general, politics, and sports domains is shown in Table 6.5. For the politics domain approximately 25% of the extracted relations are tagged by the modality tagger, which is more than for the sports or general domains. In particular, modals and propositional attitude verbs belonging to the *say* category are more prevalent. This suggests that while it is important to identify modality in the general news domain, it is particularly important in the politics domain.

The top ten most frequent trigger words found in the general domain are: the propositional attitude trigger *say*, the modal triggers *will*, *would*, *can*, *could*, *may*, *should*, *want* and *have to*, and the conditional trigger *if*. The same top ten are also observed for the politics domain (with different frequencies), and for the sports domain the propositional attitude trigger *think* replaces *want*. The similarity of these lists is perhaps due all domains belonging to the more general news genre.

6.5.1 Future Work

An obvious limitation of our approach is that it does not take into account the context in which eventualities and trigger words occur. Modality is a context-dependent phenomenon, so using the sentential context would improve accuracy. For example, the

	General	**Politics**	**Sports**
Articles	532,651	58,521	196,098
Sentences	20,683,584	2,280,312	8,056,704
Relations	96,774,467	11,265,585	37,936,677
Distribution of tags (as a percentage of relations)			
∅	77.83	74.78	78.82
Modal	14.32	16.65	13.83
ATT_say	4.77	5.37	4.24
ATT_think	0.49	0.43	0.49
Conditional	0.89	1.03	0.85
Counterfactual	0.04	0.05	0.04
Negation	1.52	1.51	1.66
Lexical Negation	0.13	0.17	0.14

Table 6.5: Relation tagging summary by news domain

word *unbelievable* is ambiguous between an *unlikely* and an *amazing, and happened* reading. Relatedly, our concept of epistemic strength is highly context-sensitive, and requires further development. A promising avenue is to develop a pre-training procedure for a modality-aware contextualized language model, in a similar direction as Zhou et al. (2020). We plan to use our modal lexicon to identify sentences with modality triggers. We will then gather human annotations of the certainty that each eventuality happened, and use this annotated data to train a modality-aware language model able to classify eventuality uncertainty. Such a system might eventually even tackle the long-tail of modal examples mentioned in Section 2.5.

Our system was developed for English, but work is already underway to develop relation extraction systems for other languages including German and Chinese. Extending to other languages would allow us to apply our methods to multilingual and cross-lingual NLP tasks. Finally, most CCG parsers, including the one used in this work, are trained on English CCGbank (Hockenmaier and Steedman, 2007b). This makes them perform well on news text, but accuracy suffers on out-of-domain sentences, primarily those involving questions. The results could be improved by retraining the parser on the CCG annotated questions dataset (Rimell and Clark, 2008; Yoshikawa et al., 2019), allowing us to apply our system to the task of open-domain Question Answering in an extrinsic evaluation.

6.6 Conclusion

This chapter presented MoNTEE, a modality-aware relation extraction system that can distinguish between eventualities that took place, did not take place, and for which there is a degree of uncertainty. Our tagger shows strong performance on an intrinsic evaluation of examples from the politics domain and our corpus analysis supports that modality is an important phenomenon to handle in this domain. Being able to make such distinctions is crucial for many downstream NLP applications, including Knowledge Graph construction and Question Answering. In the following chapter, we will investigate the tagger's usefulness in Entailment Graph Induction.

Chapter 7

Modality in Entailment Graph Induction

7.1 Introduction

Detecting modality, uncertainty and factivity is crucial to downstream NLP tasks such as Information Extraction (Karttunen and Zaenen, 2005; Farkas et al., 2010), Information Retrieval (Vincze, 2014), machine reading (Morante and Daelemans, 2012), and Question Answering (Jean et al., 2016). One might expect that it would also be useful in learning Entailment Graphs. That is, Entailment Graphs would be more reliable if learned from data in which predications are *asserted* as actually happening, rather than data with *uncertain* predications under scope of various types of modality. In this chapter (also published as (Guillou et al., 2021)) we investigate whether this is the case.

The Entailment Graph-learning algorithm depends on descriptions of eventualities in the news, observing directional co-occurrences of typed predicates and their arguments. For example, we expect to observe all the arguments of *being president*, such as *Biden* and *Obama*, also to be encountered in a sufficiently large multiply-sourced body of text as arguments of *running for president*, but not the other way around (*Hillary Clinton* will *run* but not *be president*). However, if all the reports of *Clinton* ***might*** *be president* are extracted as *be_president(Clinton)*, one would expect the learning signal to be confusing to the algorithm.

We use the method of Hosseini et al. (2018) combined with the modality tagger MONTEE (Chapter 6, (Bijl de Vroe et al., 2021)) to construct typed Entailment Graphs from raw text corpora under two different settings. In the *modality-aware* condition, modal predications are removed from the data entirely, while in the *modality-unaware* the model learns from both asserted and modal predications as usual. We show that ignoring modal distinctions counterintuitively helps Entailment Graph induction in the general domain. However, we also investigate whether this effect applies uniformly across different sub-domains, showing that the modality-aware condition performs well when evaluated on outcome predicates in the sports domain. While modality annotation is clearly useful for recognising entailment from a given text (Snow et al., 2006; MacCartney et al., 2006), to our knowledge no research has been conducted on its effect on learning Entailment Graphs.

7.2 Methods

We extend relation extraction to pay attention to modality, so that we can distinguish modal and non-modal relations in the Entailment Graph mining algorithm. This allows us to investigate the impact of modalized predicate data on the accuracy of learned entailment edges.

We extract triples of the form *predicate(arg1,arg2)* using MONTEE, the open-domain modality-aware relation extraction system described in Chapter 6. A triple is tagged as modal (MOD), propositional attitude (ATT_SAY, ATT_THINK) or conditional (COND) if the CCG dependency graph contains a path between a relation node and a node matching an entry in the MONTEE lexicon. Counterfactuals (COUNT) are tagged according to hand-crafted rules. Since we focus on uncertainty and not negation, lexical negation (LNEG) tagging is ignored.

In the modality-aware setting, we remove triples tagged by MONTEE as any kind of modal ({MOD, ATT_SAY, ATT_THINK, COUNT, COND}). In local learning, learned entailment edges then have access only to non-modal evidence: eventualities that were asserted as actually happening. For example, the edge between *win* and *lose* should now be learned only from non-modal descriptions such as *A won today against B* or *A has been defeated by B*, leaving out modal descriptions (*A could beat B*).

7.3 Experimental Setup

Using MONTEE[1], we extract 40,669,812 binary relation triples from the NEWSSPIKE corpus (Zhang and Weld, 2013). Of these, 14.57% are tagged; 10.04% MOD, 3.51%

[1] https://gitlab.com/lianeg/montee

REP_SAY, 0.38% REP_THINK, 0.61% COND, and 0.03% COUNT. We then construct three different datasets and build an Entailment Graph with each. The modality-unaware baseline, **BaselineLarge**, is trained on the complete set of triples with modality tags removed. This corresponds to the data and model in Hosseini et al. (2018). For the modality-aware **Asserted** graph, we extract only the set of 34,744,216 asserted relations (∼85% of the relations), i.e. all modal relations are excluded. To rule out effects of data size, we construct **BaselineSmall**, which is trained on a random sample of 85% relations from the total set. Comparing Asserted to BaselineLarge shows us whether it is worth filtering out modal data, and comparing Asserted to BaselineSmall shows whether asserted data or mixed data (i.e. asserted and modal) is more effective for learning entailment relations.

We follow the example of Hosseini et al. (2018) and construct typed graphs for all possible type pairs (e.g. *person-location*). As before, relation arguments are typed by linking to a Named Entity Freebase identifier (Bollacker et al., 2008) using the AIDA-light linker (Nguyen et al., 2014), and mapping these identifiers to a type in the FIGER hierarchy (Ling and Weld, 2012). The typed relations become the input to the graph learning step of the Entailment Graph mining algorithm. Following previous research, we use the BInc similarity score (Szpektor and Dagan, 2008) to compute entailment scores. We first construct local typed Entailment Graphs and then apply the globalization method across graphs Hosseini et al. (2018), see also Section 2.3.3.3.

As before, we evaluate the Entailment Graphs on LIiC-style datasets. Firstly we evaluate on the full 18,407 entailment pairs of the *Levy/Holt Entailment Dataset* (Levy and Dagan, 2016; Holt, 2018). As our training method is unsupervised and we do not tune hyperparameters, we evaluate on the complete Levy/Holt dataset rather than the dev/test split. We also evaluate on the *Sports Entailment Dataset* introduced in Chapter 4, focusing on the directional subset of 718 examples comprising entailments and pairs of match outcome predicates (e.g. *win*, *lose*, *tie*, and their paraphrases) which are always non-entailments. This subset evaluates whether Entailment Graphs can recognize, for example, that win/lose → play but win $\nleftrightarrow$ lose (with similar patterns for other paraphrases of *win, play* and *lose*). We focus on the subgraph of *organizations* as all predicates are assumed to apply to sports teams. Both datasets use binary labels for each premise/hypothesis pair: entailment (1) and non-entailment (0). We did not evaluate on ANT because the experimentation here was carried out before its creation.

We used the entGraph[2] code developed by Hosseini et al. (2018) to construct each

[2] https://github.com/mjhosseini/entGraph

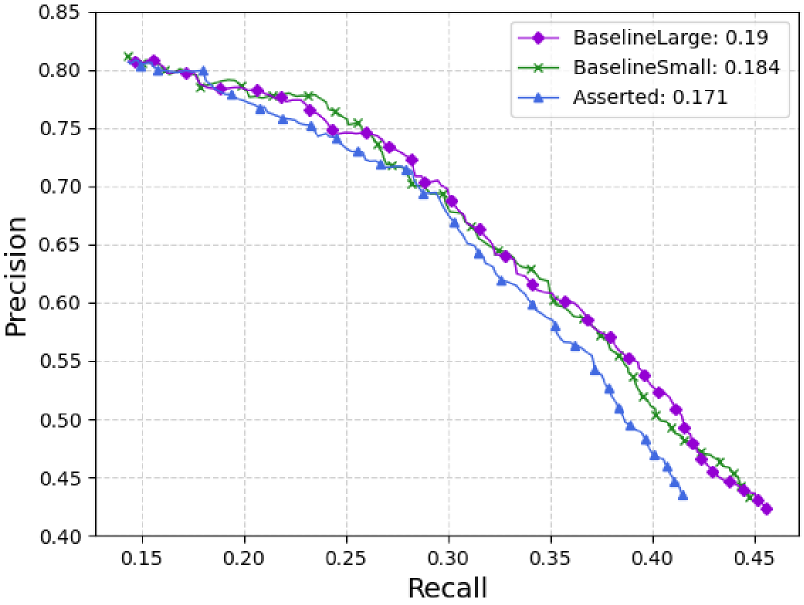


Figure 7.1: Precision-Recall on Levy/Holt

of the Entailment Graphs, and the corresponding evaluation scripts[3] to evaluate performance on the Levy/Holt dataset. Performance on the Sports Entailment Dataset[4] is evaluated using scripts[5] developed for this chapter.

We MoNTEE with the default settings, with the exception of disabling unary relation extraction (writeUnaryRels=False) and restricting binary relations to those that include at least one named entity (acceptGGBinary=False). When using entGraph to construct Entailment Graphs we raised the threshold values for infrequent predicates (minPredForArgPair=4) and argument pairs (minArgPairForPred=4), and used the default values for all other parameters. All experiments were conducted on a single server with 330GB RAM, and two Intel Xeon E5-2697 v4 2.3GHz CPUs (each with 18 cores). The computational cost of training Entailment Graphs under these settings is approximately one day for the local learning step, and eight hours for globalisation.

7.4 Results

Table 7.1 contains AUC scores for Asserted, BaselineSmall, and BaselineLarge on the *complete* Levy/Holt dataset, the directional portion of the Levy/Holt dataset (2,414

[3] https://github.com/mjhosseini/entgraph_eval

[4] https://gitlab.com/lianeg/temporal-entailment-sports-dataset

[5] https://gitlab.com/lianeg/sports-entailment-evaluation

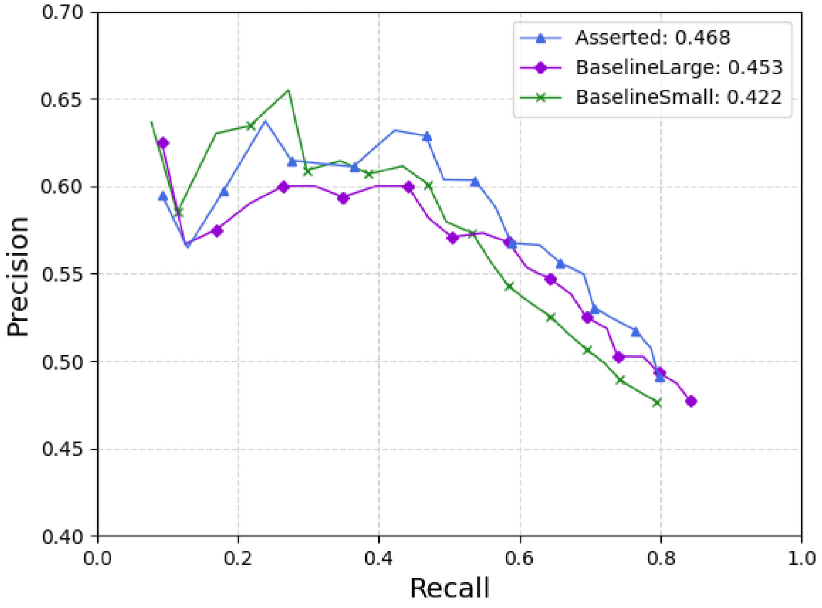


Figure 7.2: Precision-Recall on Sports

	Levy/Holt *complete*	Levy/Holt *directional*	Sports
BaselineLarge	**0.190**	**0.163**	0.453
BaselineSmall	0.184	0.157	0.422
Asserted	0.171	0.136	**0.468**

Table 7.1: AUC scores

examples), and the Sports Entailment dataset. The precision-recall curves for the Levy/Holt (*complete*) and Sports Entailment datasets are displayed in Figures 7.1 and 7.2 respectively. As before, every point on the curve represents a different entailment score threshold (higher thresholds correspond to lower recall and vice versa). We compute AUC for precision in the range [0.5, 1] and over the entire recall range, following Hosseini et al. (2018). All three Entailment Graphs cover this range and predictions with precision higher than random are important for downstream applications.

On the Levy/Holt dataset (all examples), BaselineLarge performs best overall. The strong performance of BaselineLarge compared to Asserted is in itself surprising, and indicates that it is usually not beneficial to distinguish modality when building Entailment Graphs. This can be understood as a data size issue: filtering out data is harmful as it introduces sparsity, and modal data is useful enough to provide a learning signal.

	Nodes	Edges	% Levy predicates found	
			all examples	directional
BaselineLarge	334K	72,7M	63.06	70.29
BaselineSmall	277K	58,4M	61.13	69.29
Asserted	254K	46,3M	58.51	67.92

Table 7.2: Graph size comparison and predicate coverage for Levy/Holt dataset (all examples) and its directional portion

	Nodes	Edges	% Sports predicates found
BaselineLarge	4,514	1.65M	92.86
BaselineSmall	3,823	1.29M	90.48
Asserted	3,682	1.09M	88.10

Table 7.3: *organization* subgraph size comparison and predicate coverage for the Sports Entailment Dataset

More counterintuitive, however, is that even BaselineSmall, which controls for training dataset size, outperforms Asserted. To understand why, we measured the size of each graph in terms of the number of nodes (predicates) and edges (entailment relations) it contained, and the percentage of predicates in the Levy/Holt dataset that were present in the graph (see Table 7.2). This revealed that BaselineSmall contained more of the predicates present in the Levy/Holt dataset, while also being larger in terms of both nodes and edges than Asserted. Thus, Asserted learns with more relations per predicate, while BaselineSmall has more predicate nodes overall. This may lead to the increase in recall that we see for the BaselineSmall graph.

Another explanation might be that this richer predicate coverage allows BaselineSmall to accurately correlate more of the common paraphrase examples in the Levy/Holt dataset. To this end we investigated the directional portion of the Levy/Holt dataset, which contains 2,414 examples of both the entailment pair and its reverse, where the entailment is true in one direction and false in the other. As noted by Hosseini et al. (2018) this task is much harder than that represented by the original dataset. However, the baselines both outperform the Asserted graph on the directional entailment task. We also observe a similar pattern in the percentage of predicates covered (see last column in Table 7.2). In general, we conclude that modal data is useful even for learning directional entailments.

Performance on the Sports Entailment dataset (Figure 7.2) reveals a different pattern. BaselineLarge outperforms BaselineSmall as expected, but Asserted performs best, despite lower coverage of the predicates in the Sports Entailment Dataset (see Table 7.3 for a size comparison of the *organization* subgraph). This supports the suggestion by Guillou et al. (2020) that excluding modal data may help to avoid learning entailments between disjunctive outcomes, i.e. that winning entails losing, which is not measured by the Levy/Holt dataset.

Example subgraphs of the Asserted and BaselineLarge Entailment Graphs are shown in Figure 7.3. As in Chapter 4 we show the entailment scores alongside their rank in the dataset, since the scores may be incomparable between graphs. In these graphs we observed a mixed effect. The edges between the outcome predicates *defeat* and *fall to* both have higher ranks in the Asserted subgraph. That is, the scores between these antonymous predicates are lower relative to the rest of the dataset: the intended effect. However, the score and rank of the incoming edges to *face* stay relatively constant. For these subgraphs, the majority of the benefit can therefore be found in separating the antonyms.

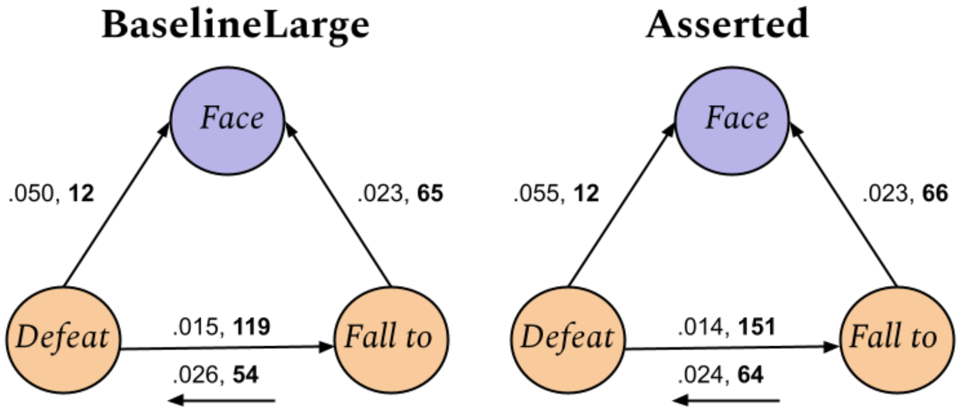


Figure 7.3: Edge values and dataset rank on *base*, for the *face* - *defeat* - *fall to* subgraph. A BInc score graph trained on the BaselineLarge dataset is shown on the left, and a BInc score graph trained on the Asserted data on the right.

7.5 Discussion & Future Work

Understanding the behaviour of the *Asserted* and modalized graphs will come down to understanding the various distributions in which modalized predications occur. One appealing intuition is that the usefulness of modalized predications derives from how they distribute over whether the described eventuality actually ends up occurring. That is, they might generally be expressed in text when the prior probability of the eventuality is already high.

For example, suppose that modalized predications are mostly used to express eventualities that actually occur (e.g. a reporter asserts "Google might buy Youtube", and subsequently Google in fact ends up buying YouTube). In that case, the data they provide should be useful to the algorithm. The DIH is built on the intuition of set inclusion over real world entities, so if modalized data mostly corresponds to actual events, the modalized data ends up being of similar quality to asserted predications. The larger dataset then simply leads to higher recall. On the other hand, modalized predications might more often describe eventualities that don't happen (e.g. somebody reports "Arsenal wants to win against Manchester", and the result is in fact a loss). That leads to an argument pair co-occurrence for *win* and *lose*, falsely strengthening the confidence in the edge $\mathit{win} \models \mathit{lose}$. In that case, we would expect the algorithm to suffer more from the added noise than it benefits from the added data.

Perhaps this explains the difference in performance between the datasets. In the case of the general-domain Levy/Holt dataset, the predicates contained within might be used in the news domain when the prior probability of the eventuality is already high — reporters might be expected not to speculate on unlikely events. This would result in distributions for the main predicates to be improved by the modalized data in spite of the uncertainty of the evidence. The *Sports* dataset then constitutes an exception in which the predicates are more heavily speculated upon despite being uncertain. As such being under modal scope does not necessarily imply a higher prior probability in that case, and the data is less valuable.

Indeed it is easy to find examples in the news corpus to support these intuitions. In the general domain we observe examples of eventualities initially being discussed with uncertainty, and later mentioned as asserted. An example of this is the acquisition of Dell by Michael Dell: on February 5th, 2013 we observe "*... founder and CEO Michael Dell and investment firm Silver Lake Partners will buy Dell.*", and subsequently, on February 6th, 2013 we read *"So Michael Dell and a private equity group* *have bought Dell and taken it private.”*. We also observe the reverse scenario in the sports domain. For example, on January 10th, 2013 we observe *“The popular opinion on this game seems to be Seattle beating Atlanta because ...”*, while shortly afterwards we are informed that *“Falcons come back to beat Seahawks”*.

Part of the effect may also simply be due to the presence of antonyms in the *Sports* evaluation dataset. In general, the worry with modal data is increasing scores of unrelated predicates, antonyms, and hypernyms, which should remain significantly lower than those of hyponyms (note that any noise added to the score of a hyponym relation is beneficial). It may be that modality poses more of a challenge to correctly labelling antonyms than to the other non-entailment categories. After all, in the speculative example of reporters discussing possible sports outcomes (“Arsenal *might win* or *might lose* this weekend”), they are speculating precisely because there are two salient contradictory events that might occur. If it is the case that modal noise is more distracting to antonyms, then the benefit of modality on the sports dataset may be due to the dataset’s high fraction of antonyms compared to Levy/Holt[6].

Note that there is an overlap in the usefulness of modality and temporality under this interpretation. Temporality is useful because it helps us distinguish a *win* on February 3rd from a *loss* on March 24th, whereas modality is useful because it helps us distinguish an actual *win* from some imagined *loss*. In both cases, the signal helps us reduce the association between antonyms by only associating data points that represent an actual co-occurrence.

We may expect to find a similar effect for other subdomains that share the disjunctive outcome property and contain speculation, for example election news, crime news and war news, where modals are used when speculating about potential and contradictory outcomes. Considering the overlap in usefulness of modality and temporality, the domains suggested in Chapter 5 may also stand to benefit from the modal signal. Specifically, it might be easier to correctly learn the semantics of predicates like *acquit* and *convict* in crime news and court case descriptions.

As mentioned previously, when controlling for training data size the *Asserted* dataset is still outperformed by the partially modalized set (*BaselineSmall*). In Section 7.4 we analyze how this may be due to the coverage of predicates of the two graphs, along with their numbers of nodes and edges. However, it is unclear why these statistics should be higher for the model resulting from the mixed dataset. Investigating the distribution of modals over different predicates may help clarify this effect —

[6] Note that ANT was constructed after this experimental work was carried out.

it could be that certain predicates are more likely to be under model scope than others, and that there is insufficient data for those predicates in the *Asserted* condition as a result.

Future research could attempt a deeper corpus study of the distribution of modals. One could investigate how often modalized predications correspond to eventualities that actually end up happening. We can also ask how this interacts with epistemic strength. In other words, do eventualities that are described as more likely (i.e. with a predication that contains a modal of high epistemic strength such as *almost certainly* or *definitely*) actually end up happening more often? If it is true that humans have reasonable intuitions about probability, our models may be able to use this information — there may be some epistemic threshold above which it becomes useful to include modalized predications. The more fine-grained categories of the modality tagger can be used for this purpose (uncertainty detection systems that use the new -3/+3 annotation scheme, as opposed to a binary label, may also be useful). A similar study can be made of other distributions of the modals, such as over data genres and predicate types. We can then leverage this information and retain or remove modal data depending on the sub-domains in question.

The recent developments in modal linguistics should be kept in mind here. The semantics of modal expressions does not behave literally as it is operationalized in the modal logics, just as negation in language does not behave as the negation operator in propositional logic. For example, *definitely(x)* is usually only licensed if there is some reasonable possibility that $\neg x$, so the literal interpretation of $P(x) = 1$ should be put aside for a more semantically and pragmatically informed interpretation.

Another possibility is to consider whether modalized predications are more likely to occur over premises or hypotheses — by extension, how they distribute over the entailment hierarchy in general. They might more frequently occur with more specific, entailed predicates like *win* and *lose*, for example, or perhaps they are used more frequently with predicates in the middle of the hierarchy. Intuitively it is also possible for this distribution to be uniform — in that case if the probability of a premise is high enough to be worth mentioning, then in general that of its entailments are too.

Finally, we can experiment with learning Entailment Graphs with modal predicate nodes, by retaining modal relations with tags attached as input. Many of these entailments are trivial, because any entailment of a consequence can be reproduced under modal scope (if $buy \rightarrow own$, then also $MOD\_buy \rightarrow MOD\_own$). We might also recover the more interesting phenomenon that following an entailment in the reverse direction can produce a modal entailment (e.g. if *beat* → *play*, then we know *play* → *MOD_beat*), and many preconditions will behave interestingly (e.g. *beat* → *play*, but also *MOD_beat* → *play*). To evaluate this idea, we will design a dataset of modal entailments, drawing inspiration from previous research on veridicality in entailment datasets (Staliūnaitė, 2018).

7.6 Conclusion

We have investigated the role of modalized predications in Entailment Graph induction, and shown that there are specific domains in which removing modal data is beneficial (such as the co-occurring antonyms in sports data). Conversely, for the general-domain predicates in the Levy/Holt dataset, modalized predications actually constitute a valuable learning signal — it is better to ignore modality for those predicates.

Chapter 8

Conclusion

The central contribution of this thesis can be summarized as follows: both *temporality* and *modality* can be valuable signals for learning an entailment-based semantics. We explored the interaction of these phenomena by studying their role in improving both the representational power and accuracy of Entailment Graphs. This endeavour contributes to the end goal of allowing systems to better understand the deeper semantics of a piece of text — drawing implications about the world that aren't stated literally.

We have explored the value of temporality and modality in three parts of the problem: the semantic representation, the learning algorithm and the training data. Representationally, we have demonstrated the necessity of including some notion of time, in order to support temporal entailment — inferences that reflect the more complicated temporal relationship between predicate pairs such as *visit* and *arrive*. In the learning algorithm, we have shown that time can be a valuable feature, while modality was useful as a signal in filtering the training data. Still, these latter two effects hold only for specific subdomains, at least under the present experimental conditions.

Finally, in support of these projects we have devised three novel evaluation datasets, one of which presents the relatively unexplored problem of temporal entailment, and two of which highlight the importance of balancing different types of entailment — directional (non-)entailment, antonymy and paraphrase. We also contributed a modality tagging tool, along with an extensive lexicon of modal trigger words.

Chapter 3 defined the Temporal Entailment problem. Acknowledging the temporal nature of the relation between *visit* and *arrive* means a model can avoid potential errors like *will visit* $\models$ *has arrived*, while still modeling the correct entailment *is visiting* $\models$ *has arrived*. In the evaluation dataset TEA we introduced this concept by varying the tense and aspect of predications, which proved challenging to both Entailment Graphs and distributional baselines. We then introduced the same phenomena in the model in Tensed Entailment Graphs, allowing the predicate nodes to be expressed in a variety of morphosyntactic tenses. To this end we added tense and aspect parsing functionality to GraphParser. Our model was able to recover some of the temporal entailments we sought after, but sparsity issues currently prevent it from being practically useful.

In Chapter 4, we explored a different temporal direction, instead injecting temporality as a learning signal in the algorithm, for inducing more accurate Entailment Graphs that contain atemporal nodes. We demonstrate a theoretical drawback of the Distributional Inclusion Hypothesis, which risks learning spurious entailments between antonymous predicates that occur with the same argument pairs (for example, *win* and *lose* are antonyms but may both appear in training data with an argument pair like (*Arsenal*, *Manchester*)). To evaluate the idea of alleviating this issue using temporality, we introduce the *Sports Entailment Dataset*. We introduce the sports domain as a test bed for entailment learning because its data is easily available, and it contains a multitude of named entities that interact frequently. We show that temporality has potential as a learning signal — the benefits of temporality exist across temporal information sources, although for temporal expressions they do not extend into higher recall ranges. Possible solutions and research directions are outlined in Section 8.3.

Chapter 5 continues this investigation beyond the sports domain, inducing graphs for different argument type pairs over the full NewsSpike training corpus. Recognizing that the more heterogenous predications of the general news domain may remain relevant for different periods of time, we dynamically extend the temporal comparison window per eventuality, using the temporally aware language model TacoLM (Zhou et al., 2020). For evaluation, we present the ANT dataset, created from general-domain antonyms drawn from WordNet (Miller, 1993). Although the sports domain effect is not reproduced in the general domain, our analysis shows there may be other subdomains for which the temporal algorithm performs best. Additionally, we show that the algorithm functions as expected; it performs well when the training data contains true predicate pairs that frequently overlap and false predicate pairs that are temporally separable, supporting the intuition behind our algorithm.

Chapter 6 prepares an exploration of linguistic modality in graph induction by constructing a modal tagger. We implement this functionality as an extension of the existing relation extraction framework. Our algorithm finds paths in a CCG dependency parse between modal trigger nodes and predicate nodes, and depends on an extensive modality lexicon composed from various resources. An intrinsic evaluation shows that our tagger is sufficiently high-quality for downstream application.

Finally, Chapter 7 presented research on the role of modality in Entailment Graph induction. We apply the tagger introduced in Chapter 6 to the training data, and compare the relative value of modal and exclusively asserted data. We show that modalized predications are as useful as asserted predications in the general domain, even when controlling for training data size. However, in the sports dataset containing antonyms, modality constitutes a valuable learning signal, mirroring the temporal experiments, and it is useful to remove modalized predications.

This thesis has developed intuitions regarding temporality and modality as features of and learning signals for entailment. Work in these directions is far from finished. We will now contribute various central avenues of research for further exploring entailment-based semantics; questions revealed by the work presented here.

8.1 Entailment Evaluation

This thesis has contributed three datasets, **TEA**, the *Sports Entailment Dataset*, and ANT. We have already discussed some shortcomings of **TEA** and associated solutions in Section 3.2.4. Here I would like to touch on some more general future research directions and lessons learned from entailment dataset construction.

We advocate for an increase in variety of inference types, capturing more linguistic phenomena. Some datasets contain crowd-sourced examples derived from uncurated source material (e.g. (Bowman et al., 2015)). The entailment pairs gathered in this manner are likely to occur in some natural distribution, which can result in many inference types in the long tail being underrepresented. Models trained on these datasets then struggle to generalize. On the other hand, many existing datasets focus on a particular type of inference, such as the **TEA** dataset (Kober et al., 2019). Even if the datasets are used exclusively for evaluation, overfitting may still occur during model development if only one such dataset drives research, once again leading to models that are not robust to novel inferences. We thus follow Poliak (2020) in suggesting a focus on specific linguistic phenomena, highlighting that various types would ideally be collected under a large benchmark.

A varied dataset like this should include inferences that test for knowledge of lexical ambiguity, especially with regard to argument variation. Datasets often contain positive inferences for a variety of senses, but do not contain the challenging negative counterparts. For example, The Levy/Holt dataset does not always include crucial misleading examples. For the lexically ambiguous *kill*, it contains the challenging pair of inferences “The salve kills cancers” ⊨ “cancers may be treated by the salve” and “Crockett was killed at the Alamo” ⊨ “Crockett died at the Alamo”, but does not contain misleading negative counterparts such as “Crockett was killed at the Alamo” ⊭ “Crockett was treated at the Alamo”. Therefore, models evaluated on this dataset can get away with populating a general *thing-thing* graph that can contains the entailment edges of all the various senses of a single word (instead of modeling the inferences separately in a *medicine-disease* and *person-person* graph as intended). Those models will then perform adequately in evaluation, but generate mistakes in downstream tasks. We suggest including these more challenging negative inferences, inspired by the trend of creating datasets that are more adversarial (Nie et al., 2020).

There are other argument-related inference types worth focusing on. For example, LIiC datasets have so far ignored *multivalent* entailments between predicates of different valencies (e.g. the binary *kill* entails a unary *die* pertaining to *win*’s object). Modeling these inferences has recently been explored (McKenna et al., 2021), but they have not been represented in datasets explicitly. McKenna et al. (2021) start with the simplest case of decreasing the valency from binary to unary predicates, but future research may consider alternatives (e.g. ternary to binary, such as *give(person_1, person_2, thing)* ⊨ *have(person_2, thing)*).

A relatively unexplored generalization of this idea is to test for inferences with entirely new (existentially bound) arguments. In the multivalent case above no new arguments are ever introduced (valency always decreases), but such inferences can sometimes be warranted. For example, in “John is publishing a documentary.” ⊨ “John has filmed a documentary” (and other examples (1) - (5) in Section 3.2.4), perhaps it is more accurate to expect models to predict the unary passive predicate *was_filmed(documentary)* along with the knowledge that *somebody* did the filming. We often make inferences about chains of events with multiple changing actors, and our models should reflect this. These inferences are akin to implicit semantic roles, and are thus also related to argument-based approaches to semantic representation such as QA-SRL (FitzGerald et al., 2018; Roit et al., 2020; Pyatkin et al., 2021a). Perhaps the right approach is to develop models that learn and represent inferences jointly between predicates and arguments (that is, entailments between predicates, entailments between arguments, and between those two categories when possible).

The latter two datasets in this thesis, *Sports* and ANT, have forced models to correctly identify paraphrases, antonyms and directional entailments and non-entailments.

Previous datasets such as Levy/Holt have mostly ignored this aspect, containing a high ratio of paraphrases. Again, models can appear more accurate than they really are under these evaluations, modeling many relations as symmetric. We believe that datasets would do well to mostly contain both directions of an inference, as this invites model development that is more attuned to this crucial property of entailment. Again, with this point we advocate for higher adversariality.

Of course, there are many other linguistic phenomena that deserve attention. Beyond the temporal entailments presented here, a more expansive dataset could contain modality (e.g. “It is likely that ... ” $\vDash$ “It is possible that ... ”) and propositional attitude. We can be inspired by the many categories studied under FraCas (Cooper et al., 1996), such as adjective intersectivity (e.g. “Max is a sick man and Max is a linguist.” $\vDash$ “Max is a sick linguist.”, but “Max is an alleged murderer and Max is a senator.” $\nvDash$ “Max is an alleged senator.” (Balcerak Jackson, 2017)), and introduce adversarial cases of monotonicity as modeled in Natural Logic (MacCartney and Manning, 2007). We can introduce metonomy (“Gas has gone up.” $\vDash$ “The price of gas has risen.”) or pragmatic phenomena such as implicatures. It is unclear whether a crowdsourcing strategy can be devised that guarantees all these particular phenomena are covered, even though any semantic model should be able to account for them. At the very least, the NLP community would benefit from a catalogue of inference types (with attention to the variety of inferences that humans can draw) and should be more cognizant of whether our datasets represent them.

The question remains, of course, of how to avoid data artifacts and biases, and prevent high-parameter supervised models from overfitting. We would like to avoid the bias of some datasets of priming data collection with automatic distributional methods (Berant et al., 2011; Zeichner et al., 2012), and at the same time avoid artefacts like hypothesis length or the presence of negation in the hypothesis being correlated with the label (Gururangan et al., 2018). ANLI (Nie et al., 2020) is on the right track. One appealing avenue might be to build a varied dataset reminiscent of FraCaS at a larger scale, with adversariality in mind and attention to data artefacts using humans in the loop.

Asking for variety and linguistically informed design stands in obvious contrast with other data creation preferences: low labor-intensiveness and low annotator training time, with the aim of driving down cost. Concessions on these points may be necessary. Dataset development and model development both play an important role in progress, but models have received an disproportionate share of monetary investment. For instance, pre-training the recent billion-parameter large language models is estimated to cost in the millions of dollars[1] (LambdaLabs, 2020), estimated with the cost of the Microsoft V100 GPUs that were used for training GPT-3. Why not expect the same for datasets? The ambitions we feel towards modeling should be extended to datasets — investing a similar level of resources could supply us with resources of enormous size, variety and quality.

8.2 Temporal Entailment and Causality

The challenges encountered in the temporal entailment task (Chapter 3) has pointed us to promising avenues of research. One of the main issues in the Tensed Entailment Graph approach was that a complicated set of morphosyntactic tense interactions needed to be learned. However, it is likely that these interactions are actually governed by a simpler set of relations between predicates: *precondition, consequence, hypernymy* and *paraphrase*, possibly among others. For example, *arrive* is a *precondition* of *visit*, and this determines the pattern of entailment between the different tenses of the predicates. Certainly this approach is at least more cognitively plausible than modeling the tensed interactions one-by-one.

As alluded to in Chapter 3, a possible line of work involves designing unsupervised methods to mine those lexical relations (e.g. *arrive* is a *precondition* of *visit*), for instance with an Entailment Graph model that incorporates temporal information. In parallel one could linguistically explore the entailment patterns that arise from lexical relations interacting with the various tense combinations, defining all these symbolic interactions manually (e.g. *p precondition of q* licenses an entailment *present progressive*(q) $\models$ *present perfect*(p), etc.). Finally, this information can be combined to correctly label premise-hypothesis pairs like *John is visiting London-John has arrived in London* as **True**. This method would offload the modeling demands of tense interactions to the manual design of a logical system, which alleviates the previously mentioned training data sparsity issues. The problem of designing these logical interactions and learning the lexical relations are likely both to be challenging tasks in themselves, however.

Again, valency plays an important part play here, so the resulting Entailment

[1] At 1287 MWh energy consumption for training (Patterson et al., 2021), the electrical bill alone would amount to hundreds of thousands — €185,000 using the average EU non-consumer household price per kWh of €0.1445 in the second semester of 2021 (EuropeanCommission, 2022).

Graphs should be multivalent (McKenna et al., 2021). This is because there may be preconditions and consequences of an eventuality that are particular to only a subset of the participants. Some of the most salient consequences of an eventuality only hold for a single participant — for example, "A kills B" has the *consequence* for B that "B is dead". "A harvests B" usually follows after the precondition that "B grows" or "B is ripe".

It may be possible to adapt the current temporal algorithm to this purpose. One path to investigate is splitting the temporal comparison window into a *before* and *after* frame, producing separate entailment scores for different temporal orderings. Perhaps the data's distribution is such that this corresponds to *precondition* and *consequence*, but this would need to be investigated in practice. A proof-theoretic approach with this background knowlege could then be tested on **TEA**[2].

At the same time, it will be useful to develop a deeper understanding of the relationship between entailment and causality. *Precondition* and *consequence* are both causal notions, separating them from ontological inferences like *car* ⊨ *vehicle* or *to run* ⊨ *to move*. While the latter have more to do with a hierarchy of types, the causal inferences are more related to how the world works — how *different* eventualities, which may combine to form larger episodes, are related to one another. These various inference types can exist side-by-side in the same entailment challenges, but our models should perhaps be aware of the distinctions. In particular, Pearl and Mackenzie (2018) have raised the concern that causal notions may not be learnable from a static dataset like a news corpus; in their terminology such a dataset is limited to the first, associative, rung on the ladder of causation. They argue that learning causation requires more dynamic interventions (rung two), and a notion of counterfactuality (rung three). Causality is a prominent part of the entailments in our NLI datasets, but it seems the causal revolution has yet to reach NLP.

8.3 The Temporal Algorithm

We presented a novel temporal algorithm for Entailment Graph induction in Chapter 4, and analyzed the conditions under which it works in Chapter 5. These investigations showed that temporality is a valuable learning signal for entailment. Still, there is room to further improve results and extend them to other domains, and our work has revealed

[2] Note that **TEA** could also be expanded with the temporal entailments we discovered among the WordNet antonyms.

some important directions for this purpose.

In the next section we again mention the algorithmic improvements suggested in 4. However, beyond these improvements, incorporating a phenomenon like temporality may also require input data that is highly accurate to begin with. Entailment Graph induction depends on numerous technologies that all still have their own multitude of problems to solve. This includes relation extraction, named entity recognition, typing and linking and coreference resolution, as well as the temporal relation extraction and modality detection systems themselves. The input to our algorithm is a set of tuples $p(a_1{:}t_1, a_2{:}t_2, [t_s, t_e])$; it contains predicates p, arguments a_n, types t_n, and time points t_s, t_e, which can be computed using a typical duration d. After Section 8.3.1 we therefore touch on each of these briefly.

Note that it will also be worth simply experimenting with a larger training data size. Temporality, along with the accuracy benefits, does introduce sparsity, and this may be offset with more data. In particular, it may also be worth expanding to a much larger temporal range, since the current NewsSpike experiments were limited to a 6-week time range. It is possible that the benefits of temporality are more evident across larger ranges, because here there is more opportunity for spurious temporal overlap, which the temporal algorithm can separate.

8.3.1 Algorithmic Design

In Section 4.5 we mentioned a few variations of the central algorithmic idea. The first idea is to improve on the fact that filtering becomes more challenging with more data. Future models could take the number of eventualities into account when computing overlap, so that overlaps become less guaranteed with a large amount of data. The second idea explores how temporality may be used for a directional signal *within* triple counts. We show this may be achievable by normalizing with the atemporal features instead of the temporal ones.

Another direction is to incorporate temporality into the (contextual) link prediction entailment scores proposed by Hosseini et al. (2019, 2021). One way of approaching this would be to alter the transition probabilities in the bipartite graph according to the temporal overlap between the relevant eventualities. Where Hosseini et al. (2019) define:

$$s_{r,q} = P(\langle q \rangle | \langle r \rangle) = \sum_{e_1, e_2 \in \mathcal{E}^2} P(\langle q \rangle | \langle e_1, e_2 \rangle) P(\langle e_1, e_2 \rangle | \langle r \rangle)$$

we could additionally multiply by a temporal overlap term $f_t((q,e_1,e_2),(r,e_1,e_2)) \in [0,1]$, that is low when the triples in question are temporally separable. For example,

$$f_t((win\_against, Manchester, Arsenal),(lose\_against, Manchester, Arsenal))$$

would be low, reducing the increment to the score $s_{win\_against,lose\_against}$. However, the additional temporal dimension may again lead to computational complexity issues for larger datasets, especially since the method depends on parallelizable matrix operations. This brings up a related point: future research would also benefit from ways of making the current algorithm more computationally efficient. This may be necessary with larger sets of input triples, both due to incorporated coreference resolution and larger training data sets.

Experimentation should become significantly easier with the pipeline improvements described in the following sections. Specifically, it simplifies teasing apart whether null results are due to design choices in the algorithm or due to input data, because the chance is reduced that a null result is caused by parts of the pipeline that were previously weak. For example, small differences in the general domain experiments could be due to various pipeline flaws. This is particularly true for temporal experiments because input noise can cause spurious overlaps, reducing the effect. With enough noise it may be possible for the temporal signal to be drowned out. A larger volume of higher quality data would therefore make it more straightforward to test variations of the features mentioned above, and achieve larger improvements.

8.3.2 Predicates

One issue in the representation of predicates is the current treatment of modifier verbs. The (non-modal) relation extraction system extracts both the modified and unmodified versions of predicates, which can lead to downstream mistakes. For example, for the sentence “Rosa Parks refused to comply with the law”, our system extracts both the predicates *comply_with* and *refuse_to_comply_with*. In this case, extracting the proposition containing the bare *comply_with* is incorrect, since it is not entailed by the sentence. On the other hand, if modified predicates are excluded entirely sparsity issues may be exacerbated. Understanding the semantics of modifier verbs would allow us to choose when to include the unmodified predicate, and would therefore be an important step towards more accurate relation extraction.

It may also be valuable to improve strategies around compound predicates such as *has_border_with* (extracted from the sentence “India has a border with China”). These can be constructed by including noun phrases in the predicate — in the example the argument *border* can be included because it forms a link between *India*, *has* and *China*. This construction is useful for light verb constructions such as *has a border with* or *takes care of*, and is helpful in covering multiword expressions. However, as with the modifiers, not all variations should always be extracted. The solution could be to extract only the compound predicate in the case of light verb constructions and multiword expressions, given a system for recognizing them. This would prevent a triple like *take(Alice, naps)*, that should be interpreted only using a compound predicate *take naps*, from being extracted and being used as evidence towards the entailments of the predicate *take(:person,:thing)* in the *person-thing* graph (causing issues, for example, if it co-occurs with *like(Alice, naps)*)).

On the other hand, there are cases where it may be better not to construct the compound predicate at all, such as when the introduced noun phrases are really arguments in their own right. For example, the current system extracts many predicates such as *want_$175-million_from()*, which may bloat the semantics with unnecessary predicates, especially when constructed with low cutoffs. What is needed is a stricter definition of what should constitute a predicate (e.g. based on whether it is a light verb, whether the argument belongs in the predicate, etc.), along with a way of classifying these categories automatically. Although current evaluations will not uncover this noise (since such predicates are never tested), it may prove inconvenient to downstream applications.

8.3.3 Arguments and Coreference

One path worth investigating on the argument side is how to differentiate between the signals provided by *general* entities (non-specific nouns like *plan*) and *named* entities (like *Obama*). The type-token distinction is relevant here: when general entities form part of an argument pair feature (say *(Obama,plan)*) they refer to a type (congregating different specific *plans* from each separate eventuality in the data), whereas *Obama* refers to the same token each time. Thus they constitute a different kind of feature than the dual named entity features typically used as examples (say *(Obama,Hawaii)* or *(Google,Youtube)*), and the consequences of their presence in the learning signal are unexplored. This investigation could also lead to better use of “GG” dual general entity pairs, which have thus far been ignored as a feature.

Relatedly, general entities often form part of larger noun phrases, and there is no clear strategy for selecting the most relevant subset of the larger phrase. For example, our data contains mention of both *plutonium plans* and *uranium plans*, but both are extracted with the generic *plans*. There is a tradeoff again here between accuracy and sparsity — in the current approach we risk incorrectly conflating tokens into the same type, but conversely we might risk splitting arguments into needlessly specific categories.

Failing to account for the more complex phrase can also risk missing the point of the sentence entirely. For example, in an article about cigarette use, our system extracted the triple *rule_on(United_States_Congress,use)*, whereas the more relevant noun phrase was *use_of_cigarettes*, or even simply *cigarettes*. The challenge is that it is highly ambiguous to decide whether an attached phrase (prepositional or otherwise) should form part of the argument (e.g. preferably ignoring *pocket* in *cigarettes in her pocket*), and whether the noun within is a pertinent one in its own right (e.g. perhaps instead extracting *perfume* from *smell of her perfume*).

The problems on both the predicate and the argument side reveal that learning semantics is something of a chicken-and-egg problem. We expect highly accurate relation extraction for learning a semantic representation, but simultaneously require a highly accurate semantic representation for approaching some of the most ubiquitous challenges in relation extraction.

We briefly discussed the importance of coreference resolution in Section 3.3.4.2. We mentioned a news article about Beyoncé, in which most of the predications in which she was involved did not refer to her by name, instead using various nominal references. Including this data could be essential to building stronger graphs. It would be interesting to pursue the hypothesis that coreference contains an essential learning signal for entailment. A straightforward experiment would be to compare the increase in performance when incorporating coreference data to the increase in performance with additional randomly sampled data. For example, one could build an Entailment Graph with 400K articles sampled from NewsSpike, and compare that baseline both to an Entailment Graph that adds the data of a robust coreference system and an Entailment Graph that samples triples from the remainder of NewsSpike to equalize the data increase.

8.3.4 Types

As mentioned in Section 3.3.4.2, argument pair types are the primary way of performing WSD in many Entailment Graph induction techniques (Berant et al., 2011; Hosseini et al., 2018), and this forms a major possible area of improvement. Firstly, the set of types chosen plays an important role in how the learning signal of different eventualities is spread across predicates. Our pipeline utilizes the first level of the FIGER hierarchy. One clear improvement might be to use a mixture of FIGER type levels. Some types in the first level are more general and common than others, and in those cases it might be worth prioritizing subcategories (e.g. beyond just *person*, using *person/politician*, *person/athlete*; or *organization/company*, *organization/sports_team*). Other first-level types are specific and exceedingly rare in our training and evaluation data, such as *astral_body* or *metropolitan_transit*, and could instead be modeled with the *thing* type. It may even be worthwhile simultaenously building graphs at multiple levels (e.g. both *person* and *person/politician*). so that *politician* information can still inform the *person* graph.

Again, there are benefits to be gained with general entities, which usually receive the *thing* type. Many general entities could be assigned more specific types, especially if context is used (for example, *person* for *the player*, *location* for *the stadium*, *organization* for *the club*). With an improved system for typing general entities we could therefore build stronger representations for all the non-*thing* type pair graphs, instead of siphoning information off to the more noisy *thing* graphs.

It will be a significant challenge to move beyond the more fundamental problem that argument types alone are insufficient for WSD. The ideal representation might involve a single graph for which the nodes and edges are less discrete, such that the entailments of a predicate can be adjusted given more context. Of course, this would require the basic assumptions of the model to be reevaluated, but the benefits in terms of increased representational power would be substantial.

8.3.5 Time and Duration

As suggested in Section 4.5, advances in temporal parsing are among the most crucial pipeline improvements to make in order to better understand the temporal signal. Results in Chapter 4 showed that the temporal expression data (the *timexOnly* condition) allowed for substantial precision gains, but was limited to low recall. It is essential to improve performance on this condition, since we can be certain that the document date information will often result in incorrect times attributed to an eventuality. Additionally, as with coreference, we may expect that more data from improved temporal parsing (i.e. including more eventualities from the same document) is better than simply increasing the dataset size.

Of course, one possibility is to use a more modern temporal parsing system. Instead of SUTime, the temporal expression component from CogCompTime (Ning et al., 2018c) could be used. Instead of linking the time expressions to predicates using dependency parse trees, a more modern temporal ordering system (e.g. (Chambers et al., 2014)) could be applied (which would not only connect temporal expressions and eventualities, but also order eventualities with respect to each other).

However, there are also appealing ideas for new systems. For example, it would be possible to develop more sophisticated ways of linking times to eventualities, expanding the current linking approach to work across the entire document. The approach could be inspired by the linguistic theory on tense, deixis and shifting Reichenbachian reference times (see Section 2.4.1.2). Furthermore, considering the similarity of temporal ordering to entity coreference resolution, it may be worth borrowing models from that field. For example, it may be possible to apply end-to-end neural coreference models (Lee et al., 2017) to the task. Memory networks (Weston et al., 2015; Sukhbaatar et al., 2015) for entity coreference (Liu et al., 2019a) could also be reworked into the temporal reference setting.

Event coreference could also be a crucial step in this procedure, allowing many eventualities that are not explicitly linked to a temporal expression to still be located in time. This can be useful inter-sententially within one document, but also between documents. For example, given the phrase *two days after the earthquake* a model that links the *earthquake* event to other eventualities in the document can also propagate the associated temporal information. When working between documents, it will be useful to have a general temporal background knowledge base, which can be invoked when a document refers to a generally known event, as in *a week before the first moon landing*. Likewise, a more specific knowledge base, that is updated dynamically with newsworthy events as a system reads articles, can provide essential temporal links. This is because articles in the news domain are written as part of a wider discourse, and as such may assume prior knowledge. For example, an article might state *2 days after Johnson's resignation*, expecting the reader to be aware of the date. Relatedly, it may be worth attempting to model event coreference and the temporal ordering task jointly.

The system would also benefit from better duration prediction models. Currently the model always defaults to TacoLM's prediction, but in some cases the exact duration is actually stated explicitly or can be inferred (for instance, "He played for the Yankees for 3 months"). Before improving the model, however, it may be worth guaranteeing that the evaluation dataset actually contains entailment pairs that benefit from varying the temporal window. ANT could be further analyzed, and if necessary extended to contain predicates like *be president*, which would benefit from a dynamic window.

An important related direction is to distinguish between temporal ranges and temporal durations. The former includes phrases like *the teams played in March*, in which eventualities of a short duration are described as occurring at some point within a larger window, while the latter actually assign a particular duration to an eventuality (e.g. *was working for Arsenal in March*). The temporal algorithm does not currently have a mechanism for taking this information into account.

8.3.6 Modality

Finally, with all these relation extraction improvements, the observed effect of modality may also change, perhaps extending into the general domain. For more research suggestions specific to the modal graphs, see 7.5.

Unfortunately, in preliminary experiments for combining temporality and modality, the temporal modal graphs were weaker than both the temporal graphs and the modal graphs. Again, this may be due to the data sparsity that accrues with both additions. However, with the cleaner signal from relation extraction improvements, modality may still provide a benefit to temporal graphs, since taking it into account should prevent additional spurious overlaps (e.g. from a simulatenous *might win* and *might lose*). A stronger modal tagger will also be useful in this case, perhaps moving towards a modality-aware language model similar to TacoLM.

8.4 Entailment and Grounding

We have already touched on the limitations of learning semantics from textual data alone, as seen in the challenge of learning causality from static data (Section 8.2). Lin (1998) also points out challenges associated with bootstrapping semantics from text: by example, the word "Westener" is used to refer to *hostages* the majority of the time in the 45 million word San Jose Mercury corpus. It is unclear how to prepare our models for this bias. Similar concerns have recently been raised by Bender and Koller (2020) — form alone is insufficient for learning meaning. These concerns seem to lead us to grounding: learning symbolic concepts by relating them to experience in various modalities, instead of through their relation to other symbolic concepts (as is the hope in language modeling).

Grounding in NLP is being explored in various forms. For example, the Visual Question Answering (VQA) task (Antol et al., 2015; Goyal et al., 2017) requires systems to jointly model images and text. Auditory (Kiela and Clark, 2017) and even olfactory (Kiela et al., 2015) modalities can improve semantic representations. Grounded language modeling is being explored with spontaneous speech data grounded in virtual reality environments (Ebert and Pavlick, 2020) — the type of data to which young children have access.

Future research will need to answer whether entailment knowledge stands to benefit from grounding as well. Ontological inferences such as *man* ⊨ *person* may be easier to learn through grounding, and temporal entailments may benefit from grounding too (e.g. using grounded visuospatial data to learn that *A drops B* ⊨ *B falls*). Still, some entailments seem taught directly through symbols; the fact that *bachelor* ⊨ *unmarried man* is probably not learned through any set-theoretic mechanism based in observation (i.e. observing that all grounded *bachelors* are also *unmarried men*), so some entailments may still require a purely symbolic signal. Likewise, these approaches should still take into account the contributions of innate knowledge, such as the knowledge that things can only be in one place at a time. This knowledge is passed down to humans through evolution rather than through grounded learning, so our models may likewise require a separate mechanism.

The intersection of grounding and entailment is only recently being explored. For example, Vu et al. (2018) show that models perform better on SNLI when they have access to visual data. Xie et al. (2019) go beyond this, proposing SNLI-VE, a *Visual Entailment* task in which the premise is an image and the hypothesis is a piece of text. Chen and Golisano (2021) introduce a similar task, moving from image data to video data. Purely textual data for learning entailment-based semantics is far from fully exploited, but the new grounding research direction may ultimately transport us beyond the inevitable limitations of textual data.

Bibliography

Adel, H. and Schütze, H. (2017). Exploring different dimensions of attention for uncertainty detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 22–34, Valencia, Spain. Association for Computational Linguistics.

Akbik, A. and Löser, A. (2012). KrakeN: N-ary facts in open information extraction. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 52–56, Montréal, Canada. Association for Computational Linguistics.

Allen, J. F. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843.

Angeli, G., Johnson Premkumar, M. J., and Manning, C. D. (2015). Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China. Association for Computational Linguistics.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., and Parikh, D. (2015). VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*.

Athanasiadou, A. and Dirven, R. (1997). Conditionality, hypotheticality, counterfactuality. *AMSTERDAM STUDIES IN THE THEORY AND HISTORY OF LINGUISTIC SCIENCE SERIES 4*, pages 61–96.

Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The Berkeley FrameNet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Baker, K., Bloodgood, M., Dorr, B., Filardo, N. W., Levin, L., and Piatko, C. (2010). A modality lexicon and its use in automatic tagging. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Balcerak Jackson, B. (2017). Structural entailment and semantic natural kinds. *Linguistics and Philosophy*, 40(3):207–237.

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. (2007). Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, page 2670–2676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Bar-Haim, R., Dagan, I., Greental, I., Szpektor, I., and Friedman, M. (2007). Semantic inference at the lexical-syntactic level for textual entailment recognition. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 131–136, Prague. Association for Computational Linguistics.

Ben Aharon, R., Szpektor, I., and Dagan, I. (2010). Generating entailment rules from FrameNet. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 241–246, Uppsala, Sweden. Association for Computational Linguistics.

Bender, E. M. and Koller, A. (2020). Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Bentivogli, L., Clark, P., Dagan, I., and Giampiccolo, D. (2010). The sixth pascal recognizing textual entailment challenge. In *Proceedings of the Third Text Analysis Conference, TAC 2010, Gaithersburg, Maryland, USA, November 15-16, 2010*. NIST.

Bentivogli, L., Clark, P., Dagan, I., and Giampiccolo, D. (2011). The seventh pascal recognizing textual entailment challenge. In *TAC*. Citeseer.

Berant, J. (2012). *Global Learning of Textual Entailment Graphs*. PhD thesis, Tel Aviv University.

Berant, J., Alon, N., Dagan, I., and Goldberger, J. (2015). Efficient global learning of entailment graphs. *Computational Linguistics*, 41(2):221–263.

Berant, J., Dagan, I., Adler, M., and Goldberger, J. (2012). Efficient tree-based approximation for entailment graph learning. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 117–125, Jeju Island, Korea. Association for Computational Linguistics.

Berant, J., Dagan, I., and Goldberger, J. (2010). Global learning of focused entailment graphs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1220–1229, Uppsala, Sweden. Association for Computational Linguistics.

Berant, J., Dagan, I., and Goldberger, J. (2011). Global learning of typed entailment rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 610–619, Portland, Oregon, USA. Association for Computational Linguistics.

Bernardy, J.-P. and Chatzikyriakidis, S. (2021). Applied temporal analysis: A complete run of the FraCaS test suite. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 11–20, Groningen, The Netherlands (online). Association for Computational Linguistics.

Bethard, S., Derczynski, L., Savova, G., Pustejovsky, J., and Verhagen, M. (2015). SemEval-2015 task 6: Clinical TempEval. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 806–814, Denver, Colorado. Association for Computational Linguistics.

Bethard, S. and Parker, J. (2016). A semantically compositional annotation scheme for time normalization. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3779–3786, Portorož, Slovenia. European Language Resources Association (ELRA).

Bethard, S., Savova, G., Chen, W.-T., Derczynski, L., Pustejovsky, J., and Verhagen, M. (2016). SemEval-2016 task 12: Clinical TempEval. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1052–1062, San Diego, California. Association for Computational Linguistics.

Bethard, S., Savova, G., Palmer, M., and Pustejovsky, J. (2017). SemEval-2017 task 12: Clinical TempEval. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 565–572, Vancouver, Canada. Association for Computational Linguistics.

Biber, D. and Finegan, E. (1989). Styles of stance in english: Lexical and grammatical marking of evidentiality and affect. *Text-interdisciplinary journal for the study of discourse*, 9(1):93–124.

Bijl de Vroe, S., Guillou, L., Johnson, M., and Steedman, M. (2022). Temporality in general-domain entailment graph induction. Under Review.

Bijl de Vroe, S., Guillou, L., Stanojević, M., McKenna, N., and Steedman, M. (2021). Modality and negation in event extraction. In *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021)*, pages 31–42, Online. Association for Computational Linguistics.

Bittar, A., Amsili, P., Denis, P., and Danlos, L. (2011). French TimeBank: An ISO-TimeML annotated reference corpus. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 130–134, Portland, Oregon, USA. Association for Computational Linguistics.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Bonyadi, A. (2011). Linguistic manifestations of modality in newspaper. *International Journal of Linguistics*, 3(1):E30.

Bos, J. (2014). Is there a place for logic in recognizing textual entailment. In *Linguistic Issues in Language Technology, Volume 9, 2014 - Perspectives on Semantic Representations for Textual Inference*. CSLI Publications.

Bos, J., Clark, S., Steedman, M., Curran, J. R., and Hockenmaier, J. (2004). Wide-coverage semantic representations from a CCG parser. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 1240–1246, Geneva, Switzerland. COLING.

Bos, J. and Markert, K. (2005). Recognising textual entailment with logical inference. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 628–635, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Celikyilmaz, A., and Choi, Y. (2019). COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. (1993). Signature verification using a ”siamese” time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS’93, page 737–744, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramèr, F., and Zhang, C. (2022). Quantifying memorization across neural language models. *ArXiv*, abs/2202.07646.

Carnap, R. (1952). Meaning postulates. *Philosophical studies*, 3(5):65–73.

Cassidy, T., McDowell, B., Chambers, N., and Bethard, S. (2014). An annotation framework for dense event ordering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 501–506, Baltimore, Maryland. Association for Computational Linguistics.

Chambers, N., Cassidy, T., McDowell, B., and Bethard, S. (2014). Dense event ordering with a multi-pass architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284.

Chang, A. X. and Manning, C. (2012). SUTime: A library for recognizing and normalizing time expressions. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*, pages 3735–3740, Istanbul, Turkey. European Language Resources Association (ELRA).

Chatzikyriakidis, S. and Luo, Z. (2014). Natural language inference in coq. *Journal of Logic, Language and Information*, 23(4):441–480.

Chen, J. and Golisano, Y. K. (2021). Explainable video entailment with grounded visual evidence. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2001–2010.

Chen, Z., Feng, Y., and Zhao, D. (2022). Entailment graph learning with textual entailment and soft transitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5899–5910, Dublin, Ireland. Association for Computational Linguistics.

Chklovski, T. and Pantel, P. (2004). Verbocean: Mining the web for fine-grained semantic verb relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*.

Claessen, K. and Sörensson, N. (2003). New techniques that improve mace-style finite model finding. In *Proceedings of the CADE-19 Workshop: Model Computation - Principles, Algorithms, Applications*.

Clark, S. and Curran, J. R. (2007). Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

Clark, S., Hockenmaier, J., and Steedman, M. (2002). Building deep dependency structures using a wide-coverage CCG parser. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 327–334, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Comrie, B. (1985). *Tense*, volume 17. Cambridge university press.

Conneau, A., Rinott, R., Lample, G., Williams, A., Bowman, S., Schwenk, H., and Stoyanov, V. (2018). XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Cooper, R., Crouch, D., Eijck, J. V., Fox, C., Genabith, J. V., Jaspars, J., Kamp, H., Milward, D., Pinkal, M., Poesio, M., Pulman, S., Briscoe, T., Maier, H., and Konrad, K. (1996). Using the framework. FraCaS: A framework for computational semantics. Technical report, Technical Report LRE 62-051 D-16, The FraCaS Consortium.

Coq (2004). *The Coq proof assistant reference manual by the Coq Development Team*. LogiCal Project. Version 8.0.

Dagan, I., Glickman, O., and Magnini, B. (2006). The pascal recognising textual entailment challenge. In *In Quiñonero-Candela et al., editors, MLCW 2005*, volume LNAI Volume 3944, pages 177–190. Springer.

Dagan, I., Lee, L., and Pereira, F. C. N. (1999). Similarity-based models of word cooccurrence probabilities. *Machine Learning*, 34:43–69.

Dancygier, B. (1998). *Conditionals and Prediction: Time, Knowledge and Causation in Conditional Constructions*, volume 87 of *Cambridge Studies in Linguistics*. Cambridge University Press.

Davidson, D. (1967). The logical form of action sentences. In Rescher, N., editor, *The Logic of Decision and Action*, pages 81–95. University of Pittsburgh Press.

De Marneffe, M.-C., Simons, M., and Tonhauser, J. (2019). The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pages 107–124.

Declerck, R. (1986). From reichenbach (1947) to comrie (1985) and beyond: Towards a theory of tense. *Lingua*, 70(4):305–364.

Declerck, R., Reed, S., and Cappelle, B. (2006). *The grammar of the English tense system: a comprehensive analysis*, volume 1. Walter de Gruyter.

Del Corro, L. and Gemulla, R. (2013). Clausie: Clause-based open information extraction. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, page 355–366, New York, NY, USA. Association for Computing Machinery.

Deo, A. (2012). Morphology. in the oxford handbook of tense and aspect ed. robert i. binnik, 155-183.

Depraetere, I. (1998). On the resultative character of present perfect sentences. *Journal of pragmatics*, 29(5):597–613.

Dettmers, T., Minervini, P., Stenetorp, P., and Riedel, S. (2018). Convolutional 2d knowledge graph embeddings.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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, Minneapolis, Minnesota. Association for Computational Linguistics.

Dligach, D., Miller, T., Lin, C., Bethard, S., and Savova, G. (2017). Neural temporal relation extraction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 746–751, Valencia, Spain. Association for Computational Linguistics.

Dowty, D. R. (1979). Word meaning and montague grammar (synthese language library 7). dordrecht: D.

Ebert, D. and Pavlick, E. (2020). A visuospatial dataset for naturalistic verb learning. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 143–153, Barcelona, Spain (Online). Association for Computational Linguistics.

Efron, B. and Tibshirani, R. (1985). The bootstrap method for assessing statistical accuracy. *Behaviormetrika*, 12(17):1–35.

Efron, B. and Tibshirani, R. J. (1994). *An introduction to the bootstrap*. CRC press.

Eichler, K., Gabryszak, A., and Neumann, G. (2014). An analysis of textual inference in German customer emails. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (*SEM 2014)*, pages 69–74, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Eichler, K., Xu, F., Uszkoreit, H., Hennig, L., and Krause, S. (2016). TEG-REP: A corpus of textual entailment graphs based on relation extraction patterns. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*

(LREC'16), pages 3367–3372, Portorož, Slovenia. European Language Resources Association (ELRA).

Eichler, K., Xu, F., Uszkoreit, H., and Krause, S. (2017). Generating pattern-based entailment graphs for relation extraction. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 220–229, Vancouver, Canada. Association for Computational Linguistics.

EuropeanCommission (2022). European commission, eurostat electricity price statistics. `https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Electricity_price_statistics#Electricity_prices_for_non-household_consumers`. Accessed: 2022-06-15.

Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Farkas, R., Vincze, V., Móra, G., Csirik, J., and Szarvas, G. (2010). The conll-2010 shared task: learning to detect hedges and their scope in natural language text. In *Proceedings of the fourteenth conference on computational natural language learning–Shared task*, pages 1–12.

Fay, R., editor (1990). *Collins Cobuild English Grammar*. Collins, United Kingdom.

Fedus, W., Zoph, B., and Shazeer, N. (2021). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961*.

Firth, J. R. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*.

FitzGerald, N., Michael, J., He, L., and Zettlemoyer, L. (2018). Large-scale QA-SRL parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2051–2060, Melbourne, Australia. Association for Computational Linguistics.

Freksa, C. (1992). Temporal reasoning based on semi-intervals. *Artificial intelligence*, 54(1-2):199–227.

Ganitkevitch, J., Van Durme, B., and Callison-Burch, C. (2013). PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, Atlanta, Georgia. Association for Computational Linguistics.

Geffet, M. and Dagan, I. (2005). The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 107–114, Ann Arbor, Michigan. Association for Computational Linguistics.

Glickman, O., Dagan, I., and Koppel, M. (2005a). A probabilistic classification approach for lexical textual entailment. In *AAAI*.

Glickman, O., Dagan, I., and Koppel, M. (2005b). Web based probabilistic textual entailment. In *Proceedings of the 1st Pascal Challenge Workshop*, pages 33–36. Citeseer.

Glockner, M., Shwartz, V., and Goldberg, Y. (2018). Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Goel, P., Prabhu, S., Debnath, A., Modi, P., and Shrivastava, M. (2020). Hindi Time-Bank: An ISO-TimeML annotated reference corpus. In *16th Joint ACL - ISO Workshop on Interoperable Semantic Annotation PROCEEDINGS*, pages 13–21, Marseille. European Language Resources Association.

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., and Parikh, D. (2017). Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Graves, A., Fernández, S., and Schmidhuber, J. (2005). Bidirectional lstm networks for improved phoneme classification and recognition. pages 799–804.

Guan, H., Li, J., Xu, H., and Devarakonda, M. (2021). Robustly pre-trained neural model for direct temporal relation extraction. In *2021 IEEE 9th International Conference on Healthcare Informatics (ICHI)*, pages 501–502.

Guillou, L., Bijl de Vroe, S., Hosseini, M. J., Johnson, M., and Steedman, M. (2020). Incorporating temporal information in entailment graph mining. In *Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs)*, pages 60–71, Barcelona, Spain (Online). Association for Computational Linguistics.

Guillou, L., Bijl de Vroe, S., Johnson, M., and Steedman, M. (2021). Blindness to modality helps entailment graph mining. In *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pages 110–116, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S., and Smith, N. A. (2018). Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Hacquard, V. (2006). *Aspects of modality*. PhD thesis, Massachusetts Institute of Technology.

Hamm, F. and Bott, O. (2018). Tense and aspect. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer 2018 edition.

Harmeling, S. (2007). An extensible probabilistic transformation-based approach to the third recognizing textual entailment challenge. In *TextEntail 2007*, pages 137–142. Max-Planck-Gesellschaft.

Hickl, A. (2008). Using discourse commitments to recognize textual entailment. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 337–344, Manchester, UK. Coling 2008 Organizing Committee.

Hinrichs, E. (1986). Temporal anaphora in discourses of english. *Linguistics and Philosophy*, 9(1):63–82.

Hintikka, K. J. J. (1962). *Knowledge and Belief: An Introduction to the Logic of the Two Notions*. Ithaca, NY, USA: Cornell University Press.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hockenmaier, J. and Steedman, M. (2007a). CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Hockenmaier, J. and Steedman, M. (2007b). CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., and Weikum, G. (2011). Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Holt, X. (2018). Probabilistic models of relational implication. Master's thesis, Macquarie University.

Horn, L. (1989). *A Natural History of Negation*. University of Chicago Press.

Hosseini, M. J. (2021). *Unsupervised Learning of Relational Entailment Graphs from Text*. PhD thesis.

Hosseini, M. J., Chambers, N., Reddy, S., Holt, X. R., Cohen, S. B., Johnson, M., and Steedman, M. (2018). Learning typed entailment graphs with global soft constraints. *Transactions of the Association for Computational Linguistics*, 6:703–717.

Hosseini, M. J., Cohen, S. B., Johnson, M., and Steedman, M. (2019). Duality of link prediction and entailment graph induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4736–4746, Florence, Italy. Association for Computational Linguistics.

Hosseini, M. J., Cohen, S. B., Johnson, M., and Steedman, M. (2021). Open-domain contextual link prediction and its complementarity with entailment graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2790–2802, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Huang, L., Sun, C., Qiu, X., and Huang, X. (2019). GlossBERT: BERT for word sense disambiguation with gloss knowledge. 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 3509–3514, Hong Kong, China. Association for Computational Linguistics.

Hwang, C. H. and Schubert, L. K. (1993). Interpreting temporal adverbials. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Hwang, C. H. and Schubert, L. K. (1994). Interpreting tense, aspect and time adverbials: A compositional, unified approach. In Gabbay, D. M. and Ohlbach, H. J., editors, *Temporal Logic*, pages 238–264, Berlin, Heidelberg. Springer Berlin Heidelberg.

Iftene, A. and Balahur-Dobrescu, A. (2007). Hypothesis transformation and semantic variability rules used in recognizing textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 125–130, Prague. Association for Computational Linguistics.

Jean, P.-A., Harispe, S., Ranwez, S., Bellot, P., and Montmain, J. (2016). Uncertainty detection in natural language: A probabilistic model. In *Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics*, pages 1–10.

Jeong, Y.-S., Joo, W.-T., Do, H.-W., Lim, C.-G., Choi, K.-S., and Choi, H.-J. (2016). Korean TimeML and Korean TimeBank. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 356–359, Portorož, Slovenia. European Language Resources Association (ELRA).

Jijkoun, V. and de Rijke, M. (2005). Recognizing textual entailment using lexical similarity.

Jordan, P. W. et al. (1994). Determining the temporal ordering of events in discourse. *Unpublished masters thesis for Carnegie Mellon Computational Linguistics Program*.

Kamp, H. and Reyle, U. (1993). *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Dordrecht: Kluwer Academic Publishers.

Karttunen, L. and Zaenen, A. (2005). Veridicity. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Katz, J. J. (1972). *Semantic Theory*. New York: Harper & Row.

Kiela, D., Bulat, L., and Clark, S. (2015). Grounding semantics in olfactory perception. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 231–236, Beijing, China. Association for Computational Linguistics.

Kiela, D. and Clark, S. (2017). Learning neural audio embeddings for grounding semantics in auditory perception. *Journal of Artificial Intelligence Research*, 60:1003–1030.

Kilicoglu, H. and Bergler, S. (2008). Recognizing speculative language in biomedical research articles: a linguistically motivated perspective. *BMC bioinformatics*, 9(11):1–10.

Kim, J.-D., Ohta, T., and Tsujii, J. (2008). Corpus annotation for mining biomedical events from literature. *BMC bioinformatics*, 9(1):1–25.

Kober, T., Alikhani, M., Stone, M., and Steedman, M. (2020). Aspectuality across genre: A distributional semantics approach. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4546–4562, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Kober, T., Bijl de Vroe, S., and Steedman, M. (2019). Temporal and aspectual entailment. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 103–119, Gothenburg, Sweden. Association for Computational Linguistics.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Kotlerman, L., Dagan, I., Magnini, B., and Bentivogli, L. (2015). Textual entailment graphs. *Natural Language Engineering*, 21(5):699–724.

Kouylekov, M. and Magnini, B. (2005). Recognizing textual entailment with tree edit distance algorithms. In *PASCAL Challenges on RTE*, pages 17–20.

Kowalski, R. and Sergot, M. (1989). A logic-based calculus of events. In *Foundations of knowledge base management*, pages 23–55. Springer.

Kozareva, Z. and Montoyo, A. (2006). The role and resolution of textual entailment in natural language processing applications. In *International Conference on Application of Natural Language to Information Systems*, pages 186–196. Springer.

Kratzer, A. (1981). Partition and revision: The semantics of counterfactuals. *Journal of Philosophical Logic*, 10(2):201–216.

Kripke, S. A. (1963). Semantical considerations on modal logic. *Acta Philosophica Fennica*, 16:83–94.

LambdaLabs (2020). Openai's gpt-3 language model: A technical overview. `https://lambdalabs.com/blog/demystifying-gpt-3/`. Accessed: 2022-06-15.

Lana-Serrano, S., Sánchez-Cisneros, D., Fernández, P. M., Moreno-Sandoval, A., and Llanos, L. C. (2012). An approach for detecting modality and negation in texts by using rule-based techniques. In Forner, P., Karlgren, J., and Womser-Hacker, C., editors, *CLEF 2012 Evaluation Labs and Workshop, Online Working Notes, Rome, Italy, September 17-20, 2012*, volume 1178 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Lassiter, D. (2011). *Measurement and modality: The scalar basis of modal semantics*. Ph. D. thesis, New York University.

Lassiter, D. (2017). *Graded modality: Qualitative and quantitative perspectives*. Oxford University Press.

Lassiter, D. and Goodman, N. D. (2015). How many kinds of reasoning? inference, probability, and natural language semantics. *Cognition*, 136:123–134.

Lee, K., Artzi, Y., Choi, Y., and Zettlemoyer, L. (2015). Event detection and factuality assessment with non-expert supervision. In *Proceedings of the 2015 Conference* *on Empirical Methods in Natural Language Processing*, pages 1643–1648, Lisbon, Portugal. Association for Computational Linguistics.

Lee, K., Artzi, Y., Dodge, J., and Zettlemoyer, L. (2014). Context-dependent semantic parsing for time expressions. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447, Baltimore, Maryland. Association for Computational Linguistics.

Lee, K., He, L., Lewis, M., and Zettlemoyer, L. (2017). End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Leeuwenberg, A. and Moens, M.-F. (2019). A survey on temporal reasoning for temporal information extraction from text. *Journal of Artificial Intelligence Research*, 66:341–380.

Levy, O. and Dagan, I. (2016). Annotating relation inference in context via question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 249–255, Berlin, Germany. Association for Computational Linguistics.

Levy, O., Dagan, I., and Goldberger, J. (2014). Focused entailment graphs for open IE propositions. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 87–97, Ann Arbor, Michigan. Association for Computational Linguistics.

Lewis, D. (1973). Counterfactuals and comparative possibility. *Journal of Philosophical Logic*, pages 418–446.

Li, T., Hosseini, M. J., Weber, S., and Steedman, M. (2022). Language models are poor learners of directional inference.

Lin, C., Miller, T., Dligach, D., Amiri, H., Bethard, S., and Savova, G. (2018). Self-training improves recurrent neural networks performance for temporal relation extraction. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, pages 165–176, Brussels, Belgium. Association for Computational Linguistics.

Lin, C., Miller, T., Dligach, D., Bethard, S., and Savova, G. (2019). A BERT-based universal model for both within- and cross-sentence clinical temporal relation extraction. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 65–71, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2*, pages 768–774, Montreal, Quebec, Canada. Association for Computational Linguistics.

Ling, X. and Weld, D. S. (2012). Fine-grained entity recognition. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, page 94–100. AAAI Press.

Liu, F., Zettlemoyer, L., and Eisenstein, J. (2019a). The referential reader: A recurrent entity network for anaphora resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5918–5925, Florence, Italy. Association for Computational Linguistics.

Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., and Wang, P. (2020). K-bert: Enabling language representation with knowledge graph. In *AAAI*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019b). Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Luo, Z. (2012). Formal semantics in modern type theories with coercive subtyping. *Linguistics and Philosophy*, 35.

Lyons, J. (1977). *Deixis, space and time*, volume 2, page 636–724. Cambridge University Press.

MacCartney, B., Grenager, T., de Marneffe, M.-C., Cer, D., and Manning, C. D. (2006). Learning to recognize features of valid textual entailments. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 41–48, New York City, USA. Association for Computational Linguistics.

MacCartney, B. and Manning, C. D. (2007). Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200, Prague. Association for Computational Linguistics.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Manning, C. D. (2006). Local textual inference: it's hard to circumscribe, but you know it when you see it–and nlp needs it.

Mausam, Schmitz, M., Soderland, S., Bart, R., and Etzioni, O. (2012). Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island, Korea. Association for Computational Linguistics.

Mazur, P. and Dale, R. (2010). WikiWars: A new corpus for research on temporal expressions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 913–922, Cambridge, MA. Association for Computational Linguistics.

McCarthy, J. and Hayes, P. J. (1969). Some philosophical problems from the standpoint of artificial intelligence. In *Readings in artificial intelligence*, pages 431–450. Elsevier.

McCawley, J. D. (1971). Tense and time reference in english. In Fillmore, C. J. and Langėndoen, D. T., editors, *Studies in Linguistic Semantics*, pages 96–113. Irvington.

McCoy, R. T., Smolensky, P., Linzen, T., Gao, J., and Celikyilmaz, A. (2021). How much do language models copy from their training data? evaluating linguistic novelty in text generation using raven.

McCready, E. and Ogata, N. (2007). Evidentiality, modality and probability. *Linguistics and Philosophy*, 30(2):147–206.

McDowell, B., Chambers, N., Ororbia II, A., and Reitter, D. (2017). Event ordering with a generalized model for sieve prediction ranking. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 843–853, Taipei, Taiwan. Asian Federation of Natural Language Processing.

McKay, T. and Nelson, M. (2000). Propositional attitude reports.

McKenna, N., Guillou, L., Hosseini, M. J., Bijl de Vroe, S., Johnson, M., and Steedman, M. (2021). Multivalent entailment graphs for question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10758–10768, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

McKenna, N. and Steedman, M. (2020). Learning negation scope from syntactic structure. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 137–142.

McKenna, N. and Steedman, M. (2022). Smoothing entailment graphs with language models.

Medlock, B. and Briscoe, T. (2007). Weakly supervised learning for hedge classification in scientific literature. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 992–999, Prague, Czech Republic. Association for Computational Linguistics.

Merriam-Webster (2021). Merriam-webster online thesaurus. `https://www.merriam-webster.com/thesaurus`. Accessed: 2021-12-16.

Mesquita, F., Schmidek, J., and Barbosa, D. (2013). Effectiveness and efficiency of open relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 447–457, Seattle, Washington, USA. Association for Computational Linguistics.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Miller, G. A. (1993). WORDNET: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Minard, A.-L., Speranza, M., Urizar, R., Altuna, B., van Erp, M., Schoen, A., and van Son, C. (2016). MEANTIME, the NewsReader multilingual event and time corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4417–4422, Portorož, Slovenia. European Language Resources Association (ELRA).

Mirza, P. and Tonelli, S. (2016). CATENA: CAusal and TEmporal relation extraction from NAtural language texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 64–75, Osaka, Japan. The COLING 2016 Organizing Committee.

Moens, M. and Steedman, M. (1988). Temporal ontology and temporal reference. *Computational linguistics*, 14(2):15–28.

Mohammad, S. M., Dorr, B. J., Hirst, G., and Turney, P. D. (2013). Computing lexical contrast. *Computational Linguistics*, 39(3):555–590.

Montague, R. (1970). Universal grammar. *Theoria*, 36(3):373–398.

Monz, C. and de Rijke, M. (2001). Light-weight entailment checking for computational semantics. In *In Proc. of the 3 rd Workshop on Inference in Computational Semantics*.

Morante, R. and Daelemans, W. (2009). Learning the scope of hedge cues in biomedical texts. In *Proceedings of the BioNLP 2009 workshop*, pages 28–36.

Morante, R. and Daelemans, W. (2012). Annotating modality and negation for a machine reading evaluation. In Forner, P., Karlgren, J., and Womser-Hacker, C., editors, *CLEF 2012 Evaluation Labs and Workshop, Online Working Notes, Rome, Italy, September 17-20, 2012*, volume 1178 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Mostafazadeh, N., Grealish, A., Chambers, N., Allen, J., and Vanderwende, L. (2016). CaTeRS: Causal and temporal relation scheme for semantic annotation of event structures. In *Proceedings of the Fourth Workshop on Events*, pages 51–61, San Diego, California. Association for Computational Linguistics.

Napoles, C., Gormley, M., and Van Durme, B. (2012). Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 95–100, Montréal, Canada. Association for Computational Linguistics.

Nguyen, D. B., Hoffart, J., Theobald, M., and Weikum, G. (2014). Aida-light: High-throughput named-entity disambiguation. *Workshop on Linked Data on the Web*, 1184:1–10.

Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., and Kiela, D. (2020). Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Niklaus, C., Cetto, M., Freitas, A., and Handschuh, S. (2018). A survey on open information extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3866–3878, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ning, Q., Subramanian, S., and Roth, D. (2019). An improved neural baseline for temporal relation extraction. 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 6203–6209, Hong Kong, China. Association for Computational Linguistics.

Ning, Q., Wu, H., Peng, H., and Roth, D. (2018a). Improving temporal relation extraction with a globally acquired statistical resource. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 841–851, New Orleans, Louisiana. Association for Computational Linguistics.

Ning, Q., Wu, H., and Roth, D. (2018b). A multi-axis annotation scheme for event temporal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1318–1328, Melbourne, Australia. Association for Computational Linguistics.

Ning, Q., Zhou, B., Feng, Z., Peng, H., and Roth, D. (2018c). CogCompTime: A tool for understanding time in natural language. In *Proceedings of the 2018 Conference* *on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 72–77, Brussels, Belgium. Association for Computational Linguistics.

Olex, A. L. and McInnes, B. T. (2021). Review of temporal reasoning in the clinical domain for timeline extraction: Where we are and where we need to be. *Journal of Biomedical Informatics*, 118:103784.

Pakray, P., Bhaskar, P., Banerjee, S., Bandyopadhyay, S., and Gelbukh, A. F. (2012). An automatic system for modality and negation detection. In Forner, P., Karlgren, J., and Womser-Hacker, C., editors, *CLEF 2012 Evaluation Labs and Workshop, Online Working Notes, Rome, Italy, September 17-20, 2012*, volume 1178 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Pan, F., Mulkar, R., and Hobbs, J. R. (2006). An annotated corpus of typical durations of events. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Parikh, A., Täckström, O., Das, D., and Uszkoreit, J. (2016). A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas. Association for Computational Linguistics.

Parsons, T. (1990). *Events in the Semantics of English: A Study in Subatomic Semantics*. MIT Press.

Partee, B. H. (1973). Some structural analogies between tenses and pronouns in english. *The Journal of Philosophy*, 70(18):601–609.

Partee, B. H. (1984). Nominal and temporal anaphora. *Linguistics and Philosophy*, 7(3):243–286.

Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L.-M., Rothchild, D., So, D., Texier, M., and Dean, J. (2021). Carbon emissions and large neural network training.

Pavlick, E. and Kwiatkowski, T. (2019). Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Pavlick, E., Rastogi, P., Ganitkevitch, J., Van Durme, B., and Callison-Burch, C. (2015). PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430, Beijing, China. Association for Computational Linguistics.

Pearl, J. and Mackenzie, D. (2018). *The Book of Why*. Basic Books, New York.

Peñas, A., Hovy, E. H., Forner, P., Rodrigo, Á., Sutcliffe, R. F., Forascu, C., and Sporleder, C. (2011). Overview of qa4mre at clef 2011: Question answering for machine reading evaluation. In *CLEF (Notebook Papers/Labs/Workshop)*, pages 1–20. Citeseer.

Peng, Y., Wang, X., Lu, L., Bagheri, M., Summers, R., and Lu, Z. (2018). Negbio: a high-performance tool for negation and uncertainty detection in radiology reports. *AMIA Summits on Translational Science Proceedings*, 2018:188.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Poliak, A. (2020). A survey on recognizing textual entailment as an NLP evaluation. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 92–109, Online. Association for Computational Linguistics.

Poliak, A., Haldar, A., Rudinger, R., Hu, J. E., Pavlick, E., White, A. S., and Van Durme, B. (2018). Collecting diverse natural language inference problems for sentence representation evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 67–81, Brussels, Belgium. Association for Computational Linguistics.

Prabhakaran, V., Bloodgood, M., Diab, M., Dorr, B., Levin, L., Piatko, C. D., Rambow, O., and Van Durme, B. (2012). Statistical modality tagging from rule-based annotations and crowdsourcing. In *Proceedings of the Workshop on Extra-Propositional Aspects of Meaning in Computational Linguistics*, pages 57–64, Jeju, Republic of Korea. Association for Computational Linguistics.

Prior, A. N. (1967). *Past, present and future*, volume 154. Clarendon Press Oxford.

Pustejovsky, J., Castano, J. M., Ingria, R., Sauri, R., Gaizauskas, R. J., Setzer, A., Katz, G., and Radev, D. R. (2003a). Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34.

Pustejovsky, J., Hanks, P., Sauri, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., et al. (2003b). The timebank corpus. In *Corpus linguistics*, volume 2003, page 40. Lancaster, UK.

Pustejovsky, J., Lee, K., Bunt, H., and Romary, L. (2010). ISO-TimeML: An international standard for semantic annotation. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Pyatkin, V., Roit, P., Michael, J., Goldberg, Y., Tsarfaty, R., and Dagan, I. (2021a). Asking it all: Generating contextualized questions for any semantic role. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1429–1441, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pyatkin, V., Sadde, S., Rubinstein, A., Portner, P., and Tsarfaty, R. (2021b). The possible, the plausible, and the desirable: Event-based modality detection for language processing. 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 953–965, Online. Association for Computational Linguistics.

Radford, A. and Narasimhan, K. (2018). Improving language understanding by generative pre-training.

Rajana, S., Callison-Burch, C., Apidianaki, M., and Shwartz, V. (2017). Learning antonyms with paraphrases and a morphology-aware neural network. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 12–21, Vancouver, Canada. Association for Computational Linguistics.

Reddy, S., Lapata, M., and Steedman, M. (2014). Large-scale semantic parsing without question-answer pairs. *Transactions of the Association for Computational Linguistics*, 2:377–392.

Rei, M. and Briscoe, T. (2010). Combining manual rules and supervised learning for hedge cue and scope detection. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning–Shared Task*, pages 56–63.

Reichenbach, H. (1947). The tenses of verbs. *Time: From Concept to Narrative Construct: a Reader*.

RelatedWords (2021). Relatedwords.org website. `https://www.relatedwords.org/`. Accessed: 2021-12-16.

Riazanov, A. and Voronkov, A. (2002). The design and implementation of vampire. *AI Commun.*, 15(2,3):91–110.

Rimell, L. and Clark, S. (2008). Adapting a lexicalized-grammar parser to contrasting domains. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 475–484, Honolulu, Hawaii. Association for Computational Linguistics.

Rocktäschel, T., Grefenstette, E., Hermann, K. M., Kočiskỳ, T., and Blunsom, P. (2015). Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*.

Rogers, A., Smelkov, G., and Rumshisky, A. (2019). Narrativetime: Dense high-speed temporal annotation on a timeline. *ArXiv*, abs/1908.11443.

Roit, P., Klein, A., Stepanov, D., Mamou, J., Michael, J., Stanovsky, G., Zettlemoyer, L., and Dagan, I. (2020). Controlled crowdsourcing for high-quality QA-SRL annotation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7008–7013, Online. Association for Computational Linguistics.

Rosch, E. and Mervis, C. B. (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, 7(4):573–605.

Rosch, E., Mervis, C. B., Gray, W. D., Johnson, D. M., and Boyes-Braem, P. (1976). Basic objects in natural categories. *Cognitive Psychology*, 8(3):382–439.

Rosenberg, S., Kilicoglu, H., and Bergler, S. (2012). CLaC Labs: Processing modality and negation. working notes for QA4MRE pilot task at CLEF 2012. In

Forner, P., Karlgren, J., and Womser-Hacker, C., editors, *CLEF (Online Working Notes/Labs/Workshop)*, volume 1178 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Rudinger, R., White, A. S., and Van Durme, B. (2018). Neural models of factuality. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 731–744, New Orleans, Louisiana. Association for Computational Linguistics.

Santus, E., Yung, F., Lenci, A., and Huang, C.-R. (2015). EVALution 1.0: an evolving semantic dataset for training and evaluation of distributional semantic models. In *Proceedings of the 4th Workshop on Linked Data in Linguistics: Resources and Applications*, pages 64–69, Beijing, China. Association for Computational Linguistics.

Saurı, R., Verhagen, M., and Pustejovsky, J. (2006). Annotating and recognizing event modality in text. In *Proceedings of 19th International FLAIRS Conference*.

Saurí, R. and Pustejovsky, J. (2009). Factbank: A corpus annotated with event factuality. *Language Resources and Evaluation*, 43:227–268.

Schmitt, M. and Schütze, H. (2019). SherLIiC: A typed event-focused lexical inference benchmark for evaluating natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 902–914, Florence, Italy. Association for Computational Linguistics.

Schoenmackers, S., Davis, J., Etzioni, O., and Weld, D. (2010). Learning first-order horn clauses from web text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1088–1098, Cambridge, MA. Association for Computational Linguistics.

Schuler, K. K. (2005). *Verbnet: A Broad-coverage, Comprehensive Verb Lexicon*. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA. AAI3179808.

Shapiro, S. (2005). Logical consequence, proof theory, and model theory. In Shapiro, S., editor, *Oxford Handbook of Philosophy of Mathematics and Logic*, pages 651–670. Oxford University Press.

Shinyama, Y., Sekine, S., and Sudo, K. (2002). Automatic paraphrase acquisition from news articles. In *In Proc. of HLT*.

Smith, C. S. (2013). *The parameter of aspect*, volume 43. Springer Science & Business Media.

Snow, R., Vanderwende, L., and Menezes, A. (2006). Effectively using syntax for recognizing false entailment. Technical report, STANFORD UNIV CA DEPT OF COMPUTER SCIENCE.

Somasundaran, S., Ruppenhofer, J., and Wiebe, J. (2007). Detecting arguing and sentiment in meetings. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, pages 26–34.

Staliūnaitė, I. R. (2018). Learning about non-veridicality in textual entailment. Master's thesis, Utrecht University.

Stanojević, M. and Steedman, M. (2019). CCG parsing algorithm with incremental tree rotation. 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 228–239, Minneapolis, Minnesota. Association for Computational Linguistics.

Stanovsky, G., Eckle-Kohler, J., Puzikov, Y., Dagan, I., and Gurevych, I. (2017). Integrating deep linguistic features in factuality prediction over unified datasets. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 352–357, Vancouver, Canada. Association for Computational Linguistics.

Steedman, M. (2000). *The Syntactic Process*. MIT Press, Cambridge, MA, USA.

Strötgen, J., Bögel, T., Zell, J., Armiti, A., Canh, T. V., and Gertz, M. (2014). Extending HeidelTime for temporal expressions referring to historic dates. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2390–2397, Reykjavik, Iceland. European Language Resources Association (ELRA).

Strötgen, J. and Gertz, M. (2010). HeidelTime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 321–324, Uppsala, Sweden. Association for Computational Linguistics.

Strötgen, J. and Gertz, M. (2011). Wikiwarsde: A german corpus of narratives annotated with temporal expressions. In *Proceedings of the conference of the German society for computational linguistics and language technology (GSCL 2011)*, pages 129–134. Citeseer.

Strötgen, J., Minard, A.-L., Lange, L., Speranza, M., and Magnini, B. (2018). KRAUTS: A German temporally annotated news corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Styler IV, W. F., Bethard, S., Finan, S., Palmer, M., Pradhan, S., de Groen, P. C., Erickson, B., Miller, T., Lin, C., Savova, G., and Pustejovsky, J. (2014). Temporal annotation in the clinical domain. *Transactions of the Association for Computational Linguistics*, 2:143–154.

Sukhbaatar, S., Weston, J., Fergus, R., et al. (2015). End-to-end memory networks. *Advances in neural information processing systems*, 28.

Szarvas, G. (2008). Hedge classification in biomedical texts with a weakly supervised selection of keywords. In *Proceedings of acl-08: HLT*, pages 281–289.

Szarvas, G., Vincze, V., Farkas, R., and Csirik, J. (2008). The BioScope corpus: annotation for negation, uncertainty and their scope in biomedical texts. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 38–45, Columbus, Ohio. Association for Computational Linguistics.

Szarvas, G., Vincze, V., Farkas, R., Móra, G., and Gurevych, I. (2012). Cross-genre and cross-domain detection of semantic uncertainty. *Computational Linguistics*, 38(2):335–367.

Szpektor, I. and Dagan, I. (2008). Learning entailment rules for unary templates. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 849–856, Manchester, UK. Coling 2008 Organizing Committee.

Szpektor, I., Shnarch, E., and Dagan, I. (2007). Instance-based evaluation of entailment rule acquisition. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 456–463, Prague, Czech Republic. Association for Computational Linguistics.

Szpektor, I., Tanev, H., Dagan, I., and Coppola, B. (2004). Scaling web-based acquisition of entailment relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 41–48, Barcelona, Spain. Association for Computational Linguistics.

Tarski, A. (1937). *Einführung in Die Mathematische Logik: Und in Die Methodologie der Mathematik*. J. Springer.

Teng, J., Li, P., Zhu, Q., and Ge, W. (2016). Joint event co-reference resolution and temporal relation identification. In *Workshop on Chinese Lexical Semantics*, pages 426–433. Springer.

UzZaman, N. and Allen, J. (2011). Temporal evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 351–356, Portland, Oregon, USA. Association for Computational Linguistics.

UzZaman, N., Llorens, H., Derczynski, L., Allen, J., Verhagen, M., and Pustejovsky, J. (2013). Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 1–9.

Van Benthem, J. (2008). A brief history of natural logic.

Van Der Auwera, J. and Ammann, A. (2005). Overlap between situational and epistemic modal marking. *World atlas of language structures*, pages 310–313.

Van Lambalgen, M. and Hamm, F. (2008). *The proper treatment of events*, volume 6. John Wiley & Sons.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Vendler, Z. (1957). Verbs and times. *The philosophical review*, 66(2):143–160.

Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Katz, G., and Pustejovsky, J. (2007). Semeval-2007 task 15: Tempeval temporal relation identification. In *Proceedings of the 4th international workshop on semantic evaluations*, pages 75–80. Association for Computational Linguistics.

Verhagen, M., Sauri, R., Caselli, T., and Pustejovsky, J. (2010). Semeval-2010 task 13: Tempeval-2. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 57–62. Association for Computational Linguistics.

Vincze, V. (2014). Uncertainty detection in natural language texts. *PhD, University of Szeged*, page 141.

Vu, H. T., Greco, C., Erofeeva, A., Jafaritazehjan, S., Linders, G., Tanti, M., Testoni, A., Bernardi, R., and Gatt, A. (2018). Grounded textual entailment. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2354–2368, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Vulić, I., Gerz, D., Kiela, D., Hill, F., and Korhonen, A. (2017). Hyperlex: A large-scale evaluation of graded lexical entailment. *Computational Linguistics*, 43(4):781–835.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2019). *SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems*. Curran Associates Inc., Red Hook, NY, USA.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Wang, S. and Jiang, J. (2016). Learning natural language inference with LSTM. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1442–1451, San Diego, California. Association for Computational Linguistics.

Webber, B. L. (1988). Tense as discourse anaphor. *Computational Linguistics*, 14(2):61–73.

Weeds, J. and Weir, D. (2003). A general framework for distributional similarity. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 81–88.

Weir, D., Weeds, J., Reffin, J., and Kober, T. (2016). Aligning packed dependency trees: a theory of composition for distributional semantics. *Computational Linguistics*, 42(4):727–761.

Weisman, H., Berant, J., Szpektor, I., and Dagan, I. (2012). Learning verb inference rules from linguistically-motivated evidence. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 194–204.

Weston, J., Chopra, S., and Bordes, A. (2015). Memory networks. In *Proceedings of the 2015 International Conference on Learning Representations, ICLR*.

White, A. S. and Rawlins, K. (2018). The role of veridicality and factivity in clause selection. In *Proceedings of the 48th annual meeting of the north east linguistic society*, pages 221–234.

White, A. S., Reisinger, D., Sakaguchi, K., Vieira, T., Zhang, S., Rudinger, R., Rawlins, K., and Van Durme, B. (2016). Universal decompositional semantics on Universal Dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723, Austin, Texas. Association for Computational Linguistics.

White, A. S., Rudinger, R., Rawlins, K., and Van Durme, B. (2018). Lexicosyntactic inference in neural models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4717–4724, Brussels, Belgium. Association for Computational Linguistics.

Williams, A., Nangia, N., and Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Wu, A. S., Do, B. H., Kim, J., and Rubin, D. L. (2011). Evaluation of negation and uncertainty detection and its impact on precision and recall in search. *Journal of digital imaging*, 24(2):234–242.

Wu, F. and Weld, D. S. (2010). Open information extraction using Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 118–127, Uppsala, Sweden. Association for Computational Linguistics.

Xie, N., Lai, F., Doran, D., and Kadav, A. (2019). Visual entailment: A novel task for fine-grained image understanding. *ArXiv*, abs/1901.06706.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. (2019). *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. Curran Associates Inc., Red Hook, NY, USA.

Yoshikawa, M., Noji, H., Mineshima, K., and Bekki, D. (2019). Automatic generation of high quality CCGbanks for parser domain adaptation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 129–139, Florence, Italy. Association for Computational Linguistics.

Yu, C., Zhang, H., Song, Y., Ng, W., and Shang, L. (2020). Enriching large-scale eventuality knowledge graph with entailment relations. In *Automated Knowledge Base Construction*.

Zeichner, N., Berant, J., and Dagan, I. (2012). Crowdsourcing inference-rule evaluation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 156–160, Jeju Island, Korea. Association for Computational Linguistics.

Zhang, C. and Weld, D. S. (2013). Harvesting parallel news streams to generate paraphrases of event relations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1776–1786, Seattle, Washington, USA. Association for Computational Linguistics.

Zhou, B., Ning, Q., Khashabi, D., and Roth, D. (2020). Temporal common sense acquisition with minimal supervision. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7579–7589, Online. Association for Computational Linguistics.

Zhou, B., Richardson, K., Ning, Q., Khot, T., Sabharwal, A., and Roth, D. (2021). Temporal reasoning on implicit events from distant supervision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1361–1371, Online. Association for Computational Linguistics.