THE UNIVERSITY *of* EDINBURGH

Explicit Discourse Modelling for Coreference and Summarization

Matt Grenander

Doctor of Philosophy

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

2025

Abstract

Understanding and responding to natural language requires a level of representation for the input text. When reading about a character in a novel, we may remember them by attributes such as their name, events they are involved in, or their relationships with others. Many modern approaches choose a straightforward strategy: they simply store the entire input document in their context window. While this approach has merit, it becomes apparent with longer documents that storing the entire input text in context may be computationally difficult and wasteful. In cases where it is feasible, it may still impede performance, as the document's length may hinder the model's ability to focus on relevant aspects of the input.

This thesis investigates whether more careful text representations are suitable for two discourse-level tasks: coreference resolution and text summarization. We are interested in maintaining an explicit representation of the discourse, which compresses the input text into a more efficient representation. Our interest in efficient representations also leads us to propose incremental models. This process mimics human language processing, where text is consumed incrementally instead of simultaneously. Incremental models are also crucial for downstream applications that require incrementality, such as in dialogue interaction.

This thesis argues that explicit discourse representations can lead to more efficient processing, better performance, or both. First, we propose an incremental, memory-based mechanism for the coreference resolution task. The system processes text sentence-by-sentence, storing encountered mentions as partial coreference clusters in a memory matrix. In an incremental setting, we show that our proposed surpasses contemporary baselines when they are constrained to an incremental setting.

Second, we consider a generative, seq2seq paradigm for coreference resolution. Instead of holding the entire document in context, we propose a compressed, model-based discourse representation. Our proposed method truncates the context to its mentions and organizes them into entity representations. We show that this representation maintains similar performance to a naively incremental system, while discarding a majority of the document's context. In the case where singleton mentions are included in the data, our compressed representation surpasses state-of-the-art performance in a more efficient manner.

Our last task considers discourse modelling in a narrative summarization task. Here, we investigate a plan-based approach, where the generated summary is grounded in

a high-level plan of summary content. We find that although summaries are well-grounded to their plans, they are no more faithful to the source document than non-planning baselines. Human evaluation shows generated plans contain an equal amount of hallucinated content as the summary, leading to summaries that grounded but unfaithful. When we replace these plans with powerful, LLM-generated ones, summary quality improves dramatically. The result emphasizes the importance of high-quality plans in planning-based approaches to summarization.

Lay Summary

Digital assistants powered by large language models (LLMs), such as OpenAI's ChatGPT or Google's Gemini, have rapidly become popular with users for everyday tasks. One useful application is the ability to input long-form text, such as books, video transcripts, or news articles, and ask the assistant to reason over the entire text. However, these tasks are rarely performed with current technology, as reasoning long documents in their entirety is both costly and inaccurate. Language models are inefficient at long range reasoning, and therefore limited when handling large volumes of text.

This thesis looks at more efficient ways to process, understand and respond to long-form text inputs. Our core idea is to design systems that explicitly model the context by maintaining a smaller representation of the text, such as entities or key events. In two chapters, we also consider incremental processing as a means to efficiently model context, where text is processed chunk by chunk.

We investigate our ideas through the lens of two NLP tasks. The first is coreference resolution, a task in which the model must track expressions referring to same real-world entities, such as *John*, *my friend*, or *him*. We develop a system that builds explicit representations of the entities encountered, updating or creating entries as new mentions of thee entities are encountered. Next, we examine a new paradigm for coreference resolution that leverages generative large language models. We apply our ideas on incremental, efficient representations to this paradigm. Our proposed system stores and organizes encountered entities in a dynamic, specialized text prompt.

The second task is automatic summarization, where a model is tasked with shortening text such as a news article, telephone conversations or a novel into a succinct summary. Here, we investigate an approach where the model first generates a 'plan' to guide summary writing. We find that generating a good quality plan is difficult and that often, inaccurate plans lead to poor summaries. However, if an oracle plan is used instead, summary quality significantly increases.

Acknowledgements

Writing a PhD thesis is certainly a long and formative journey, and there are many people I am thankful for, either for their support, feedback, discussions, or for providing timely distractions when they were needed.

I primarily wish to thank my supervisor, Mark Steedman. I am grateful for the countless and interesting discussions, timely and insightful feedback, and the general availability for providing support. These discussions have undoubtedly shaped my perspective on how to conduct good research, ask interesting questions and to think critically in periods of hype.

I am similarly thankful to my secondary supervisor, Shay Cohen. His research expertise has helped me many times throughout my PhD in asking meaningful questions that improved my research and getting past the technical hurdles. I would also like to thank Mirella Lapata for her valuable feedback during my annual reviews, and Selma Tekir and Bonnie Webber for their helpful discussions and questions. Lastly, I would like to thank Mirella (again) and Michael Collins for kindly agreeing to examine my thesis.

My research labmates have been a continual source of knowledge and I thank them for many useful discussions that influenced the direction of my research. In no particular order, they are Miloš Stanojević, Ratish Pudupully, Liane Guillou, Louis Mahon, Sander Bijl de Vroe, Nick M^{c}kenna, Tianyi Li, Mohammad Javad Hosseini, Liang Cheng, Tianyang Liu, Sabine Weber, Nikita Moghe, Elizabeth Nielsen, Katarzyna Pruś, Iona Carslaw, Sivan Milton, Zhaowei Wang and Chuang Liu, as well as my colleague Parag Jain.

I am also grateful to my colleagues during my internship at Amazon: Siddharth Varia, Paula Czarnowska, Yogarshi Vyas, Bonan Min and Kishaloy Halder. I would also like to thank Kathleen McKeown and Faisal Ladhak for their numerous helpful discussions.

Thank you to my friends in Edinburgh: Ben, Anna, Steinar, Matt, Rana, Claire, Mark, Bhargavi, Jesse, Yuelin, and outside Edinburgh: Konrad, Jackie, Johannes, Tanjiha, Chris, Patrick, Riley, Fay, Tanya and Noah. Their support may not be related to research, but provided a welcome distraction at times.

I would also like to extend my gratitude to my parents and my stepfather, for their unconditional support, which started far before my PhD studies.

Lastly, I am grateful to my loving partner Jane. Apart from the countless memories together, she has been a constant source of support and care throughout my entire PhD.

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.

(*Matt Grenander*)

Table of Contents

Chapter 1

Introduction

1.1 Motivation

Language technologies powered by Large Language Models (LLMs), such as OpenAI's ChatGPT and Google's Gemini, have rapidly emerged as innovative, practical technologies. Their ability to seemingly understand, reason and respond to a wide variety of user queries have resulted in widespread adoption. The ability to use **context** is key to their usefulness. Users may interact naturally with the system, generating long conversations or even inputting documents and whole books. The system is expected to respond appropriately based on the context, for example by resolving referring expressions or sifting through previous information.

What is unseen by users is the vast amounts of computation involved in generating answers to their queries. LLMs' neural architecture is based on transformer networks (Vaswani et al., 2017), which require a quadratic number of computations per input token. Compounding this issue is that LLMs hold onto the full input context when generating a response. By design, they do not discard previous input tokens, meaning that as users input longer and longer texts, the context length grows, and the amount of computation becomes greater and greater.

Computation is not the only concern with long contexts. Many prior works have demonstrated that transformers are not accurate when reasoning with long contexts (Liu et al., 2024; Levy et al., 2024), including practical tasks such as summarization (Shaham et al., 2022). This issue is a major bottleneck for improving language technologies, as users cannot rely on LLMs for long document processing.

Lastly, from a psycholinguistic perspective, the LLM methodology of retaining the entire context seems deeply unsatisfying. Human processing is strongly incremental

(Altmann and Steedman, 1988), and works have demonstrated that humans are continually compressing and recoding language as it is received (Christiansen and Chater, 2016). In many cases, we turn to our understanding of human processing in order to design our models.

In particular, in both Chapters 3 and 4, we will look at incremental processing as a solution to designing efficient and accurate systems. The definition of incremental processing shifts slightly between Chapters 3 and 4, but in principle remains the same. In an incremental setting, the model is required to process the text chunk by chunk, and only begins processing the next chunk of text after it has made coreference predictions for the current one. The predictions for the current text chunk are then compressed into a representation that models the discourse. In Chapter 3, the text is processed sentence-by-sentence, and the discourse is represented by contextual embeddings of the entities encountered so far. In Chapter 4, the unit of incrementality is broadened to chunks of 100 tokens rounded up to the nearest sentence, and entities are organized as formatted strings. In these chapters, the existing systems we examine are often non-incremental, or inefficient at incremental processing, making them impractical for downstream applications such as dialogue, where incrementality is a necessity. In these chapters, we are also primarily motivated to design incremental systems that are more efficient than their non-incremental counterparts.

In the era of LLMs, one valid question is whether coreference resolution is still relevant when LLMs can perform the majority of NLP tasks without the need of intermediate steps. We offer two motivations why coreference resolution and summarization are still interesting research topics despite LLMs' recent dominance:

- Motivation 1: Coreference resolution serves as an excellent test bed for assessing how effectively pre-trained language models understand and resolve reference. This task is fundamental, as reference is a ubiquitous and constantly used feature of human language. In this thesis, while we do not directly experiment with prompting LLMs for coreference, we rely on pre-trained language models such as XLNet (Yang et al., 2019) and T0 (Sanh et al., 2022a). Evaluating their ability to perform coreference and the challenges they face can be seen as a proxy for LLMs' ability to handle coreference as well.

- Motivation 2: Coreference resolution is still used as an intermediate step in many downstream applications, such as in summarization (Hua et al., 2023; Lei and Huang, 2025), knowledge graphs (Jin et al., 2022; Liu et al., 2023b; Chun and

Xue, 2024; Yan et al., 2024) and argument mining (Liu et al., 2023a). High-performing coreference resolution models are an important factor in the success of these downstream applications. In these cases, we note that LLM approaches to coreference resolution score more than 20 F1 points below state-of-the-art methods (Le and Ritter, 2024), meaning it is certainly worthwhile exploring methods outside of prompting LLMs.

1.2 Thesis statement

In this thesis, we investigate how to design systems that efficiently and accurately process language. We argue that language models do not benefit from retaining the entire all previous inputs as context, and propose explicitly modelling the discourse as a solution. Our solutions involve building explicit representations of the text encountered so far and incremental processing.

We apply our ideas to two tasks across the thesis. For the coreference task, we maintain representations of the entities while processing the text incrementally, reducing the memory and computation requirements of existing systems while maintaining high performance. In the summarization task, we propose building an abstractive plan of events in the text before generating the summary. Although we find the complex plan formulation is difficult to model, substituting in a high-quality plan allows for better summary quality and higher faithfulness to the source text.

1.3 Outline of Thesis

The thesis is structured as follows:

Chapter 2 We introduce the relevant background material for this thesis. We detail the cognitive theories underpinning our approach, then discuss the coreference resolution and summarization tasks, along with the relevant datasets and evaluation metrics. We finish with a discussion on neural architectures and pre-trained language models that we use throughout the thesis.

Chapter 3 We explore incorporating a discourse model with encoder-based coreference resolution systems. We propose a system that incrementally builds a discourse representation of the text so far, represented by entities in the text. It improves on the

computational efficiency of existing systems while maintaining competitive scores. We show that in an incremental setting, our system outperforms contemporary systems, with larger improvements on a coreference dataset for dialogues. We note performance differences between incremental and non-incremental systems, which we attribute to the caching mechanism in the base encoder.

Chapter 4 We investigate a seq2seq paradigm for learning coreference resolution. Using the same principles as Chapter 3, we design an model-based system which incrementally builds a discourse representation using entities encountered in the text. In the seq2seq formulation, we represent the entities as lists of their mentions, ordered by mention recency. We find our proposal performs strongly against a naive, Full-Prefix Incremental baseline, while effectively reducing memory requirements. However, dataset artifacts, particularly the lack of singleton annotation, emerges as a source of noise for incremental systems. The model-based system surpasses state-of-the-art methods when singletons are included in the dataset annotation, but otherwise lags behind. We explore adding NER labels to offset this bias, but find additional artifacts in the dataset prevents further improvement.

Chapter 5 We apply our ideas to a completely new task, namely automatic text summarization. In this chapter, the discourse model we consider takes a considerably different formulation. We investigate plan-guided summarization, where the summarization system generates a summary conditioned on a plan reflecting salient events from the source text. We propose a narrative plan formulation based on sub-events, and also explore QA-based plans. Despite prior work promoting plan-guided approaches, our results are negative, and we find that models tend to hallucinate non-factual content in both plans and summaries. However, we find that replacing the generated plan with a high-quality, oracle plan results in higher summary quality and faithfulness.

Chapter 6 We conclude the thesis, summarizing our overall findings and providing directions for future work.

Parts of this thesis have been published or are currently under submission at various venues:

- Chapter 3 is published at EMNLP 2022 (Grenander et al., 2022).
- Chapter 4 is under submission at EMNLP 2025.

- Chapter 5 is accepted at the 7th Workshop on Narrative Understanding (Grenander et al., 2025). The work was initiated at an internship at Amazon Web Services (AWS), and continued after the internship finished in collaboration with colleagues listed in the publication.

Chapter 2

Background

In this chapter, we introduce various aspects of the coreference resolution and summarization tasks. We describe key cognitive theories relevant to our discourse modelling approach in Section 2.1. The remaining sections details technical aspects of the thesis: we describe the coreference resolution task, datasets and evaluation metrics in Section 2.2, followed by the summarization task, datasets and evaluation metrics in Section 2.3. We then describe the base model architectures we use in the thesis in Section 2.4 and pre-trained language models in Section 2.5.

2.1 Discourse Modelling

2.1.1 Discourse Representation Theory

Discourse Representation Theory (Kamp, 1981) is a framework for semantics based on incrementally building up a mental model of the world as the discourse progresses. The framework was preceded by Webber (1978) and Karttunen (1969), who argued similarly in favour of building such as discourse model; in particular, Karttunen (1969) introduced the notion of **discourse referents**, representing entities that are under discussion in the current discourse.

DRT models discourse with what are called **discourse representation structures** (DRS). A DRS consists of discourse referents, and a set of conditions on these entities, representing the known information about them. The DRS is updated incrementally with each new utterance, with operations that encode new information or add new discourse referents.

DRT considers various anaphora such as pronouns, but is concerned with how to

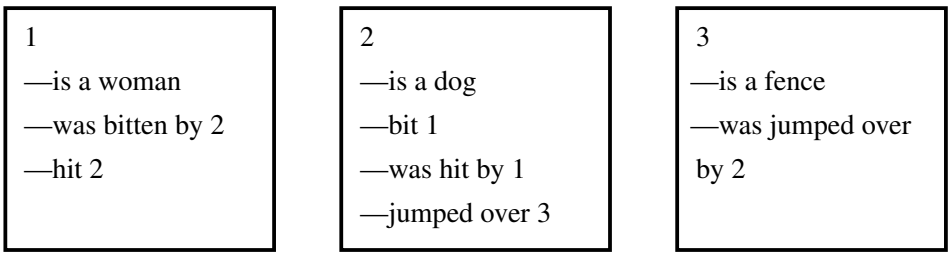


Figure 2.1: Examples of file cards in File-Change Semantics, from Heim (1983).

derive a logical form from natural language and how pronouns are resolved within sentences. In this thesis, we consider both intra- and inter-sentential coreference resolution. DRT inspires our approach towards coreference resolution, but we do not vigorously implement it in our models.

Lastly, Segmented Discourse Representation Theory (SDRT, Asher and Lascarides (2003)) is an extension to DRT (or any dynamic semantics theory) that includes discourse relations such as Contrast, Explanation and Elaboration, and models the discourse as a graph. Discourse relations aid in creating a more coherent interpretation of the discourse. The additional structure imposes constraints on the discourse, leading to a better explanation for many types of anaphora resolution compared to DRT. For example, one constraint that SDRT enforces is the Right Frontier Constraint (Polanyi, 1988), which says pronouns cannot attach to any antecedent in the discourse, but may only resolve to the last node in the graph or one that dominates it.

2.1.2 File-Change Semantics

File Change Semantics (FCS, Heim (1982, 1983)) is a framework for natural language semantics that is similar to DRT. FCS proposes modelling discourse through **file cards** which track and update entities encountered after each utterance. The collection of file cards at a given utterance can be seen as the state of the discourse up to that point.

Consider the following text, from Heim (1983):

(a) A woman was bitten by a dog. (b) She hit it. (c) It jumped over a fence.

FCS models the discourse as follows: After (a), it creates two cards, labelled (1) *a woman* and (2) *a dog*. On file card (1), it marks "bit by (2)", and correspondingly, it writes "bit (1)" on file card (2). Following (b), it updates (1) and (2) to reflect that (1) has hit (2) and that (2) was hit by (1). After reading (c), it creates a new card (*a fence*), records that it was jumped over by (2), and updates (2) to reflect that it jumped over (3).

Figure 2.1 shows FCS's model of the discourse after reading (c).

As with DRT, FCS is a theory of dynamic semantics, used to derive logical forms from natural language. Although it deals considers various types of referring expressions such as pronouns, it generally does not consider long range, inter-sentential reference as we do. However, we find the framework useful as a method for modelling discourse, and the methods we present are inspired by FCS's file-keeping.

2.1.3 Centering Theory

Centering theory (Grosz et al., 1995) is a framework for understanding and reasoning about local coherence, with implications for the realization of referring expressions. It is based on earlier work from Grosz and Sidner (1986). Grosz and Sidner model discourse structure using three components: a linguistic structure, an intentional structure, and an attentional state, where the state is modelled with a focus space stack.

Centering theory examines attentional state at the local level. It proposes that every utterance contains a entity which is the most salient. This entity is called the **backward-looking center**, and is unique to each utterance. At each new utterance, the backward-looking center may shift to a new entity, which then becomes the backward-looking center and the new focus of the discourse.

Futhermore, Centering theory defines **forward-looking centers**, as the set of potential future salient entities, of which one will serve as the backward-looking center in the next utterance. The forward-looking centers are ranked by their salience in the discourse by a variety of factors which are not fully enumerated in Grosz et al. (1995). These factors include features such as syntactic cues; for example, subjects are considered more salient than other grammatical positions. The highest-ranked forward-looking center is named the **preferred center**.

To better understand the concepts of backward-looking and forward-looking centers, considering the following discourse segment, from Grosz et al. (1995):

(a) John went to his favorite music store to buy a piano.
(b) He had frequented the store for many years.
(c) He was excited that he could finally buy a piano.
(d) He arrived just as the store was closing for the day.

Now, compare the above discourse segment with the following:

(a) John went to his favorite music store to buy a piano.

(b) It was a store John had frequented for many years.

(c) He was excited that he could finally buy a piano.

(d) It was closing just as John arrived.

Grosz et al. (1995) recognize the former text seems more coherent compared to the latter, despite both texts discussing the same entities and having the same meaning. Centering explains the difference in coherence in terms of transitions between backward-looking and forward-looking centers. Each type of transition is ordered by coherence. For example, transitions which keep the same backward-looking center are more coherent, while transitions that shift the backward-looking center to a new entity are less coherent, especially if the new entity is not the preferred center.

In the first text, *John* remains the backward-looking center and the preferred center for the entirety of the text, meaning the overall text is highly coherent. In the second text, the backward-looking center shifts at each utterance, constantly flipping between *John* and *the store*. These repeated changes results in the second text being less coherent.

Centering also constrains the usage of pronouns in coherent texts. It stipulates that entities in the current utterance cannot be expressed as pronouns unless the backward-looking center is also a pronoun. In particular, if there is only one pronoun in the current utterance, then Centering theory says the utterance is continuing to discuss the same entity, i.e. the backwards-looking center has not changed.

Lastly, Centering theory is by itself a theory of local coherence, but works such as Brennan et al. (1987) and Chai and Strube (2022) have developed coreference resolution methods through implementing the transitions discussed in Grosz et al. (1995). Poesio et al. (2004b) make a systematic attempt to verify the main claims in Centering theory through implementation, but were unable to confirm all its claims after finding the theory underspecified. In this thesis, we do not attempt any literal implementation of Centering, but we are inspired by modelling an entity's discourse salience.

2.2 Coreference Resolution

2.2.1 Task Description

Coreference resolution systems are tasked with finding all linguistic expressions in a text that refer to the same real-world entity. We refer to these referring expressions as **mentions** (Hirschman and Chinchor, 1998; Mitkov, 2002), and the real-world entity being referred to as the **referent** (Hirst, 1981). If two expressions refer to the same

Text	Labels
"I drove Joe home because he lives close to my apartment", she said.	0: [Joe, he] 1: [I, my, she]
Striking auto workers ended their 19-day occupation of a metal shop at a Peugeot S.A. factory in France. The Peugeot metalworkers began filing out of the shop after voting to abandon the occupation.	0: [Striking auto workers, their, The Peugeot metalworkers] 1: [their 19-day occupation of a metal shop at a Peugeot S.A. factory in France, the occupation] 2: [a metal shop at a Peugeot S.A. factory in France, the shop] 3: [Peugeot S.A., Peugeot]

Figure 2.2: Two examples from the coreference resolution task. The left column shows example inputs, while the right column depicts expected labels. Each integer represents a distinct entity, with the comma-separated text spans denoting its mentions in the text. Although the figure displays the mentions as text, in reality they are represented by token indices to avoid ambiguities.

entity, we say they are **co-referring**. Finally, the prior mention that gives the referring expression its meaning is called the **antecedent** (Hirst, 1981).

Figure 2.2 shows two text samples with the expected labels. In the top example, we note the dynamic nature of coreference, particularly in dialogue domains. This example includes an instance of **deixis** (Bosch, 1983), where the correct resolution for *I* depends on the speaker.

In the bottom example, we note that mentions may be nested but are otherwise non-overlapping. For example, in the second example, *their* and *their 19-day occupation ...* are syntactically related but refer to different entities. We will assume this non-overlapping nested structure when designing our models in Chapters 3 and 4.

Lastly, both examples contain expressions that could be the target of coreference, but do not happen to be in the given sample. For example, *my apartment* or *France* could feasibly be the target of coreference in a subsequent utterance. In some datasets, these expressions are annotated as mentions, where they are called **singleton mentions** (Poesio et al., 2018).

Mentions may be realized by a wide variety of syntactic categories, such as pronouns, indefinite and definite noun phrases, and proper names. Reference with specific syntactic classes has been studied extensively in the literature; we refer the reader to work on

pronouns (Bosch, 1983) and indefinite and definite descriptions (Hirst, 1987; Haddock, 1989).

Additionally, in certain languages such as Spanish and Chinese, referring expressions may even be "invisible". This phenomenon, called zero anaphora (Mitkov, 2002), allows the subject or object to be dropped when it is clear from the surrounding context. For example, in Spanish, verbs are conjugated according to the subject, meaning the identity of the subject is clear from the choice of verb.

Coreference resolution is typically separated into two major steps. In the first step, called **mention detection**, the model is tasked with identifying text spans that correspond to mentions. Although Figure 2.2 displays mention spans as text, in reality we will index each token in the text in order to avoid any ambiguity.

In the second step, called **mention clustering**, the model clusters the mentions into co-referring groups. We will refer to each group of co-referring mentions as an **entity cluster**, or a **coreference chain**. For example, in Figure 2.2's bottom example, there are 4 entity clusters.

Although in this thesis we focus solely on coreference, it is worth noting that coreference is one type of reference among other interesting phenomena. Generally, reference in text to previously mentioned entities is called **anaphora**, which includes coreference (same-identity anaphora).

One example of non-identical anaphora is **bridging anaphora** (Clark, 1975; Asher and Lascarides, 1998; Poesio et al., 2004a), where the listener can infer the identity of the referring expression based on the context and world knowledge. For example, in the utterance, "*We went to a restaurant. The server was rude.*", a listener will understand the identity of the server based on the knowledge that most restaurants have servers. From the perspective of discourse modelling, when a listener hears the antecedent "*restaurant*", they update their mental discourse model to accommodate any restaurant-related entities in the near future.

The other type of anaphora we mention here but otherwise do not address in the thesis is **split-antecedent anaphora** (Eschenbach et al., 1989; Ingria and Stallard, 1989; Kamp and Reyle, 1993). This phenomenon occurs when a referring expression has multiple antecedents, as in *John met Mary at the café, and they went to the park.* Split-antecedent anaphora may require bridging-like inferences in to resolve, such as: *John, Joe, Mary, and Sarah met a café. The girls went to the park.*, which assumes the listener knows typical male and female names in order to infer the identity of "*the girls*".

	OntoNotes	CODI-CRAC	LITBANK
Domain	Various	Dialogue	Literature
# Documents	3493	134	100
Avg. # Tokens	582.2	1102.4	2105.3
Avg. # Mentions	55.7	312.0	291.0
Avg. # Clusters (exc. singletons)	12.7	30.1	240.4
Avg. # Singletons	N/A	138.5	50.7

Table 2.1: Statistics for the coreference resolution datasets we consider in this thesis. Tokens are represented as SentencePiece tokens (Kudo and Richardson, 2018). We report the average number of clusters separately from the average number of singletons to provide a fuller picture of their distribution in each dataset.

2.2.2 Datasets

In our coreference work, we use three datasets: OntoNotes (Weischedel et al., 2013), CODI-CRAC (Khosla et al., 2021), and LitBank (Bamman et al., 2020). The three datasets differ in various ways; the main statistics are highlighted in Table 2.1.

2.2.2.1 OntoNotes

The OntoNotes 5.0 dataset (Weischedel et al., 2013) was released for the CoNLL-2012 Shared Task (Pradhan et al., 2012). Its large-scale annotation covers 3493 documents across various domains, making it a key resource in the community. OntoNotes includes 7 domains: broadcast conversations, broadcast news, magazines, telephone conversations, weblogs, and Bible passages.

The OntoNotes annotation scheme does not restrict entity types for coreference; any co-referring noun phrase is marked as a mention. Verb phrases can also be marked as co-referring if they co-refer with a noun phrase (e.g. '*the rise*' with *The stock market rose sharply.*).

One challenge with OntoNotes is the lack of singleton annotation. Mentions in OntoNotes are only annotated if they co-refer with another mention in the document, meaning many noun phrases that *could* be the target of coreference are left unmarked. This artifact presents particular challenges for our proposed systems in Chapter 4.

In order to ensure high-quality annotations, coreference is doubly annotated and adjudicated (Hovy et al., 2006). The inter-annotator agreement is reported in several

works as the dataset was developed: Hovy et al. (2006) report the inter-annotator agreement as 91.8%, while Weischedel et al. (2010) note the overall average agreement between individual annotators and the adjudicated result as 86%. In the CoNLL-2012 Shared Task (Pradhan et al., 2012), the inter-annotator agreement is provided per domain, ranging from 78.4% to 96.0%. The average agreement score is 87.4%. In all cases, the inter-annotator agreement score is measured using the *MUC* score between annotators / adjudicator.

2.2.2.2 CODI-CRAC

The CODI-CRAC 2021 dataset was introduced in the CODI-CRAC 2021 Shared Task (Khosla et al., 2021). It annotates coreference resolution in dialogues, drawing from four existing corpora: AMI (Carletta, 2006), Light (Urbanek et al., 2019), Persuasion, (Wang et al., 2019) and Switchboard (Godfrey et al., 1992). The four corpora's domains cover meeting dialogues, role playing games, a persuasion task, and telephone conversations, respectively. We are primarily interested in the dynamic nature of dialogue and how its inherent incremental nature may be beneficial for our incremental models.

The annotation scheme is generally similar to OntoNotes but differs in one important way. Unlike OntoNotes, CODI-CRAC includes singleton annotation, which are called *markables*. Markables are annotated as all NPs, even non-referring NPs such as expletive pronouns (e.g. '*It*' in *It was raining.*) and predicative NPs (e.g. '*a linguist*' in *John is a linguist*). As shown in Table 2.1, this annotation decision results in a far greater number of singleton mentions compared to LitBank, which we discuss next.

No inter-annotator agreement statistics are detailed in the Shared Task report; however, Khosla et al. (2021) specify that the dataset is singly annotated by two annotators, with spot checks carried out by a third annotator.

We note that a follow-up work, the CODI-CRAC 2022 Shared Task (Yu et al., 2022a), expanded the 2021 dataset to a total of 218 documents. However, our experiments using CODI-CRAC in Chapter 3 precede this dataset's release.

2.2.2.3 LitBank

The LitBank dataset (Bamman et al., 2020) consists of annotated samples selected from literary texts. The data is drawn from 100 works of fiction in the US public domain written between 1719 and 1922. LitBank's long document length is appealing as it poses efficiency challenges.

LitBank's annotation scheme again largely follows OntoNotes, with the exception that singletons are annotated. Another important difference is LitBank only considers six entity types: people, facilities, locations, geo-political entities, organizations, and vehicles, resulting in much fewer singleton mentions compared to CODI-CRAC.

The majority of LitBank is singly annotated; however, the authors doubly annotate 10% of LitBank (10 full texts) to measure inter-annotator agreement. As before, inter-annotator agreement is computed using the *MUC* score between annotators. They find the *MUC* score to be 95.5%. The authors speculate the very high annotation agreement may be due to restricting the number entity types, unlike in OntoNotes.

2.2.3 Evaluation Metrics

In this thesis, we use the three main evaluation metrics set out in the CoNLL-2012 Shared Task (Pradhan et al., 2012): *MUC* (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998) and $CEAF_{\phi_4}$ (Luo, 2005). Both B^3 and $CEAF_{\phi_4}$ are meant to address deficiencies in previously proposed metrics, but all three are commonly used. In order to get a single score to compare each system, we follow Pradhan et al. (2012)'s precedent and compute the average of the three metrics, and report it as the **CoNLL Score**.

We will assume a set of reference clusters $\mathcal{C} = \{C_1, \ldots, C_n\}$, where C_i is a single entity cluster (i.e. a list of mentions) and predicted clusters $\mathcal{K} = \{K_1, \ldots, K_m\}$. If two mentions are co-referring, we will also say they are *linked*.

2.2.3.1 *MUC*

The MUC metric, proposed by Vilain et al. (1995), is often referred to as a *link*-based metric (Pradhan et al., 2012). It computes recall as the number of common links in $\mathcal{C}$ and $\mathcal{K}$ divided by $|\mathcal{C}|$, and the precision as the number of common links divided by $|\mathcal{K}|$.

Figure 2.3's top example provides a simple demonstration of the *MUC* metric. The ground truth consists of two entities, *1–2* and *3–4*, and the model only successfully predicts *1–2*. In this scenario, *MUC* provides an intuitive recall score of 50%, and a precision score of 100%. Compared to *MUC*, other metrics are more difficult to interpret in this case.

Although MUC is simple and interpretable, it also carries certain drawbacks. For example, it does not consider singleton mentions, since they do not link to other mentions. It also tends to weigh more frequently mentioned entities higher, since these

Ground Truth	System Prediction	*MUC*		B^3		$CEAF_{\phi_4}$	
		Rec.	Prec.	Rec.	Prec.	Rec.	Prec.
①—② ③—④	①—② ③ ④	50.0	100.0	75.0	100.0	83.3	55.6
①—②—③—④ ⑤—⑥ ⑦—⑧—⑨—⑩	①—②—③—④ ⑤—⑥ ⑦—⑧—⑨—⑩	100.0	87.5	100.0	60.0	55.6	83.3
①—②—③—④ ⑤—⑥ ⑦—⑧—⑨—⑩	①—②—③—④ ⑤—⑥ ⑦—⑧—⑨—⑩	100.0	77.8	100.0	36.0	19.0	57.1

Figure 2.3: Examples of scoring differences among difference coreference evaluation metrics, adapted from Vilain et al. (1995); Bagga and Baldwin (1998) and Luo (2005). Each figure represents a potential output, with numbers representing distinct mentions and links representing coreference. For example, in the top left figure, mentions *1* and *2* are co-referring, and *3* and *4* are co-referring. In the top right figure, the model has correctly predicted *1* and *2* are co-referring, but has missed the link between *3* and *4*.

Top: In simple examples, *MUC* provides a clear and interpretable score, while B^3 and $CEAF_{\phi_4}$ are harder to interpret.

Middle: In some use cases, merging two large entities together may be seen as a more serious error compared to merging smaller entities. B^3 and $CEAF_{\phi_4}$ will penalize these errors more strongly than the *MUC* metric.

Bottom: If the model erroneously merges many clusters, *MUC* and B^3 will still yield high recall scores. Similarly, if the model predicts very few links, *MUC* and B^3 will give high precision scores. $CEAF_{\phi_4}$ scores these errors more severely by focusing on gold and predicted entities instead of links or mentions.

entities contain more links. This tendency may or may not be desirable depending on the use case.

2.2.3.2 B^3

B^3 (Bagga and Baldwin, 1998) was proposed to address shortcomings in *MUC*; in particular the issue that *MUC* weighs all links equally. B^3 is considered a *mention*-based metric, as it computes mention overlaps between the predicted and gold entity clusters.

Recall is computed as follows: say mention m is in predicted entity cluster K and m is also in the gold cluster C. Then the overlap value is scored as $\frac{|K \cap C|}{|C|}$, i.e. the number of mentions common to K and C divided by the number of entities in C. The overlap values are then averaged over all gold mentions. Precision is calculated similarly but by swapping the roles of predicted and gold clusters: $\frac{|K \cap C|}{|K|}$, and then averaged over all predicted mentions.

The middle row of Figure 2.3 shows an example where *MUC* exhibits undesirable characteristics that B^3 aims to fix. In this example, the model mistakenly merges two large entities *1–2–3–4* and *7–8–9–10*. Bagga and Baldwin (1998) contend this error is worse than if the model erroneously merges the smaller entity *5–6* with one of the larger ones. However, *MUC* regards these two errors as equivalent, since both involve adding a single incorrect link. Accordingly, it assigns both predictions a precision score of 87.5%. Since the B^3 score is based on the number of mentions in each entity chain, it scores merging two large entities more unfavourably compared to *MUC*.

2.2.3.3 $CEAF_{\phi_4}$

CEAF (Luo, 2005) imposes a constraint that at most one entity in the predicted clusters may be matched to one gold entity, then finds the optimal matching between predicted and gold clusters. Because of the requirement that entities are aligned at most one-to-one, $CEAF_{\phi_4}$ is known as an *entity*-focused metric.

In their work, Luo (2005) proposes several similarity metrics between entity clusters, of which the ϕ_4 function is the most popular. For a predicted entity cluster K and gold entity cluster C, ϕ_4 computes their similarity as $\frac{2|K \cap C|}{|K|+|C|}$. To find the optimal matching, the problem is posed as a maximum bipartite matching problem, which is efficiently solved by the Kuhn-Munkres algorithm. Recall is then defined by summing the optimal similarity values and dividing by the number of mentions in $\mathcal{C}$, and precision is defined by dividing by the number of mentions in $\mathcal{K}$.

The bottom row in Figure 2.3 shows a case where $CEAF_{\phi_4}$ presents an advantage over MUC and B^3 metrics. In this example, the model has output a trivial response where all mentions are co-referring. MUC and B^3 still assign a recall score of 100%, despite the lack of any meaningful predictions. In this case, $CEAF_{\phi_4}$ severely penalizes the model output, since $CEAF_{\phi_4}$ requires entities to be aligned one-to-one and the model has only output a single entity.

2.3 Automatic Summarization

2.3.1 Task Description

Automatic text summarization is an NLP task where the system's goal is to create a summary from a longer text. The summary should be concise, fluent, accurate, and contain relevant aspects from the source text. Generally, summarization datasets contain reference summaries which the summarization system aims to replicate.

There are two main approaches to designing summarization systems. In **extractive summarization**, the system simply highlights fragments such as sentences from the source text and returns these fragments as the final summary. In **abstractive summarization**, the system is more akin to a human writer, freely synthesizing the summary token-by-token. In this thesis we only consider the latter approach.

Although abstractive summarization can produce more varied summaries compared to extractive systems, they also face challenges regarding generating summaries that do not misrepresent the source text. Many works have shown that abstractive summarization are prone to generating false assertions in their summaries, which are often called **hallucinations** (Kryscinski et al., 2019; Maynez et al., 2020; Lin et al., 2022; Ji et al., 2023; Wang et al., 2023). Many evaluation metrics attempt to quantify the predicted summary's faithfulness to the source text (Fabbri et al. (2022); Zha et al. (2023); Song et al. (2024), among others).

In general, evaluating summarization systems is challenging, as writing a summary is a subjective process and it is not clear which aspect best reflects a good summary. Evaluation may focus on aspects such as coverage of the source text, fluency, conciseness or faithfulness. We detail the metrics we consider in this thesis in Section 2.3.3.

Lastly, automatic summarization may consider many different text domains, such as news (Hermann et al., 2015; Narayan et al., 2018), scientific articles (Cohan et al.,

2018), TV show scripts (Chen et al., 2022), etc. Source texts may also comprise of a single document or multiple documents (Fabbri et al., 2019), or contain multiple reference summaries (Wang et al., 2022).

2.3.2 Datasets

The summarization datasets we consider focus on long documents with a clear event-focused narrative structure, as we believe that tasks with a narrative structure are more likely to benefit from discourse modelling approaches. We choose two datasets covering (1) short stories and (2) TV show transcripts.

2.3.2.1 SQuALITY

SQuALITY is a long document summarization dataset covering short stories from Project Gutenburg (Wang et al., 2022). The dataset contains 50/25/52 documents across training, validation, and test splits. Each document is paired with four summaries written by human writers, who also verify their peers' writing. The writing pipeline ensures the quality of each summary is very high. On average, the dataset contains 7648 tokens per document and 591 tokens per summary. An example from the SQuALITY dataset is shown in Figure 2.4.

We are interested in SQuALITY for two main reasons. First, as mentioned previously, short stories in SQuALITY contain a narrative structure, which may benefit from an explicit discourse modelling approach. Second, documents in SQuALITY are far longer than other traditional summarization domains such as news articles. The extreme length results in highly compressed summaries which require a greater degree of abstraction to capture. Many existing LLMs struggle with long inputs, particularly with recalling key details and hallucinations (Liu et al., 2024; Levy et al., 2024), and this phenomenon is observed in SQuALITY as well (Wang et al., 2022). We are also interested in difficulties in evaluation, as Wang et al. (2022) note that many automatic metrics weakly correlate with human judgements.

2.3.2.2 SummScreen-FD

SummScreen is a long document summarization task based on TV episode transcripts (Chen et al., 2022). Summaries in this dataset tend to be very abstractive, as plot details are usually indirectly expressed by characters in dialogue. The document length also

"Split" Campbell and I brought our ship down to a quiet landing on the summit of a mile-wide naked rock, and I turned to the telescope for a closer view of the strange thing we had come to see. It shone, eighteen or twenty miles away, in the light of the late afternoon sun. It was a long silvery serpent-like something that crawled slowly over the planet's surface. There was no way of guessing how large it was, at this distance. It might have been a rope rolled into shape out of a mountain—or a chain of mountains. It might have been a river of bluish-gray dough that had shaped itself into a great cable. Its diameter? If it had been a hollow tube, cities could have flowed through it upright without bending their skyscrapers. It was, to the eye, an endless rope of cloud oozing along the surface of the land. No, not cloud, for it had the compactness of solid substance. We could see it at several points among the low foothills. Even from this distance we could guess that it had been moving along its course for centuries. Moving like a sluggish snake. It followed a deep-worn path between the nearer hills and the high jagged mountains on the horizon. What was it? "Split" Campbell and I had been sent here to learn the answers. Our sponsor was the well known "EGGWE" (the Earth-Galaxy Good Will Expeditions.)

. . .

I forgot about the moving trees, then, and took over the telescope. Mobile trees were not new to me. I had seen similar vegetation on other planets—"sponge-trees"—which possessed a sort of muscular quality. If these were similar, they were no doubt feeding along the surface of the slope below the rocky plateau. The people in the clearing beyond paid no attention to them. I studied the crowd of people. Only the leader wore the brilliant garb. The others were more scantily clothed. All were handsome of build. The lemon-tinted sunlight glanced off the muscular shoulders of the males and the soft curves of the females. "Those furry elbow ornaments on the females," I said to Split, "they're for protection. The caves they live in must be narrow, so they pad their elbows." "Why don't they pad their shoulders? They don't have anything on their shoulders." "Are you complaining?" We became fascinated in watching, from the seclusion of our ship. If we were to walk out, or make any sounds, we might have interrupted their meeting. Here they were in their native ritual of sunset, not knowing that people from another world watched. The tall leader must be making a speech. They sat around him in little huddles. He moved his arms in calm, graceful gestures. "They'd better break it up!" Split said suddenly. "The jungles are moving in on them." "They're spellbound," I said. "They're used to sponge-trees. Didn't you ever see moving trees?" Split said sharply, "Those trees are marching! They're an army under cover. Look!" I saw, then. The whole line of advancing vegetation was camouflage for a sneak attack. And all those natives sitting around in meeting were as innocent as a flock of sitting ducks. Split Campbell's voice was edged with alarm. "Captain! Those worshippers—how can we warn them? Oh-oh! Too late. Look!" All at once the advancing sponge-trees were tossed back over the heads of the savage band concealed within. They were warriors—fifty or more of them—with painted naked bodies. . .

Captain Linden and his lieutenant "Split" Campbell make up the first manned expedition from Earth to this particular planet, aiming to investigate a large silver river on its surface. The seemingly-endless silvery strip that traveled the planet's surface was unidentifiable as of yet. They see the river-like thing early on, but Campbell spots a humanoid through his telescope–this being is much like a human man, including the fact that he wore clothing. Captain Linden decides it's time for introductions, as if he senses he can trust this being, but they watch as a female and then many other people join the first man on the surface, seemingly coming out of an underground city. Linden and Campbell think their ship is out of sight, and watch a ritual that the man is performing to the setting sun. . .

The story relates the experience of two agents who travel to an unnamed planet for Earth-Galaxy Good Will Expeditions (EGGWE). An unmanned camera has brought pictures from the planet back to Earth, showing two features of particular interest: 1) a human-like species, the Benzendella, living there, and 2) a rope-like, silvery undulating river. Captain Linden is the commander of the mission; his lieutenant is "Split" Campbell. After traveling millions of miles to reach the planet, the men land and use their telescope to check their surroundings before alighting from the spaceship. They see the river and the human-like beings who look like human ancestors from a million years ago. As they watch, the leader of the humans seems to perform a kind of ritual, but then, Linden notices some trees moving uphill and watches in horror as warriors toss the trees aside and launch an attack on the humans using clubs or whips with stones tied to the ends. . .

Figure 2.4: An excerpt from a SQuALITY sample (Wang et al., 2022). The top cell shows two passages from the document, while the bottom two cells show parts of two human-written summaries. In total, each document in SQuALITY is paired with four human-written summaries.

OPEN AT STARS HOLLOW TOWN CENTER
[Miss Patty is trying to get a group of children to dance around a Maypole. Kirk watches from the side of the gazebo.]
MISS PATTY: No, no, boys. You go clockwise! Clockwise! Can't you tell time?
CHRISSY: The other way, boys! They're not getting it, Miss Patty.
MISS PATTY: Well, the wedding's tomorrow. We gotta get it.
CHRISSY: It's the Banyan boys. They won't do what I tell them.
KIRK: Nice maypole, Patty! Really organized!
MISS PATTY: Oh, shut up, Kirk!
LORELAI: Tough day, Patty?
MISS PATTY: I've worked with Joan Crawford. This is worse.
KIRK: I don't call that a "maypole." I'd call that a "maybe-not pole."
LORELAI: What's with him?
MISS PATTY: He's mad because I made Chrissy my dance captain over him.
LORELAI: Kirk has forty-three jobs.
MISS PATTY: Would you go talk to him, please? I got my hands full here.
LORELAI: The Banyan boys?
MISS PATTY: Oh! Lucifer tired of them in hell and dumped them here.
LORELAI: I'll talk to Kirk.
MISS PATTY: Thank you. All right, Chrissy, let's give it another go!
LORELAI: Hey, Kirk, maybe you want to ease up on Patty a little.
KIRK: But the maypole is an expertise of mine. I re-enacted the dwarf's maypole choreography from "The Safety Dance" video, my junior-high talent show. Chicks were falling at my feet. I'm less than impressed, Patty!
LORELAI: Take a break, please. I've seen Miss Patty get violent. It's not pretty. Remember that time?
KIRK: That's when she beat me up.
LORELAI: Yeah. Let's not repeat that.
KIRK: Okay.
[Jess sits on a nearby bench. He and Lorelai glance at each other as she passes. He looks like he's reading Punk Planet, but inside the magazine he's hiding You're Not Alone, one of the self-help books Luke gave him.]
CUT TO DRAGONFLY INN KITCHEN
JACKSON: They're the best I've got.
SOOKIE: That's sad for you and the whole vegetable industry.
JACKSON: They're the best in the state. I stand by them.
SOOKIE: They're puny. They're tasteless.
JACKSON: Puny? These are not puny.
SOOKIE: If they're small enough to shove up our son's nose, they're too small!
JACKSON: No way could you shove one of these up Davey's nose.
SOOKIE: Bet you five bucks.
JACKSON: Get him in here!
LORELAI: [entering] Hey, guys. You probably shouldn't shove a radish up your son's nose. Just thinking out loud.
SOOKIE: All right, I'll take these if it's all you've got...

As Stars Hollow prepares for the Renaissance-themed wedding of Liz and T.J., Kirk feels slighted when Miss Patty chooses someone else as the maypole dance captain; Lorelai discovers an equine visitor in the Dragonfly Inn's lobby; Rory's suite mates depart for summer vacation; Lorelai urges Mrs. Kim to call Lane; Lorelai unsuccessfully attempts to get Emily to admit to the separation during Friday night dinner; T.J. revels in the wonders of tights, but discovers their main drawback during the ceremony; Mrs. Kim gets some good advice from Lorelai after initially fleeing in horror at the sight of Zach and Brian and Lane's shabby apartment, and eventually returns for tea armed with a game plan; Rory calls Dean to rescue her from the disastrous date Emily has arranged for her with the son of a friend; Luke and Lorelai share significant glances during a slow dance at the wedding, which prompts Luke to ask Lorelai out on a date...

Figure 2.5: An excerpt from a SummScreen-FD sample (Chen et al., 2022). The top cell shows the TV show transcript, while the bottom shows the summary.

presents a challenge, as the dataset contains on average 7605 tokens per document and 114 tokens per summary.

Following prior work (Hua et al., 2023; Narayan et al., 2023), we use the FullDreaming (FD) subset, which contains 3673/338/337 examples for training, development, and test sets across 88 TV shows. The other fold provided in SummScreen, TVMegaSite, covers just 10 TV shows, increasing the risk of learning non-generalizable artifacts from the data. An example from the SummScreen-FD dataset is shown in Figure 2.5.

Similar to SQuALITY, we are interested in SummScreen due to the narrative structure captured by TV show transcripts and the extreme length of each transcript. The SummScreen summarization task is also highly abstractive, in the sense that transcripts have very low ngram overlap with summaries (Chen et al., 2022).

2.3.3 Evaluation Metrics

Summarization evaluation is generally subjective and new methodologies are continually proposed. In this thesis, we evaluate summarization systems along two aspects: **summary quality** and **faithfulness**. Summary quality measures the relevancy of information in the summary, while faithfulness measures the degree to which summary content is consistent with the source text. We measure summary quality with **ROUGE** (Lin, 2004), and faithfulness with a trio of recently published metrics: **AlignScore** (Zha et al., 2023), **QAFactEval** (Fabbri et al., 2022) and **FineSurE** (Song et al., 2024). We consider several faithfulness metrics as a key finding in Chapter 5 will show the three metrics are inconsistent with each other, and we therefore conduct a human evaluation as well.

2.3.3.1 ROUGE

ROUGE (Lin, 2004) is a standard set of summary quality metrics which measure word-level overlap with a reference summary. It is provided in several varieties which differ in how the overlap is computed. In ROUGE-N, the ngram overlap is computed between the system and reference summaries; for example, ROUGE-1 will measure word overlap, ROUGE-2 will measure bigram overlap, etc. In ROUGE-L, the longest common subsequence is computed between the system and reference summaries.

ROUGE is not the only metric for measuring summary quality; another popular metric is BERTScore (Zhang* et al., 2020). BERTScore measures similarity between a system and reference summary by computing the cosine similarity between their BERT

embeddings (Devlin et al., 2019). However, Deutsch and Roth (2021) finds BERTScore is not better than ROUGE at capturing summary quality, and we stick to ROUGE to more easily compare to contemporary works.

2.3.3.2 AlignScore

AlignScore (Zha et al., 2023) is an NLI-based metric for measuring factual consistency between predicted summaries and the source text. The scorer is a pre-trained language model, RoBERTa (Liu et al., 2019b), fine-tuned on NLI data and a variety of other domains such as paraphrasing and fact verification, unified into a single entailment framework. AlignScore splits the source document (i.e. the "context") into 350 token chunks and then predicts whether each predicted summary sentence (i.e. the "claims") is entailed by any chunk in the source. Since only one true entailment is needed to validate the predicted summary sentence, AlignScore takes the max value over all chunks in the source text for each predicted summary sentence. The final score is computed by taking the average value of these max entailment scores. Note that AlignScore relies solely on the predicted summary and source document, and therefore does not require any reference summary.

More formally, given a context $\boldsymbol{o}$ and a claim $\boldsymbol{l}$, the context $\boldsymbol{o}$ is split into chunks $\boldsymbol{o_i}$ and the claim is split into sentences $\boldsymbol{l_j}$. AlignScore is computed as:

$$\text{AlignScore}(\boldsymbol{o}, \boldsymbol{l}) = \underset{j}{\text{mean}} \max_{i} \text{alignment}(\boldsymbol{o_i}, \boldsymbol{l_j})$$

where the alignment() function is the probability of entailment from the fine-tuned RoBERTa model.

Zha et al. (2023) verify their metric on various factual consistency benchmarks including the SummaC (Laban et al., 2022) and TRUE (Honovich et al., 2022) datasets. They find AlignScore correlates better with human annotations compared to other recently proposed summarization metrics. In particular, AlignScore outperforms other popular NLI-based metrics such as SummaC-Conv (Zha et al., 2023).

2.3.3.3 QAFactEval

QAFactEval (Fabbri et al., 2022) is a QA-based metric measuring faithfulness between predicted summaries and source texts. Intuitively, QAFactEval generates question and answer pairs from the predicted summary, then uses the source document to verify the factuality of each answer. The proportion of correctly answered questions then serves

as the faithfulness score. It is implemented as a pipeline of several pre-trained models: a question-generation (QG) model, a question-answering (QA) model, and a scoring model to measure answer overlap. In their work, Fabbri et al. (2022) ablate each of these components; here we describe the best-performing metric.

1. **Answer Selection**: The goal of this step is to select units that can be compared by question answering. Fabbri et al. (2022) extract all NP chunks from the predicted summary as potential targets.

2. **Question Generation:** A question generation module is queried to generate a relevant question using the predicted summary as context and the selected answer from the previous step as input. Ideally, the question should be related to some salient aspect of the source document. QAFactEval uses BART, an encoder-decoder model (Lewis et al., 2020), fine-tuned on QA2D, a dataset of declarative sentences associated to question-answer pairs (Demszky et al., 2018) for this step.

3. **Question Answering:** A question answering model now answers the question from the previous step using the entire source document as input. The answer is seen as the source of truth for the generated question in the previous step. The authors find Electra, an extractive model which displays strong question-answering abilities (Clark et al., 2020), works best.

4. **Answer Overlap Evaluation:** The goal now is to compare the extracted answer from step 1 with the ground truth answer from step 3. QAFactEval uses a learned scoring model for this step, the LERC score (Chen et al., 2020). This model outputs a score from 1 – 5 corresponding to the degree the extracted answer from step 1 matches the expected answer from step 3. This step also checks if the question is answerable using the Electra model from step 3, and if not, the model is penalized and scores zero for this question.

5. **Question Filtering:** Lastly, since some questions may be noisy, QAFactEval filters out low-quality questions by checking if Electra can answer the question using the summary as context. If the question cannot be answered, it is not counted in the final score.

Finally, the scores from the remaining questions are averaged and reported as the QAFactEval score. Note that similar to AlignScore, QAFactEval relies solely on

the predicted summary and source document; the reference summary is not used for computing scores.

We opt to use QAFactEval over other QA-based faithfulness metrics such as QuestEval (Scialom et al., 2021) as Fabbri et al. (2022) demonstrate QAFactEval correlates more highly with human annotations on the SummaC benchmark (Laban et al., 2022).

2.3.3.4 FineSurE

FineSurE (Song et al., 2024) is a suite of LLM-based metrics measuring three summary-related dimensions: faithfulness, conciseness and completeness. We focus on the faithfulness dimension as it is most relevant to our motivation in Chapter 5. FineSurE measures faithfulness by enumerating sentences in the system summary and asking the LLM whether each sentence is supported by the source document. If the sentence is not supported, the LLM outputs an error category (out of 8 possible errors) and a short reasoning. Determining each sentence's factuality, error category and reasoning is handled with a single call to the LLM; the exact prompt can be found in Appendix A. Although Song et al. (2024) mainly experiment with GPT-4 (OpenAI et al., 2024), we use Claude due to restrictions in our use of available LLMs. As in AlignScore and QAFactEval, FineSurE does not require reference summaries for measuring model performance. Song et al. (2024) verify their method correlates strongly with human annotations on two summarization benchmarks: FRANK (Pagnoni et al., 2021) and REALSumm (Bhandari et al., 2020).

Several other LLM-based summarization metrics have been proposed recently. Two other recently published works are PRisma (Mahon and Lapata, 2024) and G-Eval (Liu et al., 2023c).

PRisma measures summary quality by extracting atomic facts from the reference and system summaries, then for each atomic fact, determines if it is present in the corresponding system/reference summary. However, PRisma independently checks whether each atomic fact is supported by the context, meaning repeated calls are needed for each summary. We initially experimented with PRisma but found the repeated LLM calls both rapidly exceeded our budget and required long evaluation times. In contrast, FineSurE only requires a single call for each summary.

G-Eval evaluates system summaries by first prompting the LLM to design the evaluation criteria for the task, then prompting it to output scores from 1 to 5 along the desired criteria. The final score is a sum of the possible score tokens weighted by their probabilities according to the LLM. Compared to FineSurE and PRisma, this method is

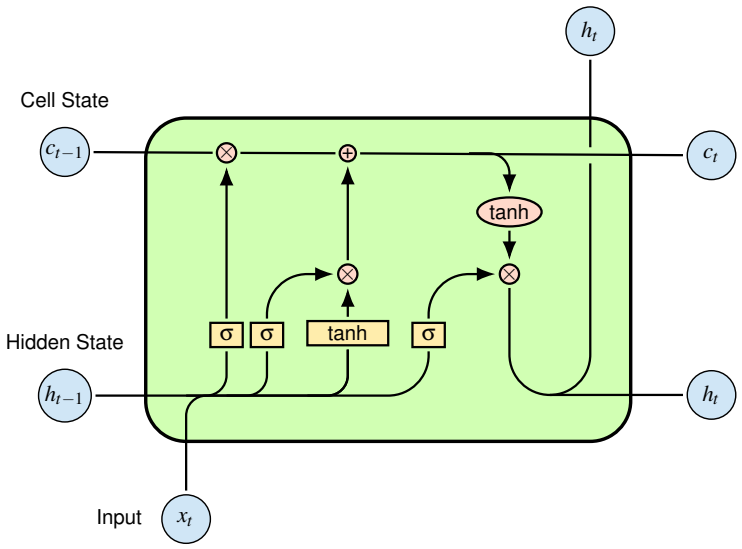


Figure 2.6: A visualization of the LSTM cell. Pink shaded parts denote pointwise operations, while the yellow boxes represent learnable parameters, with activation types labelled.[1]

more opaque, as the user cannot inspect the predicted factuality of individual sentences or atomic facts in the summary. Moreover, FineSurE achieves comparable performance on the FRANK benchmark (Pagnoni et al., 2021) and better performance than G-EVal on the REALSumm benchmark (Bhandari et al., 2020).

2.4 Base Model Architectures

In this section, we detail the various neural network architectures we use throughout the thesis.

2.4.1 Long Short-Term Memory Networks (LSTMs)

Long Short-Term Memory Networks (Hochreiter and Schmidhuber, 1997) are a class of neural networks well-suited for sequential tasks. They are designed to alleviate issues with recurrent neural networks (RNNs), particularly with gradients that tend to either explode or vanish (Bengio et al., 1994).

The equations governing the LSTM architecture are designed to learn long-distance dependencies. They consist of a series of gates, to either 'forget' non-essential information, 'update' the LSTM cell with new information, or 'output' the model's prediction for the current time step. Another key feature of LSTMs is that the model's parameters

are broken up into a **cell state** and a **hidden state**. The cell state records long-term contextual information, while the hidden state holds more immediate representations.

We briefly explain the equations governing the LSTM. In the following, W and b describe learnable parameters, x_t is an input vector, h_{t-1} is previous hidden state and C_{t-1} is the previous cell state.

The first step is to compute how much information from the context should be forgotten. This is achieved with the forget gate f_t:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Next, we compute the input gate layer which decides how much new information is added to the cell state, and we create new candidate values $\tilde{C}_t$ to update the cell state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

The cell state is now updated by deciding how much to forget the old state (left part of the summation), and how much to update by (right part):

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

where $*$ is a pointwise multiplication. Finally, the output is computed by first computing an output gate, o_t. The output gate then decides which parts of the cell state we want to output as the new hidden state, as shown below:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * \tanh(C_t)$$

These equations define one step of the LSTM. The overall process is repeated for each step in the input sequence. A visualization of the architecture is shown in Figure 2.6.

[1] Adapted from https://tex.stackexchange.com/a/432344 and https://colah.github.io/posts/2015-08-Understanding-LSTMs. Accessed on May 12, 2025.

2.4.2 Stack-LSTM

Stack-LSTMs (Dyer et al., 2015) are a variation on the LSTM architecture, adapting them to serve as a stack for parsers. The key innovation is to add a 'Pop' operation in the form of a stack pointer. If a Pop operation is triggered, the stack pointer moves back to the previous LSTM cell. Subsequent elements always extend from the current pointer location, rather than from the right-most element of the sequence. This technique is well-suited for modelling stack operations in natural language processing (such as in parsing), and we make use of this architecture for tracking the left boundary of mentions in Chapter 3.

2.4.3 Encoder-Decoder

The Encoder-Decoder architecture, also known as seq2seq, is a widely used architecture for a variety of NLP tasks. First proposed for machine translation (Cho et al., 2014; Sutskever et al., 2014), they can be adapted to many sequence-to-sequence tasks, and we use this formulation for coreference resolution in Chapter 5.

The encoder-decoder architecture consists of an encoder, which maps the inputs $x_1, \ldots, x_n$ to a hidden representation $h_1, \ldots, h_k$, and a decoder, which maps $h_1, \ldots, h_k$ to the desired outputs $y_1, \ldots, y_m$.

More formally, given input vectors $x_1, \ldots, x_n$, the encoder maps the inputs to a hidden representation:

$$h_1, \ldots, h_k = \text{Encoder}(x_1, \ldots, x_n)$$

The decoder then generates the outputs from the hidden representation:

$$y_1, \ldots, y_m = \text{Decoder}(h_1, \ldots, h_k)$$

The exact architecture of the encoder and decoder may vary; for example, they may be LSTMs, as in Cho et al. (2014) and Sutskever et al. (2014), or transformer networks, as we describe in the following section.

2.4.4 Transformer

The transformer architecture (Vaswani et al., 2017) is an encoder-decoder model where the encoder and decoder are formed of components called transformer layers. At the core of each transformer layer is the attention mechanism (Bahdanau et al., 2015).

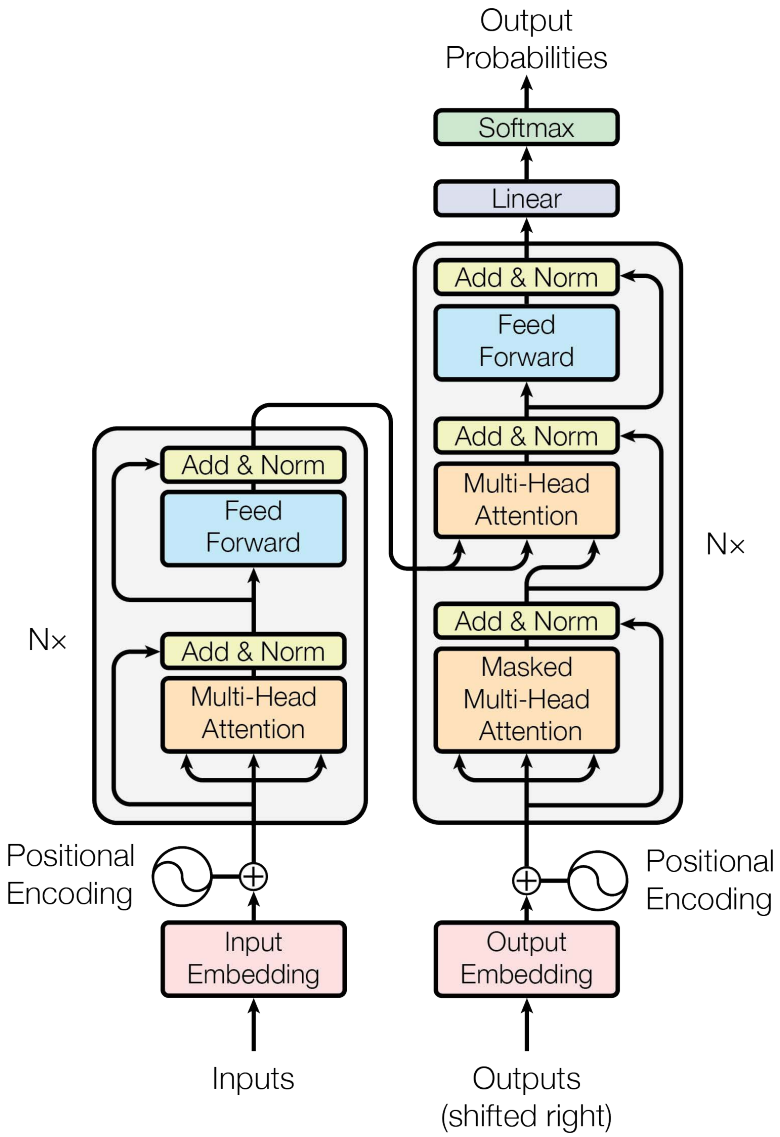


Figure 2.7: A visualization of the transformer architecture, from Vaswani et al. (2017). The encoder is shown as the left block, while the decoder is on the right side. Each transformer block consists of multiple components, such as the multi-head attention, layer normalization, full feed-forward network, and residual connections. The N denotes that each block is repeated N times in the encoder and decoder.

In the attention mechanism, inputs $X = (x_1, \ldots, x_n)$ are mapped through linear transformations to three matrices, called the query matrix Q, key matrix K and value matrix V. The output of the attention mechanism is then given by:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where d_k is the dimension of the key matrix.

In essence, a similarity score is computed between Q and K (i.e. through matrix multiplication), and then transformed via the softmax function into a probability distribution, called the attention weights or attention distribution. The final output is the sum of vectors in V weighted by the probabilities in the attention weights. The division by $\sqrt{d_k}$ prevents the multiplication of Q and K from growing too large in magnitude.

The attention mechanism is used in different ways in the transformer architecture. In the encoder, it is implemented as *self-attention*, where the key, value and query matrices all originate from the inputs or previous encoder layers. In the decoder, self-attention is used alongside *cross-attention*, where the query matrix comes from the labels (or the outputs of the previous decoder layer), and the key and value matrices are the encoder's outputs.

Vaswani et al. (2017) also found it beneficial to separate the attention mechanism into what they call multi-head attention. The key, value, and query matrices are each linearly projected into distinct representations. Each head is learned independently and computed in parallel. After the attention mechanism is computed for each head, the outputs are concatenated and projected back to the original dimension.

While the attention mechanism is the main component of the transformer, there are other parts to the full architecture, such as layer normalization, full feed-forward networks, and residual connections, which we do not cover here. The full diagram is shown in Figure 2.7.

Lastly, transformers also include a positional encoding scheme. In LSTMs, processing is sequential (i.e. leftmost token to rightmost), so the token's position in the sequence is captured by the LSTM cells. However, in transformers, the transformer layers compute the attention mechanism across all tokens simultaneously, which does not include any information about their position in the input (or output) sequence. To avoid this issue, Vaswani et al. (2017) add positional encodings for each token embedding, based on sinusoidal functions.

2.4.5 Transformer-XL

One disadvantage of the basic transformer architecture is that the dimensions of the attention matrices (e.g. Q, K and V) are fixed, meaning that the model has a maximum output sequence length. This feature can present issues if the text length stretches longer than the allowed limit. Transformer-XL (Dai et al., 2019) aims to solve this issue. Its architecture is a variation on the basic transformer, allowing it to extend to infinite sequence length at inference time. The architecture is convenient for our use case because it allows us to efficiently run incremental settings without re-inputting the entire sequence for every new sentence or text chunk.

There are two modifications made to the transformer architecture in order to achieve this goal. The first change is a specialized architecture that caches and reuses previously computed segments during training and inference. This is implemented by introducing a segment-level recurrence mechanism to the transformer layer, allowing the hidden states of a new segment to attend to the previous one. This change allows inputting new text without re-inputting previously processed tokens, which we will exploit for our incremental system in Chapter 3.

The second change is the positional encoding in the transformer is replaced with a *relative* position encoding. Instead of adding an absolute positional encoding to each token embedding, Transformer-XL adds a relative positional encoding when computing the attention scores between key and query matrices. The relative positional encoding is associated with the token distance between the query and key representation, rather than an absolute position.

2.5 Pre-trained Language Models (PLMs)

Many successful approaches in NLP have not only benefitted from advances in neural architectures, but also from pre-training on large data corpora. This technique originates from static word embeddings such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). These methods train shallow neural networks to predict a target word from its surrounding context; however, they do not distinguish how meanings of words may differ in different contexts or syntactic positions. Later, ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) proposed contextualized word embeddings by predicting a distinct embedding for each word depending on its context. In ELMo, this goal is achieved with bidirectional LSTMs; in BERT, with a transformer

encoder. In Chapter 3, we use XLNet (Yang et al., 2019), a pre-trained language model extending BERT with a modified pre-training objective. Importantly, XLNet uses the Transformer-XL architecture, meaning it can efficiently handle an incremental setting.

Pre-training has also found utility in encoder-decoder architectures. BART (Lewis et al., 2020) introduced pre-training tasks in the discourse domain (i.e. beyond token prediction) such as permuting sentence order and rotating the document start token. In T5 (Raffel et al., 2020), the authors convert a variety of NLP tasks into natural language prompts, requiring the model to learn to solve various NLP tasks based on the text prompt alone. The T5 model motivated the T0 (Sanh et al., 2022a) encoder-decoder model, which we rely on in Chapter 5 as our base model. The T0 model expands the T5 pre-training scheme to a much larger set of natural language pre-training tasks.

The dominant architecture for large-scale pre-training is the decoder-only transformer. Many popular chat applications such as Anthropic's Claude (Anthropic, 2024), Google's Gemini (Comanici et al., 2025) and OpenAI's GPT-4 (OpenAI et al., 2024) use decoder-only transformers, and the term large language model (LLM) typically refers to models based on this architecture. In Chapter 5, we explore using Claude Sonnet 3.5 (Anthropic, 2024), a decoder-only LLM released in June 2024. Sonnet 3.5's specific training and model details are not publicly available; however, Sonnet 3.5 is known to outperform many other LLM models on a wide variety of benchmarks such as reasoning, math, coding, reading comprehension and question answering (Anthropic, 2024). Its strong performance on many NLP benchmarks motivates us to use it in Chapter 5 as an oracle to generate sub-event plans.

In this thesis, we will also make a distinction between *small* and *large* language models, which we refer to as SLMs and LLMs. This distinction is ever-changing, but in this thesis, we will define SLMs as having less than 4 billion parameters, and LLMs as any larger model. The distinction is important as many works have found performance improves significantly as the model size and the amount of pre-training data increase. For example, the GPT (Generative Pre-trained Transformer) model series (Radford et al., 2018, 2019; Brown et al., 2020) share the same decoder-only transformer architecture, but each successive model enlarges the model size and the amount of pre-training data, with each iteration resulting in improved performance. The performance improvement from GPT-2 (Radford et al., 2019) to GPT-3 (Brown et al., 2020) is particularly remarkable, as it allows for a learning technique called in-context learning. Unlike the traditional supervised fine-tuning paradigm, in-context learning is an inference-time learning approach where examples are provided directly

in the model's context (or prompt). This technique is much simpler than traditional fine-tuning, only requiring a small amount of samples (sometimes as few as 1 or 2), but can seemingly only be leveraged by larger models such as GPT-3.

While LLMs' powerful performance is attractive, SLMs are interesting in their own right, as LLMs' large size is often prohibitive to run or train, especially on smaller devices such as mobile phones. Getting SLMs to work as well as LLMs is a challenge we explore in Chapter 5, using Claude Sonnet 3.5 (Anthropic, 2024) as the LLM and Phi-mini-3.5 (Abdin et al., 2024) as the SLM.

Phi-mini-3.5 is a 3.8 billion parameter, decoder-only transformers model pre-trained on 3.3 trillion tokens. Abdin et al. (2024) show that Phi-mini-3.5 performs comparably to much larger models despite a much smaller parameter count. It also boasts a 128K context window, which is important for the long document tasks we examine in Chapter 5. Based on these two advantages, we select Phi-3.5-mini as the SLM in our experiments.

Chapter 3

Sentence-Incremental Neural Coreference Resolution

In this chapter, we focus on encoder-based coreference resolution systems, e.g. models that rely on BERT (Devlin et al., 2019) or other encoders for their processing. Our hypothesis is that **building a discourse representation is an efficient and effective strategy for incremental coreference resolution in encoder-based models**. In this chapter, we represent discourse as a memory matrix of entities encountered so far, using their hidden representations.

We show that existing coreference resolution systems are not well-adapted for incremental settings such as dialogue, and fall into one of two groups:

1. State-of-the-art non-incremental models that incur quadratic complexity in document length with high computational cost, and

2. Memory network-based models which operate incrementally but do not generalize beyond pronouns.

Focusing on the first approach, we suggest a new coreference resolution evaluation setting requiring systems to process coreference clusters sentence-by-sentence, and show that existing systems neither efficiently nor accurately handle this setting. We propose an incremental system based on shift-reduce parsing which incrementally builds entity clusters, and show it outperforms state-of-the-art systems in an incremental setting while efficiently processing text. A key contribution of our approach is a mention detection module that operates in $O(n)$ span complexity, where n is the number of tokens in the document.

3.1 Introduction

In the previous chapter, we discussed how coreference resolution is typically performed in two steps: in *mention detection*, the model predicts which expressions are referential, and in *mention clustering*, the model computes each mention's antecedent. Many systems follow a mention-pair formulation from Lee et al. (2017), in which all possible spans are ranked and then scored against each other. In particular, methods that augment this approach with large, pre-trained language models achieve state-of-the-art results (Joshi et al., 2019, 2020).

Despite impressive performance, these methods are computationally demanding. For a text with n tokens, they will score up to $O(n^2)$ spans, followed by up to $O(n^4)$ span comparisons. They also process documents non-incrementally, requiring access to the entire document before processing can begin. These properties present challenges when insufficient computational resources are available, or when the task setup is incremental, such as in dialogue (e.g. Khosla et al. 2021). From a cognitive perspective, these methods are also unappealing because research on "garden-path" effects show that humans resolve referring expressions incrementally (Altmann and Steedman, 1988).

These drawbacks sparked renewed interest in incremental coreference resolution systems, in which document tokens are processed sequentially. These approaches use memory networks to track entities in differentiable memory cells (Liu et al., 2019a; Toshniwal et al., 2020a). These models demonstrate proficiency at proper name and pronoun resolution (Webster et al., 2018). However, they seem unlikely to generalize to more complicated coreference tasks due to a strict interpretation of incrementality. Both Liu et al. (2019a) and Toshniwal et al. (2020a) resolve mentions word-by-word, making coreference decisions possibly before the full noun phrase has been observed. The approach is adequate for proper names and pronouns, but it may fail to distinguish entities who share the same phrase prefix. For example, in Figure 3.1, three mentions all begin with 'Hong Kong', though all belong to separate entities. In this case, it is difficult to see how a system using word-level predictions would resolve these mentions to different entities.

Motivated by this work, we propose a system that processes a document incrementally at the sentence-level, creating and updating coreference clusters after each sentence is observed. The system addresses deficiencies in memory network-based approaches by delaying mention clustering decisions until the full mention has been observed. These goals are achieved through a mention detector based on shift-reduce parsing,

In 2004, on the Waterfront Promenade originally constructed for viewing only the scenery of Hong Kong Island and Victoria Harbor, the Hong Kong Tourism Board also constructed the Avenue of Stars, memorializing Hong Kong's 100-year film history.

Figure 3.1: An example from the OntoNotes dataset which highlights the need for incremental systems to identify spans rather than tokens as mentions. The mentions cannot be resolved solely from the prefix 'Hong Kong', and the clustering decision should be delayed until the full mention is observed.

which identifies mentions by marking left and right mention boundaries. Mention candidates are then passed to an online mention clustering model similar to Toshniwal et al. (2020b) and Xia et al. (2020). The model proposes a linear number of spans per sentence, reducing computational requirements and maintaining more cognitive plausibility compared to non-incremental methods.

In order to compare non-incremental and incremental systems on equal footing, we propose a new **sentence-incremental** evaluation setting. In this setting, systems receive sentences incrementally and must form partial coreference chains before observing the next sentence. This setting mimics human coreference processing more closely, and is a more suitable evaluation setting for downstream tasks in which full document access is generally not available (e.g. for dialogue (Andreas et al., 2020)).

Using the sentence-incremental setting, we demonstrate that our model outperforms comparable systems adapted from partly incremental methods (Xia et al., 2020) across two corpora, the OntoNotes dataset (Pradhan et al., 2012) and the CODI-CRAC 2021 corpus (Khosla et al., 2021). Moreover, we show that in a conventional evaluation setting, where the model can access the entire document, our system retains close to state-of-the-art performance. However, the sentence-incremental setting is substantially outperformed by non-sentence-incremental systems. Analyzing the difference between these two settings reveals that the encoder is heavily dependent on how many sentences it can observe at a time. The analysis suggests better representations of the entities and their context may improve performance in the sentence-incremental setting. Nevertheless, our results in this chapter provide state-of-the-art baselines for sentence-incremental evaluation.

3.2 Background and Related Work

Non-incremental mention-pair models have dominated the field in recent years, with many following the formulation presented by Lee et al. (2017). Several extensions have led to performance improvements, such as adding higher-order inference (Lee et al., 2018), and replacing the encoder with BERT and SpanBERT (Joshi et al., 2019, 2020). Extensions to this approach have looked at reformulating the problem as question-answering (Wu et al., 2020), simplifying span representations (Kirstain et al., 2021), and incorporating coherence signals from centering theory (Chai and Strube, 2022). Lee et al. (2017)'s approach is high performing but computationally demanding, requiring $O(n^4)$ span comparisons for a document with n tokens. Table 3.1 compares Lee et al. (2017)'s span complexity with our proposal, highlighting the considerable reduction in the number of span comparisons.

Toshniwal et al. (2020b) and Xia et al. (2020) adapt the non-incremental system of Joshi et al. (2020) so that mention clustering is performed incrementally. However, in their formulation, document encoding, mention detection and certain clustering decisions still fully depend on Joshi et al. (2020). This dependency means these 'part-incremental' systems still incur a quadratic span complexity, as shown in Table 3.1. While our mention clustering component similarly incrementally builds entity clusters, we demonstrate how the remaining pipeline (i.e. the document encoder and mention detection components) can also be incrementalized. Yu et al. (2020) similarly present an incremental mention clustering approach where mention detection is performed non-incrementally as Lee et al. (2017).

Memory network-based approaches identify co-referring expressions by writing and updating entities into cells within a fixed-length memory (Liu et al., 2019a; Toshniwal et al., 2020a). These models demonstrate how fully incremental coreference systems can be achieved. However, the formulation operates on token-level predictions, and does not easily extend to either nested mentions or certain multi-token mentions (e.g. in Figure 3.1).

Cross-document coreference resolution (CDCR) requires systems to compute coreference chains across documents, raising scalability challenges as the number of documents increases. Given these challenges, incremental CDCR systems are crucial (Allaway et al., 2021; Logan IV et al., 2021) due to lower memory requirements. However, these works are not directly comparable to ours since they assume gold mentions are provided as input.

Model	Incremental Components	Span Complexity
SpanBERT	None	$O(n^4)$
longdoc	Mention Clustering	$O(n^2m)$
ICoref	Mention Clustering	$O(n^2m)$
Part-Inc (Ours)	Mention Detection + Mention Clustering	$O(nm)$
ICoref-*inc*	All	$O(n^2m)$
Sent-Inc (Ours)	All	$O(nm)$

Table 3.1: The list of systems we compare, alongside their incrementality (on a sentence-level) and span complexity. 'All Components' means document encoding, mention detection and mention clustering. n is the number of tokens and m is the number of entities.

Other, earlier, incremental coreference systems also often ignore or diminish the role of mention detection. For example, Webster and Curran (2014) use an external parser for mention detection, requiring an additional model. Klenner and Tuggener (2011) assume gold mentions as input.

Our incremental mention detector bears similarities to certain models for nested named-entity recognition (NER). In particular, Wang et al. (2018) present an incremental neural model for nested NER based on a shift-reduce algorithm. Their deduction rules differ greatly from ours as they model mention spans using complete binary trees, and are aimed at NER rather than mention detection.

3.3 Method

Given a document, the goal is to output a set of clusters $\mathcal{C} = \{C_1, \ldots, C_K\}$, where mentions within each cluster are co-referring. We assume mentions may be nested but otherwise do not overlap. This assumption allows us to model mentions using a method analogous to shift-reduce, where shifting corresponds to either incrementing the buffer index or marking a left mention boundary, and reducing corresponds to marking a right boundary and resolving the mention to an entity cluster.

3.3.1 Shift-Reduce Framework

The main idea is to mark mention boundaries using PUSH, POP or PEEK actions, or to pass over a non-boundary token with the ADVANCE action. After POP or PEEK actions, a mention candidate is created using the current top-of-stack and buffer elements. The resulting mention candidate is then either resolved to an existing cluster or initialized as a new entity cluster.

We represent the state as $[S, i, A, \mathcal{C}]$, where S is the stack, i is the buffer index, A is the action history and $\mathcal{C}$ is the current set of clusters. At each time step, one of four actions is taken:

- PUSH: Place the word at buffer index i on top of the stack, marking a left mention boundary. Figure 3.2 represents this formally as adding token w_i to the stack S. Figure 3.3 shows two examples of the PUSH action marking left mention boundaries at tokens 'Auto' and 'their'.

- ADVANCE: Move the buffer index forward. Figure 3.2 denotes this action as advancing the buffer index i to $i+1$. Figure 3.3 demonstrates moving the buffer forward in steps 2, 4, 5, 8 and 10.

- POP: Remove the top element from S and create a mention candidate using this element and the current buffer element. Score the candidate against existing clusters and resolve it (or create a new cluster). Figure 3.2 represents this action as removing the top element v (i.e. the left mention boundary) from the existing stack $S|v$, then resolving the mention with boundaries (v, w_i) to $\mathcal{C}$. Figure 3.3 provides two examples of the POP action: in step 3, the token 'Auto' is removed from the stack to create the mention 'Auto workers', while in step 9, the token 'their' is removed from the stack to create 'their strike'.

- PEEK: Create a mention candidate using the top element on the stack and the current buffer element, without removing the top element from the stack. Score the candidate against existing clusters and resolve it (or create a new cluster). Figure 3.2 expresses this action as maintaining the stack $S|v$ (without change), while resolving the mention (v, w_i) to $\mathcal{C}$. Figure 3.3 shows how this action is used in step 7. The token 'their' remains on the stack, while the system resolves the mention 'their' to the cluster containing 'Auto workers'.

The PEEK action does not alter the stack but is otherwise identical to POP. This action is critical for detecting mentions sharing a left boundary.

$$\text{Initial} \quad [\varnothing, 0, \varnothing, \varnothing]$$

$$\text{Final} \quad [\varnothing, n, A, \mathcal{C}]$$

$$\textsc{Push} \quad \frac{[S, i, A, \mathcal{C}]}{[S|w_i, i, A|\textsc{Push}, \mathcal{C}]}$$

$$\textsc{Advance} \quad \frac{[S, i, A, \mathcal{C}]}{[S, i+1, A|\textsc{Advance}, \mathcal{C}]}$$

$$\textsc{Pop} \quad \frac{[S|v, i, A, \mathcal{C}]}{[S, i, A|\textsc{Pop}, \mathcal{C}|\textsc{Coref}(v, w_i)]}$$

$$\textsc{Peek} \quad \frac{[S|v, i, A, \mathcal{C}]}{[S|v, i, A|\textsc{Peek}, \mathcal{C}|\textsc{Coref}(v, w_i)]}$$

Figure 3.2: Deduction rules for our coreference resolver. $[S, i, A, \mathcal{C}]$ denotes the stack S, buffer index i, action history A, and cluster set $\mathcal{C}$. The COREF function indicates that span (v, w_i) is clustered and added to $\mathcal{C}$. $S|v$ means the stack S with token index v on top.

Several hard action constraints ensure that only valid actions are taken and the final state is always reached. For example, PUSH can only be called once per token, or else the model would be marking the left boundary multiple times. The full list of constraints is described in Section 3.3.1.1.

We denote the set of valid actions as $\mathcal{V}(S, i, A, \mathcal{C})$. The conditional probability of selecting action a_t based on state $\mathbf{p}_t$ can then be expressed as:

$$p_M(a_t | \, \mathbf{p}_t) = \frac{\exp(w_{a_t} \cdot f_M(\mathbf{p}_t))}{\sum_{a' \in \mathcal{V}(S,i,A,\mathcal{C})} \exp\left(w_{a'} \cdot f_M(\mathbf{p}_t)\right)},$$

where f_M is a two-layer neural network, and w_{a_t} is a column vector selecting action a_t.

If POP or PEEK operations are predicted, the mention candidate is then scored against existing clusters. Depending on these scores, the mention is either (a) resolved to an existing cluster, or (b) initialized as a new entity cluster. Define the set of possible coreference actions as $\mathcal{A}_k$, which includes resolving to existing clusters $\{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$ and creating new cluster $\mathcal{C}_{k+1}$. We can write the conditional probability of coreference

	Action	Stack	Buffer	Clusters
1.	PUSH	[Auto]	Auto workers ended their strike	∅
2.	ADVANCE	[Auto]	workers ended their strike	∅
3.	POP	[∅]	workers ended their strike	{ Auto workers }
4.	ADVANCE	[∅]	ended their strike	{ Auto workers }
5.	ADVANCE	[∅]	their strike	{ Auto workers }
6.	PUSH	[their]	their strike	{ Auto workers }
7.	PEEK	[their]	their strike	{ Auto workers, their }
8.	ADVANCE	[their]	strike	{ Auto workers, their }
9.	POP	[∅]	strike	{ Auto workers, their } { their strike }
10.	ADVANCE	[∅]	∅	{ Auto workers, their } { their strike }

Figure 3.3: Example of the shift-reduce system for the sentence "Auto workers ended their strike". ∅ denotes the empty stack or empty cluster set. Expressions within brackets mean they are co-referring. In each step, the Stack and Buffer show the result of applying the given action.

prediction z_j based on mention candidate m_j as:

$$p_C(z_j|\, m_j) = \frac{\exp(w_{z_j} \cdot s_C(m_j))}{\sum_{z' \in \mathcal{A}_k} \exp(w_{z'} \cdot s_C(m_j))},$$

where s_C is a function scoring the mention candidate against $\{\mathcal{C}_1, \ldots, \mathcal{C}_k, \mathcal{C}_{k+1}\}$ (described in Section 3.3.2.2).

The terminal state is reached when the final buffer element has been processed and the stack is empty. At this point, all mentions have been clustered and we return all non-singleton entity clusters.

3.3.1.1 Action Constraints

To ensure the final state is always reached, it is necessary to enforce a set of rules during mention detection:

1. ADVANCE can only be called on the final token if the stack is empty.
2. POP and PEEK can only be called if the stack is non-empty.
3. PUSH can only be called once per token, ensuring that left boundaries are only marked once.

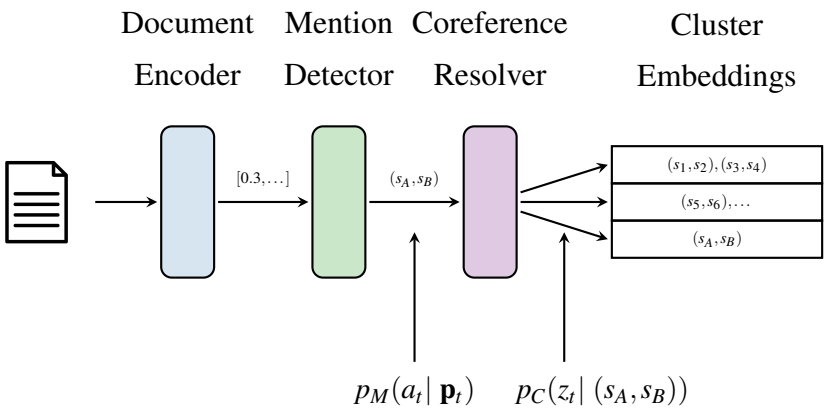


Figure 3.4: A summary of the overall algorithm. After document encoding, the mention detector predicts transition actions PUSH, POP, PEEK or ADVANCE using the parser state $\mathbf{p}_t$. If a mention is predicted, the coreference resolver then clusters it to an existing cluster representation or creates a new cluster. Clustering a mention implies a coreference relation with mentions in the cluster. The steps can all be performed incrementally, assuming the document encoder is also incremental.

4. PUSH cannot directly follow POP or PEEK. Allowing this action sequence would either admit multiple paths to the same mention or non-nested overlapping mentions.

5. POP cannot directly follow PEEK, or else the same mention would be proposed twice.

6. PEEK cannot be called on the final token. This action would imply the stack is non-empty on the final token, and that POP must be called.

3.3.2 Neural Implementation

3.3.2.1 Mention Detector

Document tokens are first encoded using a pre-trained language model. The concatenated word embeddings, $x_1, \ldots, x_n$, form the **buffer** for the shift-reduce mechanism. Assuming current word x_i and time step t, we denote the buffer as $\mathbf{b}_t = x_i$.

The stack is represented using a Stack-LSTM (Dyer et al., 2015). Let $x_{s_1}, \ldots, x_{s_L}$ be the currently marked left mention boundaries pushed to the stack. Then the **stack** representation at time t is:

$$\mathbf{s}_t = \text{StackLSTM}[x_{s_1}, \ldots, x_{s_L}].$$

We encode the action history $a_0, \ldots, a_{t-1}$ with learned embeddings for each of the four actions. The action history at t is encoded with an LSTM over previous action embeddings:

$$\mathbf{a}_t = \text{LSTM}[a_0, \ldots, a_{t-1}].$$

Then, the parser state is represented by the concatenation of buffer, stack, action history and additional mention features ϕ_M:

$$\mathbf{p}_t = [\mathbf{b}_t; \mathbf{s}_t; \mathbf{a}_t; \phi_M(\mathbf{b}_t, \mathbf{s}_t)],$$

where ϕ_M denotes learnable embeddings corresponding to useful mention features such as span width and document genre. For span width, we use embeddings measuring the distance from the top of the stack to the current buffer token (i.e. $i - s_L$), or 0 if the stack is empty.

3.3.2.2 Mention Clustering Model

The mention clustering is similar to previous online clustering methods (Toshniwal et al., 2020b; Xia et al., 2020; Webster and Curran, 2014), though we take care to avoid dependence on non-incremental pre-trained language models which have already been fine-tuned to this task.

Given a mention candidate's span representation v, we score v against the existing entity cluster representations $\mathbf{m}_1, \ldots, \mathbf{m}_k$:

$$\begin{aligned} s_C(v) &= [f_C(\mathbf{m}_1, v), \ldots, f_C(\mathbf{m}_k, v), \alpha] \\ f_C(\mathbf{m}_i, v) &= \text{MLP}([v, \mathbf{m}_i, v \odot \mathbf{m}_i, \phi_C(v, \mathbf{m}_i)]) \\ i^* &= \underset{i \in \{1, \ldots, k+1\}}{\arg\max}\ s_C(v) \end{aligned}$$

where f_C is two-layer neural network, α is a threshold value for creating a new cluster, $v \odot \mathbf{m}_i$ is the element-wise product and ϕ_C encodes useful features between v and $\mathbf{m}_i$: the number of entities in $\mathbf{m}_i$, mention distance between v and $\mathbf{m}_i$, the previous coreference action and document genre.

If the scores between v and all cluster representations $\mathbf{m}_1, \ldots, \mathbf{m}_k$ are below some threshold value α (i.e. $i^* = k+1$), we initialize a new entity cluster with v. Otherwise, we update the cluster representation $\mathbf{m}_{i^*}$ via a weighted average using the number of entities represented by $\mathbf{m}_{i^*}$:

$$\mathbf{m}_{i^*} \leftarrow \beta \cdot \mathbf{m}_{i^*} + (1-\beta) \cdot v,$$

where $\beta = \frac{|\mathbf{m}_{i^*}|}{|\mathbf{m}_{i^*}|+1}$ is the weighting term.

3.3.2.3 Training

Training is done via teaching forcing. At each time step, the model predicts the gold action given the present state. The state is then updated using the gold action. At each step, we compute mention detection loss $\mathcal{L}_M$ and coreference loss $\mathcal{L}_C$.

The mention detection loss $\mathcal{L}_M$ is calculated using the cross-entropy between the predicted mention detection action and gold action $a_{t^*} \in \mathcal{V}(S,i,A,\mathcal{C})$:

$$\mathcal{L}_M = -\sum_t \log p_M(a_{t^*} \mid \mathbf{p}_t),$$

where t sums over time steps across all documents.

Similarly, the coreference loss $\mathcal{L}_C$ is defined by the cross entropy between the highest-scoring coreference action and the gold coreference decision $z_{j^*} \in \mathcal{A}_k$:

$$\mathcal{L}_C = -\sum_j \log p_C(z_{j^*} \mid m_j),$$

where j sums over mentions across all documents. The entire network is then trained to optimize the sum of the two losses, $\mathcal{L}_M + \mathcal{L}_C$. During inference, we predict actions using greedy decoding, updating the state solely with predicted actions.

Figure 3.4 presents a summary of the various components and the overall algorithm.

3.4 Experiments

3.4.1 Datasets

We train and evaluate our system on the **OntoNotes 5.0** dataset (Weischedel et al., 2013), described in Section 2.2.2.1. We use the same setup as the CoNLL-2012 Shared Task (Khosla et al., 2021). We evaluate using the *MUC* (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998) and $CEAF_{\phi_4}$ (Luo, 2005) metrics and their average (the CoNLL score), using the official CoNLL-2012 scorer.

We also test models on the **CODI-CRAC 2021** corpus (Khosla et al., 2021) described in Section 2.2.2.2. The dataset suits incremental systems well since dialogue can be naturally presented as incremental utterances. Given the small dataset size, we

use it for evaluation only, using models trained on OntoNotes. Since OntoNotes marks document genre (which systems often use as a feature), we associate CODI-CRAC documents with OntoNotes' 'telephone conversation' genre, since it is the most similar. We remove singleton clusters due to lack of annotation in the training set. We again evaluate using *MUC*, B^3 and $CEAF_{\phi_4}$, using the official Universal Anaphora scorer (Yu et al., 2022b).

3.4.2 Model Components

3.4.2.1 Document Encoder

Many coreference resolution models use SpanBERT (Joshi et al., 2020) as a base model (Wu et al. (2020); Toshniwal et al. (2020b); Xia et al. (2020); Xu and Choi (2020), among others), since Joshi et al. (2020) demonstrate SpanBERT's proficiency for entity-related tasks such as coreference resolution. However, SpanBERT is unsuitable for incremental applications because it expects all its input simultaneously and cannot partially process text while waiting for future input.

Instead, we turn to XLNet[1] (Yang et al., 2019), which extends the earlier Transformer-XL model (Dai et al., 2019). As we described in Sections 2.4.5 and 2.5, XLNet differs from typical pre-trained language model encoders as it can efficiently cache and reuse its previous outputs. The caching mechanism allows for recurrent computation to be performed efficiently. Cached outputs provide a context to the current sentence being processed.

We experiment using XLNet in two settings: in the **Sentence-Incremental** (Sent-Inc) setting, each sentence is processed sequentially, and partial coreference clusters are computed before the next sentence is observed. After each sentence is processed, we accumulate XLNet's outputs (up to a cutoff point) and reuse them when processing the next sentence. We limit the number of cached tokens so that the cached and 'active' tokens do not exceed 512, so that our work remains comparable to other recent works. Although the mention detector is token-incremental and the mention clustering component is span-incremental, the document encoder is sentence-incremental, so overall we describe the system as sentence-incremental.

In the **Part-Incremental** (Part-Inc) setting, we allow XLNet to access multiple sentences simultaneously, up to 512 tokens. This setting is comparable to experiments in Xia et al. (2020) and Toshniwal et al. (2020b), where document encoding is also non-

[1]We use the *base* version due to memory restrictions.

incremental. In our case, both mention detection and mention clustering components remain incremental as in the Sentence-Incremental setting. In this way, we can isolate the effect of sentence-incrementality on the document encoder (XLNet).

3.4.2.2 Span Representation

We use a similar span representation to Lee et al. (2017): for a span (i, j), we concatenate word embeddings (x_i, x_j), an attention-weighted average $\bar{x}$ and learnable embeddings for span width and speaker ID (the speaker for (i, j)). We use 20-dimensional learned embeddings for all features (span width, speaker ID, document genre, action history, mention distance and number of entities in each cluster).

3.4.2.3 Training

We use Adam to train task-specific parameters, and AdamW for XLNet's parameters (Kingma and Ba, 2015; Loshchilov and Hutter, 2019). The gradient is accumulated across one document before updating model weights. We use a learning rate scheduler with a linear decay, and additionally warmup SpanBERT's parameters for the first 10% update steps. For the mention detector, we balance the loss weights based on the frequency of each action in the training set. This step is important because most tokens do not correspond to mention boundaries, meaning the ADVANCE action is by far the most prevalent in the training set.

Training converges within 15 epochs. The model is implemented in PyTorch (Paszke et al., 2019). A complete list of hyperparameters is included in Appendix B.2.

3.4.3 Comparisons

We compare against several recent works with varying degrees of incrementality. Table 3.1 summarizes their differences in incrementality compared to ours, as well as the span complexity. Joshi et al. (2020) is a non-incremental formulation: it adopts the *end-to-end* formulation from Lee et al. (2017), replacing the LSTM encoder with their novel SpanBERT architecture.

The longdoc (Toshniwal et al., 2020b) and ICoref (Xia et al., 2020) systems adapt Joshi et al. (2020) so that mention clustering is done incrementally. However, both models avoid modifying the non-incremental document encoding and mention detection steps from Joshi et al. (2020), and the resulting systems are only partly incremental.

Since Toshniwal et al. (2020b) and Xia et al. (2020) only experiment with SpanBERT-large, we re-train their implementations with SpanBERT-base to fairly compare against our own systems.

Xia et al. (2020) also provide a truly sentence-incremental version of their system, which we call ICoref-*inc*[2]. This version is trained by encoding tokens and proposing mentions sentence-by-sentence, independently processing each sentence as it is observed while maintaining entity clusters across sentences. Since ICoref-*inc* is fully sentence-incremental, it provides the fairest comparison to our own Sentence-Incremental setting. Using more incremental components increases the coreference task's difficulty, as the system must rely on partial information when making clustering decisions.

We do not compare against Liu et al. (2019a) and Toshniwal et al. (2020a)'s token-incremental models. Besides being generally unsuitable for span-based coreference, they also do not handle nested mentions. Roughly 11% of OntoNotes' mentions are nested, meaning that training these systems on OntoNotes is infeasible.

3.4.3.1 Span Complexity

Table 3.1 also compares the *span complexity* between systems, in terms of how many spans must be scored and compared. This comparison is analytic and not runtime-based, and so ignores handcrafted memory-saving techniques such as eviction and span pruning. Joshi et al. (2020) score all possible spans and compare them pairwise, meaning their system runs in $O(n^4)$, where n is the number of tokens. Toshniwal et al. (2020b) and Xia et al. (2020) reduce the complexity to $O(n^2m)$, where m is the number of entities, by incrementally clustering mentions. Finally, our systems' span complexity is $O(nm)$. Our mention detector proposes $O(n)$ spans, as we can show each action is linearly bounded in the number of tokens. Our reduced complexity speaks to its increased cognitive plausibility compared to the part- and non-incremental systems, which consider a quadratic number of spans.

Note that the runtime is not comparable because non-incremental methods process the entire document in parallel, whereas ours is not parallelizable and therefore slower. We also note that this comparison does not have any bearing on memory requirements, since Toshniwal et al. (2020b) and Xia et al. (2020) both maintain constant memory through either learned or handcrafted eviction strategies.

[2]Specifically, this system is the "Train 1-sentence / Inference 1-sentence" model from Xia et al. (2020)'s Table 4.

Enc. Size	Inc.	Model	*MUC*	B^3	$CEAF_{\phi_4}$	Avg. F1
Large	None	SpanBERT (Joshi et al., 2020)	85.3	78.1	75.3	79.6
		CorefQA+SP (Wu et al., 2020)	88.0	82.2	79.1	83.1
		s2e+Longformer (Kirstain et al., 2021)	85.8	79.1	76.1	80.3
		s2e + se_ct (Chai and Strube, 2022)	86.3	79.6	76.7	80.9
Base	None	SpanBERT (Joshi et al., 2020)	83.7	75.8	72.9	77.4
		CorefQA+SP (Wu et al., 2020)	86.3	77.6	75.8	79.9
Base	Part	longdoc (Toshniwal et al., 2020b)	83.2	74.8	71.4	76.4
		ICoref (Xia et al., 2020)	83.6	75.0	72.5	77.0
		Part-Inc (Ours)	82.9	74.4	71.6	76.3
Base	All	ICoref-*inc* (Xia et al., 2020)	76.7	64.0	63.4	68.0
		Sent-Inc (Ours)	78.8	68.6	62.5	70.0

Table 3.2: Main results on the OntoNotes 5.0 test set. Enc. Size refers to the Encoder Size and Inc. refers to the degree of incrementality. Note that scores for Xia et al. (2020) and Toshniwal et al. (2020b) differ from their reported results because we re-train them with SpanBERT-*base* instead of *large*. The full precision and recall scores can be found in Appendix B.

3.5 Results

3.5.1 OntoNotes

The main results for OntoNotes are shown in Table 3.2. First, SpanBERT (Joshi et al., 2020), being non-incremental, unsurprisingly outperforms other systems, both part and sentence incremental.

Within partly incremental systems, the ICoref model (Xia et al., 2020) performs best, below SpanBERT by 0.4 F1. Our Part-Inc model performs comparably to longdoc (Toshniwal et al., 2020b), only trailing ICoref by 0.7 F1 points.

The advantages of our method are more evident in the sentence-incremental evaluation. Since ICoref-*inc* relies on SpanBERT to encode tokens and score mentions, its performance suffers considerably when evaluated in the sentence-incremental setting. In contrast, the Sent-Inc model effectively uses the history of previous processed sentences and outperforms ICoref-*inc* by 2 F1 points. Still, both systems suffer considerably when

Model	LIGHT	AMI	PERS.	SWITCH.	Avg.
SpanBert-base (Joshi et al., 2020)	57.7	33.8	53.7	50.2	48.9
ICoref (Xia et al., 2020)	54.7	33.7	51.5	48.1	47.0
Part-Inc (Ours)	53.5	32.4	50.5	46.9	45.8
ICoref-*inc* (Xia et al., 2020)	45.5	21.9	41.3	36.6	36.3
Sent-Inc (Ours)	50.5	31.6	46.4	44.7	43.3

Table 3.3: Main results for the CODI-CRAC 2021 corpus. All scores denote the CoNLL F1 score (average of *MUC*, B^3 and $CEAF_{\phi_4}$).

compared to their part-incremental counterparts: ICoref drops by 9 F1 points and our model by 6.3 F1. In Section 3.6, we explore the main causes of this drop.

We perform statistical significance testing between all pairs of experimental settings with the base encoder size, using a paired bootstrap test (Efron and Tibshirani, 1994). We use Dror et al. (2018)'s public implementation of the bootstrap as described in (Berg-Kirkpatrick et al., 2012). The bootstrap test fails to find statistical significance between *any* experimental setting, including in Part Incremental vs. Sentence Incremental pairings. However, there are well-known biases in the bootstrap test which may cause it to fail. Notably, the bootstrap may fail if the test dataset is too small, as it assumes the test set distribution exhibits little deviation from the population distribution (Dror et al., 2018). Despite the lack of statistical significance finding, we are confident that the scores support our conclusions, particularly the large drops in performance between Part Incremental and Sentence Incremental settings. We also try the Permutation test (Noreen, 1989), which in theory is better suited for small datasets, but it again fails to find statistical significance between any experimental setting.

3.5.2 CODI-CRAC

The results on the CODI-CRAC corpus are shown in Table 3.3. We note that scores are generally lower than in OntoNotes, reflecting the out-of-domain experimental setting, as models are trained on OntoNotes and evaluated on CODI-CRAC. We observe many of the same trends as in OntoNotes: the non-incremental SpanBERT again surpasses other models, achieving 2.9 F1 higher than ICoref.

Within partly incremental systems, our Part-Inc system trails ICoref by 1.2 F1. We omit the longdoc results from this table, after finding its performance surprisingly

plummets when evaluated on CODI-CRAC. On all subsets, it scores below 2 F1, indicating issues with model transfer. Other works have explored this topic in depth (Toshniwal et al., 2021), and we do not investigate further here.

In the Sentence Incremental setting, although our Sent-Inc model again outperforms Xia et al. (2020)'s ICoref-*inc*, the performance difference is much larger here: 7 F1 compared to 2 F1 in OntoNotes. The gap between the Sent-Inc and Part-Inc is also much smaller: only 2.5 F1 points compared to 6.3 F1 on OntoNotes. The difference in performances between the two datasets may suggest our model is better suited to the inherent incrementality in a dialogue setting.

As on OntoNotes, we test statistical significance between all experimental settings using the bootstrap test. The test again finds to find statistical significance between any setting. As before, we note that the small test data size may cause the test to fail, and we remain confident in our conclusions.

3.6 Analysis

The dramatic performance gap between the Sent-Inc and Part-Inc settings may be surprising. Since coreference resolution is processed incrementally by humans, why does access to future tokens affect the Sent-Inc model so heavily?

To investigate this issue deeper, we design additional k-Sentence-Incremental settings. In each setting, the system accesses k sentences $(S_1, \ldots, S_k)$ at a time as active input, and $512 - \sum_{i=1}^{k} |S_k|$ tokens through its cache. In each setting, the model observes the same number of tokens (512), but varies the amount of active input vs. cache. The mention detection and mention clustering steps remain the same and are still incremental; the only change is in the encoder.

Varying k in this way allows us to the test the effect of more or less incrementality on the system. When $k = 1$, we recover the original Sent-Inc model. When k is large enough (in practice, 24), we get the Part-Inc model. For each $k \in \{1, 4, 8, 12, 16, 20, 24\}$, we fully train the corresponding model on OntoNotes as described in Section 3.4, and evaluate on the validation set.

The results are shown in Figure 3.5. There are a few notable characteristics. The first is that as k increases, we see a much more dramatic lift when k is small (e.g. moving from 1 to 4 sentences) compared to when k is large. This effect corresponds to the intuition that coreferring expressions are usually close to their antecedent. The more coreference chains the model can observe simultaneously (i.e. that avoid the cache), the

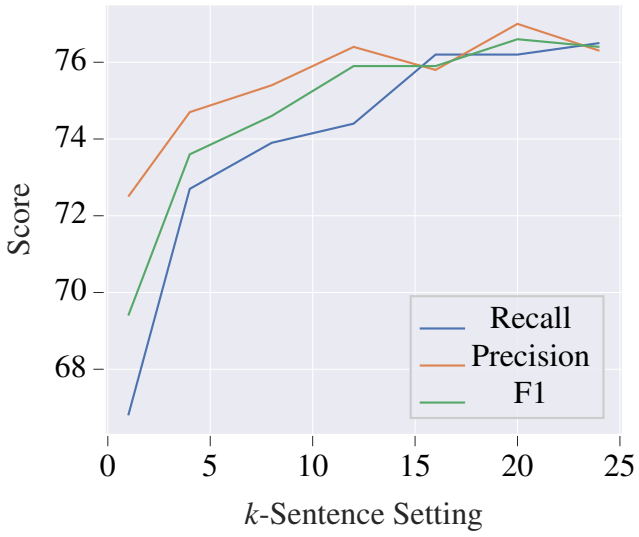


Figure 3.5: The CoNLL F1 performance (average of *MUC*, B^3 and $CEAF_{\phi_4}$) of each k-Sentence-Incremental model on the OntoNotes validation set.

better it is at resolving them.

The second noteworthy trend is that increasing k improves recall (9.7%) far more than precision (3.8%). Although not shown here, we observe this trend across all three metrics within the CoNLL score (*MUC*, B^3 and $CEAF_{\phi_4}$). The result means that finding and resolving true coreference links (i.e. reducing false negatives) is a far more serious obstacle for the Sent-Inc model than for Part-Inc. Since the only difference in these models is how many embeddings are cached, the result suggests caching or not caching embeddings plays a large role in finding and correctly resolving mentions.

3.6.1 k-Sentence-Incremental Mention Detection

We repeat the experiment in Section 3.6 for mention detection. For each k-Sentence-Incremental setting, we evaluate on the validation set and record the mention detection recall, precision and F1.

The results are shown in Figure 3.6. Certain trends remain the same as for the CoNLL score, namely that performance rises more when k is small compared to when it is large. However, we do not see the same dramatic difference in recall between $k = 1$ to $k = 24$ settings as in Section 3.6. Here, the difference in recall between the two settings is around 3%, whereas in Figure 3.5 it is 9.7%.

Overall, the reduced severity between $k = 1$ and $k = 24$ settings compared to Figure 3.5 most likely indicate that XLNet's caching deficiencies affect mention clustering

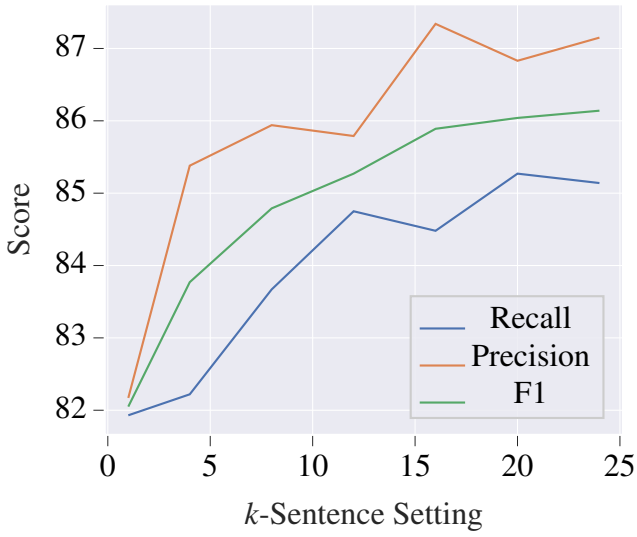


Figure 3.6: Mention Detection F1 performance of each k-Sentence-Incremental model on the OntoNotes validation set.

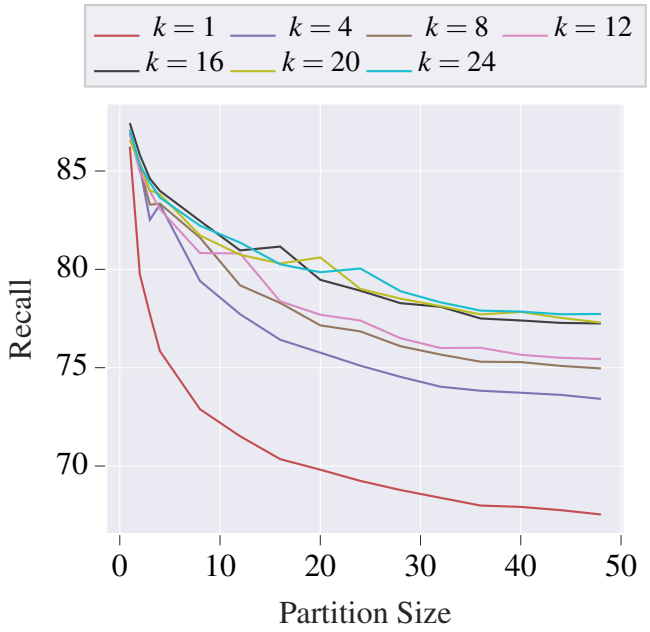
(particularly false negatives) more seriously than mention detection.

3.6.2 Partitioning Document Clusters

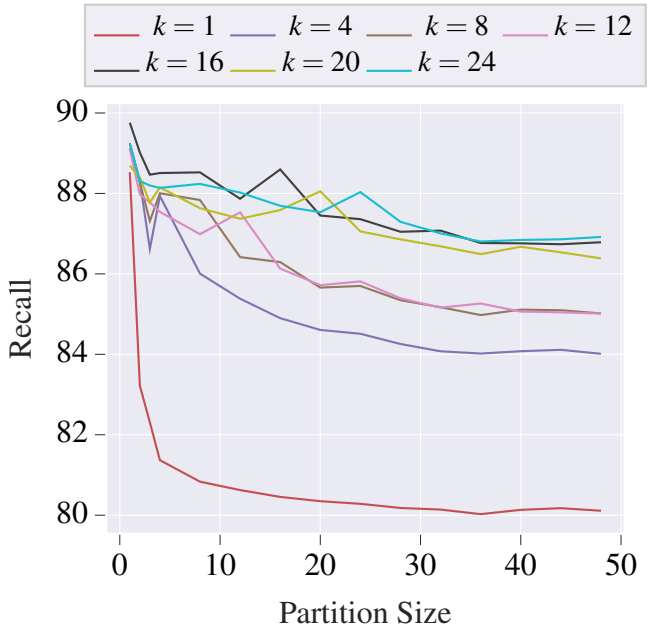
We explore the deficiency in the previous section further, guided by the hypothesis that XLNet relies on active inputs and cannot effectively use its caching mechanism. We partition each document into segments of k sentences, and call the number of sentences in each segment the 'partition size'. Within each segment, we maintain the original coreference links. However, we remove the coreference links between segments. For each k-Sentence-Incremental model, we similarly partition their coreference predictions, and evaluate against the partitioned gold labels on the OntoNotes validation set.

Each segment therefore is independent from the other, and we can measure how reliant the model is on its active inputs by observing performance change across partition sizes. For example, when the partition size is 1, the only coreference links are intra-sentential ones. In this case, models are only evaluated on their intra-sentential coreference resolution ability. When the partition size is large, the models are evaluated on the documents' original coreference chains. Since the previous experiments demonstrated shifts in incrementality heavily affect recall, we measure the mention detection recall and CoNLL recall score.

The results are shown in Figure 3.7. All models do well when the partition size is one, reflecting the fact that intra-sentential coreference is generally simpler than distantly



(a)



(b)

Figure 3.7: Evaluation results on the OntoNotes validation set when the gold labels and k-Sentence-Incremental predictions are partitioned according to various sizes. Figure 3.7a shows the CoNLL recall scores for coreference resolution. Figure 3.7b shows the mention detection recall scores. Notice that whenever k is equal to the partition size, there is a noticeable performance increase, indicating that XLNet relies heavily on active inputs rather than its cache.

	MUC			B^3			$CEAF_{\phi_4}$			Avg.
Model	Rec.	Prec.	F1	Rec.	Prec.	F1	Rec.	Prec.	F1	F1
Sent-Inc	76.4	78.7	77.6	68.0	68.2	68.1	56.1	70.6	62.6	69.4
−speaker	76.4	79.0	77.7	68.0	68.6	68.3	56.3	70.8	62.7	69.5

Table 3.4: Results on the OntoNotes validation set comparing the Sent-Inc model with and without speaker embeddings.

Model	LIGHT	AMI	PERS.	SWBD.	Avg.
Sent-Inc	50.6	32.5	48.5	44.4	44.0
−speaker	50.1	32.0	49.6	43.3	43.8

Table 3.5: Results on the CODI-CRAC dev set comparing the Sent-Inc model with and without speaker embeddings.

linked mentions. As the partition size increases, model performance decreases as the coreference chains become more spread apart and raise the task difficulty. Crucially, we notice a upward performance bump whenever the partition size matches the k-Sentence-Incremental setting, for both coreference performance and mention detection. When k matches the partition size, the model observes coreference chains that are always within the active input window. This performance bump therefore indicates XLNet is much better at mention detection and coreference when the coreference chain occurs within its active inputs. Performance suffers whenever the model must rely more on its memory (whenever k is not equal to the partition size). In particular, these results suggest that more powerful pre-trained language models, for example ones that can take better advantage of cached representations, may be more successful at incremental coreference resolution.

3.6.3 Speaker Embeddings

The ICoref-*inc* model from Xia et al. (2020) is an important comparison point as the only baseline in the sentence-incremental setting. While ICoref-*inc* does not rely on speaker embeddings, our own models (both Part-Inc and Sent-Inc) do. Given the important role of speaker identity in a dialogue setting, it is useful to know the effect of removing these embeddings in our models.

We compare the Sent-Inc model with and without speaker embeddings in Table 3.4

	MUC			B^3			$CEAF_{\phi_4}$			Avg.
Model	Rec.	Prec.	F1	Rec.	Prec.	F1	Rec.	Prec.	F1	F1
SpanBERT+*e2e*	82.5	84.3	83.4	75.8	76.8	76.3	71.7	74.7	73.1	77.6
XLNet+*e2e*	70.3	80.8	75.2	63.7	73.0	68.0	67.5	70.4	68.9	70.7

Table 3.6: Results from non-incremental methods on the OntoNotes validation set with different pretrained language models. *e2e* refers to the non-incremental coreference resolution technique from Lee et al. (2018), which is adapted in Joshi et al. (2020).

for OntoNotes, and Table 3.5 for CODI-CRAC. We find that speaker embeddings play little to no role in resolution performance. In OntoNotes, removing speaker embeddings improves CoNLL F1 by 0.1, and in CODI-CRAC, it decreases performance by 0.2 F1. In both cases, the results are unlikely to be statistically significant. The finding indicates that Sent-Inc's advantage over ICoref-*inc* is not simply due to feature selection but a true modelling advantage. It also suggests that further performance improvements are possible if speaker identity can be better represented, since Sent-Inc effectively ignores the speaker embeddings. One possibility from Wu et al. (2020) (which we will use in Chapter 5), is to preprocess the text with speaker tags directly included in the input, rather than including it separately. This way, the document encoder directly learns how to handle speakers, instead of relying on a separate embedding in downstream classifiers.

3.6.4 XLNet in Non-Incremental Baselines

Choosing XLNet as the document encoder is motivated by the fact that XLNet can efficiently cache and reuse input, making it suitable for incremental processing. However, XLNet can also be used non-incrementally in the same way as SpanBERT. In particular, we can train Joshi et al. (2020)'s coreference system using XLNet instead of SpanBERT. This experiment allows us to compare how the choice of pre-trained language model affects performance.

Table 3.6 shows results of training Joshi et al. (2020) with an XLNet encoder instead of SpanBERT on the OntoNotes dev set. XLNet significantly underperforms compared to SpanBERT, scoring almost 7 CoNLL F1 points lower. Surprisingly, XLNet is an effective document encoder for our Part-Incremental formulation (achieving 76.3 F1 on the OntoNotes test set), but ineffective when used in Joshi et al. (2020)'s non-

incremental setup. We do not attempt swapping the fine-tuned XLNet into Toshniwal et al. (2020b) or Xia et al. (2020) as it seems unlikely to yield useful results.

3.7 Conclusion

In this chapter, we propose a sentence-incremental coreference resolution model using a shift-reduce formulation. The model delays mention clustering until the full span has been observed, alleviating a key flaw with previous incremental systems. It efficiently processes text, and avoids scoring a quadratic number of spans during mention detection.

In a sentence-incremental setting, our method outperforms strong baselines adapted from state-of-the-art systems. When access to the full document is allowed, the proposed system achieves similar performance to state-of-the-art methods while maintaining a higher level of incrementality. We investigate why this relaxation has such a dramatic effect, finding that the document encoder does not make effective use of its cache.

Our sentence-incremental results suggest an important point: non-incremental methods are not effective tools when they must be used incrementally. Incremental systems that perform as well as non-incremental ones have significant implications for downstream applications where text is received incrementally, such as dialogue systems or conversational question answering (e.g. Andreas et al. 2020; Martin et al. 2020). Our proposal demonstrates an important step towards highly effective, incremental coreference resolution systems. In the following chapter, we experiment with more powerful seq2seq approaches using larger pre-trained language models.

Chapter 4

Efficient Seq2seq Coreference Resolution Using Entity Representations

This chapter explores a different paradigm for learning coreference, using seq2seq models. The seq2seq approach directly maps text to text, with the output string encapsulating coreference predictions for the input document. It represents a fundamentally distinct approach from the encoder-based approach we explored in Chapter 3, which focuses on creating span representations corresponding to mentions. Seq2seq approaches avoid creating span representations entirely, and leverage large pre-trained models without fine-tuning task-specific parameters. These design choices lead to state-of-the-art performance on benchmark datasets such as OntoNotes (Weischedel et al., 2013).

Figure 4.1 shows the expected predictions of two recent seq2seq coreference systems, Link-Append (Bohnet et al., 2023) and Token Action (Zhang et al., 2023), on the given input "*Auto workers ended their strike*". Both systems output a sequence of tokens corresponding to coreference predictions. In the Link-Append system, this output is a sequence of transition actions corresponding to coreference predictions, such as linking "*their*" to "*Auto workers*" with a right arrow ($\rightarrow$) token. In Token Action, the output is the input document, augmented with special tokens corresponding to mention boundaries and cluster IDs. In both cases, the seq2seq model is trained to output a token sequence corresponding to the correct coreference prediction. In this chapter, we extend Zhang et al. (2023)'s Token Action model, which we will describe more fully in Section 4.3.1.

In this chapter, we examine how discourse representations can benefit the seq2seq

Model	Prediction
Link-Append	their → Auto workers ; their strike → [2 ; SHIFT
Token Action	<m> Auto workers | 0 </m> ended <m> <m> their | 0 </m> strike | 1 </m>

Figure 4.1: Expected labels for the input "*Auto workers ended their strike*" from the Link-Append (Bohnet et al., 2023) and Token Action (Zhang et al., 2023) systems. In the top cell, Link-Append links "*their*" to its antecedent "*Auto workers*" (creating cluster 1), assigns the span "*their strike*" to cluster 2, then terminates with the special SHIFT token. In the bottom cell, the Token Action system marks mentions with special <m> and </m>, assigning spans "*Auto workers*" and "*their*" to cluster 0 and "*their strike*" to cluster 1. We describe Zhang et al. (2023) more formally in Section 4.3.1.

approach. Similar to Chapter 3, we hypothesize that **building a discourse representation is an efficient and effective strategy for seq2seq approaches to incremental coreference resolution**. In this chapter, we represent the discourse as a list of entities, stored as plain text strings. Our method works by extracting and re-organizing mention-level tokens into entity representations, and discarding the majority of other input tokens. Our method's ability to re-organize and compress inputs means it efficiently handles incremental coreference resolution, differing from prior methods which must re-input the entire output at each step. However, our proposed model underperforms against non-incremental models, and we point to artifacts in the OntoNotes annotation scheme, namely the lack of singleton annotation, as a major factor in this result. On LitBank, where singleton mentions are annotated, our models pass state-of-the-art performance. Our results indicate that building a discourse model based on entities is a feasible strategy for incremental coreference resolution in seq2seq resolvers.

4.1 Introduction

Seq2seq approaches to coreference resolution have reached state-of-the-art abilities on coreference benchmarks and have put forward an exciting new direction for the task (Bohnet et al., 2023; Zhang et al., 2023). In this chapter, we are interested in an *incremental* setting, where the model receives the text piece-by-piece and must detect mentions and compute coreference relations in the given chunk of text before receiving

the next one. This setting more closely imitates human language processing, which is strongly incremental (Altmann and Steedman, 1988; Christiansen and Chater, 2016). Real-world settings such as dialogue are inherently incremental and would benefit from advances in this domain.

In an incremental setting, existing seq2seq approaches are largely unsatisfying: Bohnet et al. (2023) requires the entire output to be re-input for each new sentence in the discourse, while Zhang et al. (2023) assumes the entire document is static and always available. These limitations motivate us to explore a seq2seq solution that can efficiently perform coreference in an incremental setting.

We hypothesize that re-inputting the entire output at each iteration may be unnecessary for incremental coreference resolution, and that condensing the output text may suffice. Using Zhang et al. (2023)'s seq2seq model as a starting point, we propose an **Model-based Incremental representation**. For each new text chunk, we only re-input text spans corresponding to predicted entities, and discard tokens outside of these spans. Entities are then re-ordered such that newly mentioned entities appear at the rightmost position of the entity list.

Our model represents a fundamentally different approach to prevailing language modelling trends: we aggressively discard tokens from the context window, and instead keep a "memory" representing the current state of the document, corresponding to a discourse model.

On OntoNotes (Weischedel et al., 2013), we find that our method achieves just 0.7 CoNLL F1 below a full-prefix, incremental baseline based on Zhang et al. (2023) while reaching almost twofold compression in the input length. We find similar results on CODI-CRAC, where our proposed model-based approach slightly underperforms the non-incremental baseline. On LitBank, our model's performance surpasses contemporary methods, while slightly lagging behind the Full-prefix baseline by 0.8 F1.

Finally, we analyze model errors on the validation set. We find that the incremental setting affects the model's ability to accurately predict coreference for named entities and definite noun phrases on OntoNotes. We hypothesize that incremental processing is particularly difficult on OntoNotes as the dataset does not annotate singletons, resulting in an unfavourable source of noise. This finding is reinforced by our results on LitBank, which includes singleton annotation and where we do not see this effect. In the Model-based Incremental model, we find the loss of context in the Model-based representation hurts recall performance in cases involving deixis (for example, the authorial "*We*" referring to "*CNN*" in a news piece).

Incremental:
<m><m>Wetland Park | 0 </m>workers | 1 </m>are now in the middle of intensive work.<m>They | 1 </m>will complete <m>the park's | 0 </m>entire construction by the beginning of <m>2006 | 2 </m>, to be able to participate in <m>the <m>2006 | 2 </m>Discover Hong Kong Year campaign | 3 </m>. We have <m>established | 4 </m><m>the year 2006 | 2 </m>as Discover Hong Kong Year. Why is <m>that | 4 </m>? Because, as everyone knows, <m>our Disneyland | 5 </m>will open in September of <m>this year | 2 </m>. In addition, we will have Ngong Ping 360, that is, the cable car, er, to the Giant Buddha. <target>They add to what we already have, like the Avenue of Stars, which is also very famous. Moreover, er, we are including our software. Hong Kong's software is very well known. Like what's used in our Symphony of Lights. We hope to use, er, a variety of hardware and software to package this entire 2006 Discover Hong Kong Year. Without planning it in advance, they chose to settle here. A dream that has been anticipated for more than twenty years will soon come true here.</target>

Model-based Incremental Representation:
<e><m>Wetland Park workers </m><m>They </m>| 1</e>
<e><m>Wetland Park</m><m>the park's </m>| 0 </e>
<e><m>the 2006 Discover Hong Kong Year campaign </m>| 3 </e>
<e><m>established </m><m>that </m>| 4 </e>
<e><m>our Disneyland </m>| 5 </e>
<e><m>2006<m><m>2006</m><m>the year 2006</m><m>this year</m>| 2 </e>
<context>Because, as everyone knows,<m>our Disneyland | 5 </m>will open in September of <m>this year | 2 </m>. In addition, we will have Ngong Ping 360, that is, the cable car, er, to the Giant Buddha. </context> <target>They add to what we already have, like the Avenue of Stars, which is also very famous. Moreover, er, we are including our software. Hong Kong's software is very well known. Like what's used in our Symphony of Lights. We hope to use, er, a variety of hardware and software to package this entire 2006 Discover Hong Kong Year. Without planning it in advance, they chose to settle here. A dream that has been anticipated for more than twenty years will soon come true here.</target>

Label:
They add to what we already have, like the Avenue of Stars, which is also very famous. Moreover, er, we are including <m>our software | 6 </m>.<m><m>Hong Kong's | 7 </m>software | 6 </m>is very well known. Like what's used in our Symphony of Lights. We hope to use, er, a variety of hardware and software to package <m>this entire 2006 Discover Hong Kong Year | 3 </m>. Without planning <m>it | 8 </m>in advance, they <m>chose | 8 </m>to settle here. A dream that has been anticipated for more than twenty years will soon come true here.

Figure 4.2: **Top**: An example of the Full-prefix incremental setting. The text in blue represents previously labelled chunks. The model will predict coreference clusters in the next chunk, enclosed by `<target>` and `</target>` tokens. **Middle**: An example from the Model-based Incremental representation. Mentions from the same cluster are grouped together and labelled with their cluster identity. The fixed-length context, shown in blue, consists of previously labelled sentences as in the incremental setting. **Bottom**: The expected output for this sample.

4.2 Related Work

Our proposed method directly builds on recent seq2seq methods for coreference resolution (Bohnet et al., 2023; Zhang et al., 2023). In these methods, an encoder-decoder model learns to directly map input text to a labelled text representing coreference predictions. The two methods differ both in their annotation schema as well as the general procedure for processing the document.

Zhang et al. (2023)'s method explores several annotation schemas for coreference resolution. Each schema considers a set of added special tokens, corresponding to decisions such as marking mention boundaries and making a clustering prediction. They fine-tune models from the T5 family (Raffel et al., 2020) to encode the input document and generate an annotated document corresponding to cluster predictions. Notably, their method assumes the entire document is always available and they do not explore incremental settings. We will describe their method in more detail in Section 4.3.1.

Similarly, Bohnet et al. (2023) fine-tune the mT5 encoder-decoder model (Xue et al., 2021) to output text corresponding to coreference predictions. Unlike Zhang et al., Bohnet et al. processes text sentence-by-sentence in a transition-based approach. The set of actions include steps such as appending to an existing entity cluster, linking directly to a previous text span, or creating a new entity. After each action, the system re-inputs the entire output again as context, meaning computation costs rise considerably for long documents.

In Chapter 3, we explored entity representations which maintain a set of hidden representations corresponding to each entity cluster (Yu et al., 2020; Toshniwal et al., 2020b; Xia et al., 2020; Xu and Choi, 2022; Grenander et al., 2022), with similar approaches in cross-document coreference resolution (Allaway et al., 2021; Logan IV et al., 2021). In contrast to seq2seq approaches, these methods are generally less accurate and more complex. They require additional task-specific parameters alongside the base encoder, as well as a separate mention detection step (Yu et al., 2020; Toshniwal et al., 2020b; Xia et al., 2020; Xu and Choi, 2022; Grenander et al., 2022), or gold mentions as input (Allaway et al., 2021; Logan IV et al., 2021).

Our method of compressing and simplifying processed inputs is motivated by cognitive theories of human language processing such as File-Change Semantics (Heim, 1983), as described in Section 2.1. Heim argues that discourse processing can be regarded as "file-keeping", in which file cards keep track of entities and their properties in each discourse segment. New file cards are either added or existing ones updated

as new entities are encountered or new information about them becomes available. Our approach can be seen as implementing a simplified version of this theory, without predicates.

More recently, the Now-or-Never bottleneck (Christiansen and Chater, 2016) proposes that humans rapidly compresses and re-codes language as it is encountered. They argue that the speed humans understand language suggests it is compressed and re-organized, or otherwise forgotten. Although their mechanism differs from ours, we apply this compression principle in our approach.

Many methods for efficient coreference resolution exist; however, most focus on improving the dominant "End-to-End" approach (Lee et al., 2017). For example, Dobrovolskii (2021), Martinelli et al. (2024) and Kirstain et al. (2021) improve the efficiency of span representations to ease span comparisons. These methods are not applicable to seq2seq approaches such as Zhang et al. and Bohnet et al.. Ahmed et al. (2023) focus on efficiency in event coreference resolution but their method largely does not apply to entity resolution.

Outside of coreference, Nawrot et al. (2024) look at retrofitting LLMs to dynamically pool and compress tokens as they are predicted. Although their motivation is very similar to ours, their technique focuses on decoder-only architectures in language modelling, and does not result in an interpretable discourse model as in our approach.

4.3 Method

We first describe Zhang et al. (2023)'s seq2seq approach (4.3.1), which serves as the basis for our proposal. We then describe how we incrementalize their model to provide a comparable baseline in Section 4.3.2. Finally, in 4.3.3 we explain how we simplify and compress the input using the Model-based Incremental representation.

4.3.1 Seq2seq Coreference Resolution

Zhang et al. (2023) frames coreference as a text-to-text task, which takes the document as input and outputs the same document annotated with mention boundaries and coreference relations. Here, we describe their **Token Action** formulation, referring the reader to Zhang et al. (2023) for other settings.

Token Action augments the vocabulary with special tokens marking mention boundaries and cluster predictions. Given a mention $(x_i \ \ldots \ x_j)$ referring to entity cluster l_1,

the model outputs:

$$\texttt{<m>}\ x_i\ \ldots\ x_j\ |\ l_1\ \texttt{</m>}$$

where <m> and </m> are special tokens marking the mention start and end, and | is a special token preceding cluster predictions. The cluster prediction is always an integer greater or equal to 0.

At inference time, the model's output probabilities are modified to either output the next token from the document or special tokens. The outputs are constrained to always form valid coreference predictions, e.g. | must precede l_1.

4.3.2 Full-Prefix Incremental Baseline

In Zhang et al. (2023)'s seq2seq formulation, the entire document is encoded in a single pass, with the assumption that the entire document is statically available. As discussed previously, this assumption may not hold for various reasons, such as in dialogue applications. Here, we adapt a full-prefix, incremental baseline where the model receives a text chunk and must compute potential mentions and coreference relations before receiving the next text chunk.

We assume the model has been divided into text chunks $[c_1, \ldots, c_N]$, and we denote c_n's labelled sequence as $\hat{c}_n$. Given text chunk c_n, we concatenate all previously labelled text chunks, then append c_n as the target chunk. We introduce special <target> and </target> tokens, which indicate the beginning and end of the target chunk. After processing text chunk c_n and generating predictions $\hat{c}_n$, the input for the next chunk c_{n+1} is

$$\hat{c}_1 \ldots\ \hat{c}_n\ \texttt{<target>}\ c_{n+1}\texttt{</target>}$$

At test time, we replace each labelled chunk $\hat{c}_i$ with the model's own predictions. An example of the incremental setting is shown in Figure 4.2.

Note that Bohnet et al. (2023) predict coreference relations incrementally in a similar setup. The main difference lies in their transition actions, which allow the model to predict cluster identities, directly link mentions, or create a singleton mention.

4.3.3 Model-based Incremental Representation

After processing each text chunk, the Model-based Incremental representation compresses the model outputs, keeping tokens from mention spans and discarding all others. Each entity is represented by its mentions' tokens, including mention bound tokens <m>

and </m>. It is then demarcated with special <e> and </e> tokens, e.g., an entity with mentions $[m_1, \ldots, m_k]$ with cluster ID l_1 will be:

$$\texttt{<e><m>}\ m_1\ \texttt{</m>}\ \ldots\ \texttt{<m>}\ m_k\texttt{</m>|}\ l_1\ \texttt{</e>}$$

New mentions are either appended to their referent cluster or initialized as a new cluster.

We would also like the representation to reflect the notion that recently mentioned entities tend to be more salient in the discourse (Grosz et al., 1995). After an entity is mentioned, we promote it to the rightmost position of the representation. This reordering signals to the model which entities are likely to be relevant to the current discourse.

We hypothesize that in some cases, solely tracking entities may not provide sufficient context to resolve new mentions. We experiment with adding a limited context window of previously labelled sentences immediately preceding the target chunk, marked with special <context> and </context> tokens. Sentences beyond the window are dropped.

To summarize, suppose after chunks $c_1, \ldots, c_n$, the model has observed entity clusters $[1 \ldots K]$. Then the input representation for target c_{n+1} is:

$$\texttt{<e>}\ldots\ \texttt{|}\ l_1\ \texttt{</e>}\ \ldots\ \texttt{<e>}\ldots\ \texttt{|}\ l_K\ \texttt{</e>}$$
$$\texttt{<context>}\ \tilde{c}_n\ \texttt{</context>}$$
$$\texttt{<target>}\ c_{n+1}\ \texttt{</target>}$$

where $\tilde{c}_n$ consists of the annotated sentences immediately preceding c_{n+1} up to some fixed window length. An example is shown in Figure 4.2.

4.4 Experiments

4.4.1 Datasets

As in Chapter 3, we train and evaluate our proposed method using the OntoNotes dataset (Pradhan et al., 2012). We also train and evaluate on the LitBank dataset (Bamman et al., 2020), as described in Section 2.2.2.3. The dataset contains 100 documents covering literary text extracts, each containing over 2000 tokens on average. Unlike OntoNotes, LitBank includes singleton annotation. We perform the suggested 10-fold cross validation with an 80-10-10 train-validation-test split, due to the dataset's small size. Lastly, we include results on the CODI-CRAC 2021 corpus, as in Chapter 3.

4.4.2 Metrics

As before, we follow the standard evaluation in the CoNLL-2012 Shared Task (Pradhan et al., 2012) and report *MUC* (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998) and $CEAF_{\phi_4}$ (Luo, 2005) metrics and their average (the CoNLL score).

4.4.3 Comparisons

For all experiments, we use the T0 model (Sanh et al., 2022b), a 3B parameter encoder-decoder model based on T5 (Raffel et al., 2020). All models are implemented using the Hugging Face library (Wolf et al., 2019).

4.4.3.1 Non-Incremental

As a first baseline and for reproducibility, we re-train Zhang et al. (2023)'s non-incremental Token Action model using T0. This experiment allows us to more directly compare non-incremental and incremental settings.

4.4.3.2 Full-Prefix Incremental Baseline

We incrementalize the Token Action model using the method described in Section 4.3.2. We set the chunk size to 100 tokens, rounded up to the nearest sentence.

4.4.3.3 Model-based Incremental

We compress the inputs using the Model-based Incremental representation described in Section 4.3.3. As in the incremental baseline, we set the chunk size to 100. We experiment with varying the context length across $\{0, 50, 100, 200\}$. Using 0 context tokens examines an extreme case where only the mention spans and cluster identities are available to the model.

4.4.3.4 Other Baselines

We include Zhang et al. (2023)'s highest performing Copy method which uses the 11B parameter $T0_{pp}$ model (Sanh et al., 2022b), as well as Bohnet et al. (2023)'s Link-Append model, which uses the 13B parameter mT5 model (Xue et al., 2021). We also compare against several contemporary non-seq2seq baselines, which typically extend Lee et al. (2017)'s "End-to-End" approach. Lastly, we include an LLM-based approach (Le and Ritter, 2024).

Model	*MUC*	B^3	$CEAF_{\phi_4}$	Avg. F1
GPT-4 (Le and Ritter, 2024)	73.7	62.7	52.3	62.9
SpanBERT (Joshi et al., 2020)	85.3	78.1	75.3	79.6
CorefQA (Wu et al., 2020)	88.0	82.2	79.1	83.1
s2e+Longformer (Kirstain et al., 2021)	85.8	79.1	76.1	80.3
wl-coref+RoBERTa (Dobrovolskii, 2021)	86.3	79.9	76.6	81.0
ASP + Flan-$T5_{XXL}$ (Liu et al., 2022)	87.2	81.7	78.6	82.5
Link-Append (Bohnet et al., 2023)	87.8	82.6	79.5	83.3
Copy + $T0_{pp}$ (Zhang et al., 2023),	87.6	82.4	79.5	83.2
Token Action + $T0_{3B}$ (Zhang et al., 2023)	87.2	81.5	78.5	82.4
Token Action + $T0_{3B}$, Non-Incremental	87.0	81.0	78.2	82.0
Full-Prefix Incremental	85.5	79.0	74.3	79.6
Model-Based Incremental, Context 0	84.6	77.3	72.3	78.1
Model-Based Incremental, Context 50	84.8	78.0	73.3	78.7
Model-Based Incremental, Context 100	85.1	78.2	73.4	78.9
Model-Based Incremental, Context 200	85.0	78.2	73.5	78.9

Table 4.1: Results on the OntoNotes test set. The bottom section shows our baselines and proposed methods. The full precision and recall scores can be found in Appendix C.1.

4.4.3.5 Training

For all experiments, we use the same hyperparameters as in Zhang et al. (2023)'s Token Action model. We train each model for 30 epochs using the Hugging Face Transformers library (Wolf et al., 2019) on 4 NVIDIA RTX A6000 48 GB GPUs. Each training run takes ~24 hours. Note that Zhang et al. (2023) train for 100 epochs, and we do not expect our reproduced baseline to fully recover their scores.

4.5 Results

4.5.1 OntoNotes

Table 4.1 shows our main results on the OntoNotes test set. First, we note that our reproduction of Zhang et al.'s Token Action scores 0.4 CoNLL F1 below them. As

mentioned, the difference may be attributed to training epochs, as Zhang et al. train their models for 100 epochs. while we train for 30 epochs due to computational limitations.

The **Full-Prefix baseline** scores 2.4 F1 points lower than the non-incremental counterpart. The model sees very little degradation in precision scores, improving on the baseline by 0.6 and 0.7 on *MUC* and B^3, and dropping 0.2 on $CEAF_{\phi_4}$. The decrease is mostly in recall performance, where the gap is 3.6, 4.7 and 7.2 for *MUC*, B^3 and $CEAF_{\phi_4}$ respectively. $CEAF_{\phi_4}$ is an entity-focused metric, and the large recall gap suggests that the Full-Prefix baseline tends to miss entire entities relative to the baseline, rather than individual mentions.

The drop in recall scores is unsurprising due to the challenging incremental setting. This baseline receives chunks of document at a time, and once a decision on anaphoricity has been made, it cannot be revisited after receiving the next chunk. In contrast, Zhang et al. (2023)'s non-incremental model observes the entire document simultaneously, and can decide more easily if distant mentions co-refer.

The **Model-based Incremental representation** slightly underperforms compared to the Full-Prefix baseline, but overall achieves very close scores. We find 100 context tokens performs best, achieving only 0.65 CoNLL F1 below the Full-Prefix baseline. Compared to the Full-Prefix baseline, improvements in precision are slightly offset by declines in recall performance across all three metrics. The biggest drop is from $CEAF_{\phi_4}$ recall, where the Model-based Incremental model scores 2 points lower.

Using 200 context tokens provides no further gains compared to using 100 context tokens. This result suggest that some context is important for the Model-based Incremental representation, but its usefulness saturates after a point. On the other hand, using no context at all achieves the lowest score. Impressively, it only suffers a 1.5 CoNLL drop relative to the Full-Prefix baseline, despite discarding all tokens outside of mention spans.

As in Chapter 3, we perform statistical significance testing using Dror et al. (2018)'s bootstrap implementation between all pairs of settings (Non-Incremental, Full-Prefix Incremental and Model-Based Incremental settings). As in Chapter 3, the test fails to find statistical significance, likely due to small test data size. We again believe that the bootstrap result does not impact our conclusions, but rather reflects the difficulty of evaluating on small test datasets.

We note that Bohnet et al.'s Link-Append and Zhang et al.'s Copy + $T0_{pp}$ model outperform all settings. Their underlying encoder-decoder models are significantly larger than our own: Bohnet et al. uses the 13B mT5 model (Xue et al., 2021), while

Model	*MUC*	B^3	$CEAF_{\phi_4}$	Avg.
Joshi et al. (2020)	89.5	78.2	67.6	78.4
Toshniwal et al. (2021)	-	-	-	79.3
Zhang et al. (2023)	-	-	-	78.3
Non-Incremental	88.8	77.5	68.3	78.2
Full-Prefix Incremental	90.3	80.3	71.6	80.7
Model-Based Incremental, Context=100	89.9	79.0	70.9	79.9

Table 4.2: Results on the LitBank test set. Joshi et al. (2020) is reported by Thirukovalluru et al. (2021).

Model	LIGHT	AMI	PERS.	SWBD.	Avg.
SpanBert-base (Joshi et al., 2020)	57.7	33.8	53.7	50.2	48.9
ICoref (Xia et al., 2020)	54.7	33.7	51.5	48.1	47.0
Part-Inc	53.5	32.4	50.5	46.9	45.8
Non-Incremental	76.7	36.3	69.3	62.4	60.8
Model-Based Incremental, C=100	72.9	41.8	63.6	60.1	59.4

Table 4.3: Results on the CODI-CRAC 2021 corpus. All scores denote the CoNLL F1 score (average of *MUC*, B^3 and $CEAF_{\phi_4}$).

Zhang et al.'s uses the 11B $T0_{pp}$ model (Sanh et al., 2022b). Our experiments with the T0 model uses 3B parameters with a similar amount of pre-training data as other T5 models. Similarly, we do not report results from Chapter 3 on OntoNotes. The XLNet-base encoder used in Chapter 3 only holds 110M parameters and the scores in Chapter 3 are expectedly lower.

4.5.2 LitBank

Our results on LitBank are shown in Table 4.2. The Full-Prefix baseline and Model-based Incremental model achieve the highest scores among available systems, with the Model-based Incremental model scores 1.6 CoNLL F1 higher than Zhang et al. (2023). It drops slightly (-0.8 F1) in performance relative to the Full-Prefix baseline, as on OntoNotes. Interestingly, the performance drop in OntoNotes between incremental and non-incremental systems is not repeated in LitBank, which we explore further in

Section 4.6.4.

4.5.3 CODI-CRAC

Our results on the CODI-CRAC 2021 corpus are shown in Table 4.3. We are unable to report the Full-Prefix Incremental baseline scores as its highly memory-intensive nature prevents us from running it with our available GPUs. Although overall, CODI-CRAC averages fewer tokens per document than LitBank, the AMI corpus within CODI-CRAC consists of 6K tokens per document (Khosla et al., 2021), which is too large for our resources.

We observe the non-incremental baseline surpasses the Model-Based approach, scoring 1.4 CoNLL F1 higher on average across the four datasets in CODI-CRAC, echoing the results on OntoNotes. However, we note that differences across the four datasets are unequal: while the Non-Incremental baseline is 5.7 F1 higher on PERSONA, the Model-Based Incremental system surpasses the baseline by 5.5 F1 on AMI. As noted before, AMI's texts are far longer than the other datasets in CODI-CRAC. This feature may advantage the Model-Based approach, as the compression approach allows it to significantly reduce the context size.

We test for statistical significance between the Non-Incremental and Model-Based Incremental settings using the bootstrap test. As before, the test fails to find statistical significance, again possibly reflecting the small dataset size.

We report the methods and compared baselines from Chapter 3 for completeness; however, these methods should not be directly compared to the other methods in this chapter as their encoder-based architectures only contain 110M parameters. The $T0_{3B}$ encoder-decoder model we use throughout this chapter uses 3B parameters, and it is not reasonable to expect matching performance.

4.6 Analysis

4.6.1 Compression Ratio

Since the Model-based Incremental model only keeps a fraction of the total input at each iteration, its input length grows much slower as new chunks are processed. Here, we quantify the reduction in size of the Model-based Incremental model's inputs relative to the Full-Prefix baseline. We define the compression ratio (CR) as the ratio of the Full-Prefix baseline's input length to Model-based Incremental's input length on the

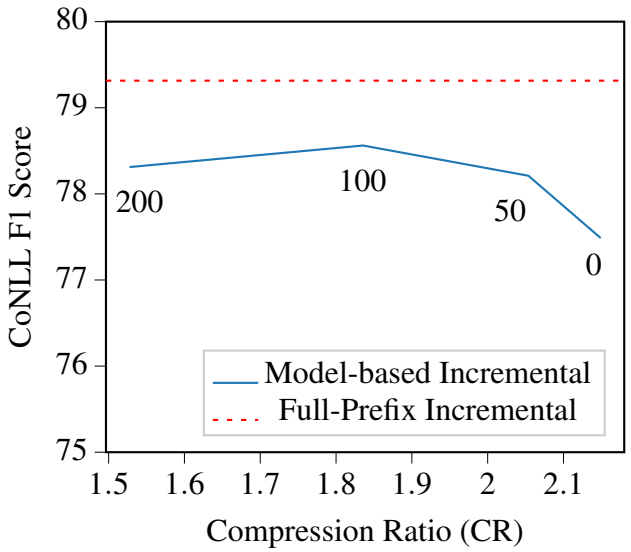


Figure 4.3: Compression Ratio experiments on the OntoNotes validation set.

Model	Peak Mem. (GB)	Peak Mem. w/o Fixed Costs (GB)
Non-Incremental	18.1	7.4
Full-Prefix Incremental	16.2	5.5
Model-based Incremental, C=100	14.5	3.8

Table 4.4: GPU memory usage on the LitBank validation set, fold 0. The 'Peak Mem. w/o Fixed Costs' column subtracts the memory cost of the T0 encoder-decoder model, which is constant across the three settings.

last target chunk of each document. We compute the ratio for each document in the OntoNotes validation set, and report the average in Figure 4.3.

The highest performing Model-based Incremental model, using 100 context tokens, achieves CR=1.8 while scoring 0.75 F1 points below the Full-Prefix baseline. On the right side of the figure, higher compression is achieved at the expense of performance: when no context tokens are used, the Model-based Incremental representation achieves CR=2.1 while retaining 97.7% of Full-Prefix baseline performance.

4.6.2 GPU Memory Usage

More practically, we can also ask whether the Model-based Incremental model reduces GPU memory usage compared to other baselines. Since the Model-based Incremental

	MUC			B^3			$CEAF_{\phi_4}$			Avg.
Setting	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	F1
Model-Based Rep.	86.5	82.2	84.3	80.6	75.3	77.9	79.2	68.5	78.6	78.6
Document Order	87.1	81.6	84.3	81.3	74.4	77.7	79.5	67.8	78.4	78.4

Table 4.5: Results on entity ordering on the OntoNotes validation set. Model-Based Rep. refers to the regular Model-based Incremental representation, while Document Order drops the entity re-ordering step.

model requires keeping less tokens in context, we expect GPU memory usage should decrease. We use the `torch.cuda.memory._record_memory_history` function in PyTorch to record GPU memory usage while running inference, then run the Non-Incremental, Full-Prefix Incremental and Model-based Incremental (C=100) models on the LitBank, fold 0 validation set. We inspect the memory usage[1] to determine maximum memory usage as well as the memory footprint of the models themselves (i.e. without processing documents).

Table 4.4 shows the GPU usage of the three settings. On the right column, we only consider the cost of document processing, i.e. without the fixed cost of the model itself, which is constant (10.7 GB) across the three settings. Comparing the non-incremental and Model-based Incremental models, maximum GPU usage decreases from 7.4 to 3.8 GB, representing a 1.9x reduction, similar to our compression ratio analysis. The result shows that the Model-based Incremental system is a practical method for reducing GPU memory usage.

4.6.3 Entity Ordering

The Model-based Incremental model dynamically re-orders entities each time a mention is detected. We analyze the contribution of this step by ablating the re-ordering step. We train the model as usual with 100 context tokens, but instead of re-ordering, we keep entities in the same order as they appear in the document. The result is shown in Table 4.5.

We find that entity re-ordering has a small effect, lifting the CoNLL F1 score by 0.2. We also evaluate the 'Document Order' ablation on the test set and find a larger drop of 0.45 CoNLL F1 relative to the Model-based Incremental model, indicating it plays a

[1] https://docs.pytorch.org/memory_viz

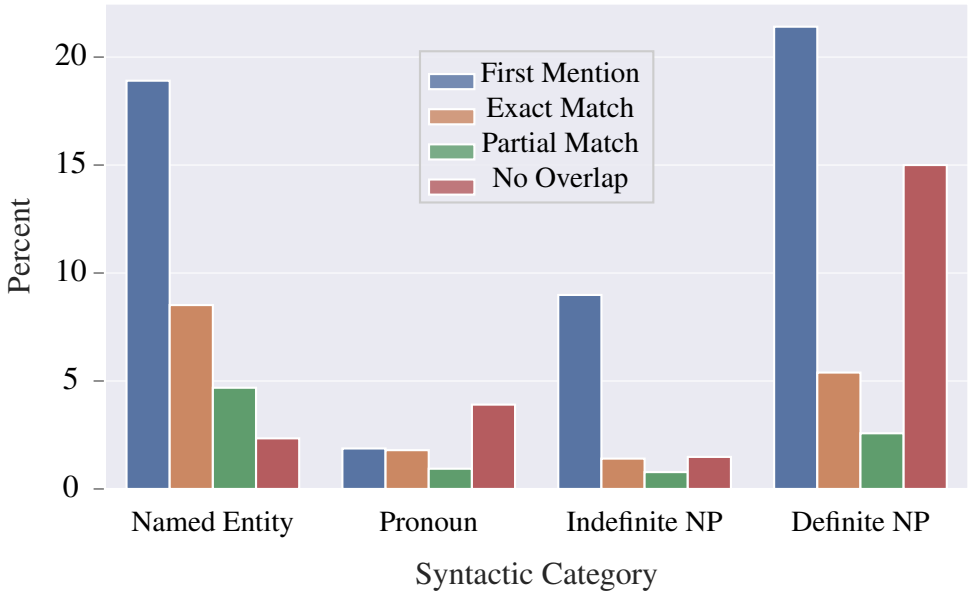


Figure 4.4: Mentions correctly predicted by the non-incremental model but missed by the Full-Prefix Incremental model, separated by syntactic category and string overlap with antecedent.

small effect in coreference performance.

4.6.4 Sources of Error in Incremental vs. Non-Incremental Settings

In our OntoNotes results, we find a 2.5 F1 point gap between incremental and non-incremental settings. While not directly comparable, Bohnet et al. (2023)'s system does not suffer from any performance gap compared to Zhang et al. (2023)'s non-incremental model, despite also implementing an incremental setting. This observation leads us to examine the main sources of error in our Full-Prefix baseline.

We collect all mentions in the OntoNotes validation set that are correctly detected by the non-incremental baseline but missed by the Full-Prefix model. Inspired by Otmazgin et al. (2023), we divide mentions into four broad syntactic categories: (1) named entities, (2) pronouns, (3) indefinite NPs and (4) definite NPs.[2] For each mention, we further record whether the mention's direct antecedent is an exact string match, a partial string match[3] or otherwise has no string overlap. For example, the mention *he* with antecedent *John* will be marked as a pronoun with no overlap to its antecedent, while mention *John* with antecedent *John Doe* will be marked as a named entity with partial overlap to

[2]We use the OntoNotes gold POS tags and set of simple heuristic rules to assign each mention to its syntactic category.

[3]We record a partial string match if either the mention or its antecedent are substrings of each other.

its antecedent. Since the first mention of an entity has no antecedent, we record these mentions as a separate category.

The results are shown in Figure 4.4. Named entities and definites dominate the missed mention set, accounting for 76% of all mentions. Although certain categories are not easy to remedy, such as definite noun phrases with no overlap to their antecedent, we expect other categories, such as named entities with exact match, to be relatively straightforward. Assuming their antecedent is also missed, named entities and definite noun phrases with exact or partial matches to their antecedent account for 40% of missed mentions.

This error type reflects a fundamental difficulty of incremental settings: the model must decide early on in a text whether a given span of text is likely to be involved in coreference and cannot revisit the decision in later chunks. The OntoNotes annotation schema compounds this difficulty: a mention is only marked if it co-refers with another mention in the text, rather than if it simply *could* be the target of coreference. These "singleton" mentions are occasionally annotated in other coreference datasets (Uryupina et al., 2016; Yu et al., 2022a; Bamman et al., 2020), and we note that we do not experience this drop in performance on LitBank, which includes singletons. This artifact means the model does not learn whether a given span of text *could be the target of coreference* but rather whether it is *likely to have been annotated as such in the current document*. We suspect this error type is a major cause of lower recall scores, in particular the entity-focused $CEAF_{\phi_4}$ metric.

Interestingly, Bohnet et al. (2023) does not seem to suffer similarly, despite also being an incremental model. We suspect this difference is due to their distinct set of transition actions. Bohnet et al. (2023) allows for a 'Link' action, which marks two text spans as co-referring without first creating a mention. However, the technique cannot be extended to our Model-based Incremental representation, as it requires keeping the entire document in context at all times.

Lastly, we carry out oracle experiments on adding back gold coreference links to the Full-Prefix model across several steps: (1) named entities that are exact matches of their antecedent, (2) named entities that are partial matches of their antecedent, (3) definite noun phrases that are exact matches of their antecedent, and (4) definite noun phrases that are partial matches of their antecedent.

The results are shown in Figure 4.5. At each step, the CoNLL F1 increases, with the largest increase (1.2 F1) coming from adding exact match named entities. After adding back gold mentions across all four steps, the resulting CoNLL score equals the

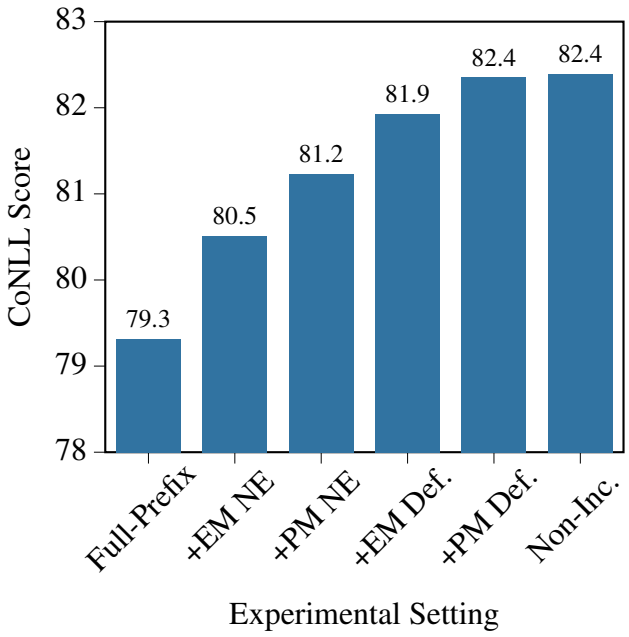


Figure 4.5: Results on the OntoNotes validation set after adding gold coreference links corresponding to specified syntactic categories. Non-Inc.=Non-Incremental, EM=Exact Match, PM=Partial Match, NE=Named Entities, Def.=Definite noun phrases.

non-incremental baseline's performance.

4.6.5 NER-Augmented Inference

Since our results show that simply adding named entities with an exact string match to their antecedent can significantly improve performance, we carry out two further oracle experiments on adding NER information to the Full-Prefix model during inference. In both approaches, we augment the Full-Prefix model's predicted clusters with information from the gold NER labels in OntoNotes.[4]

Our first method involves forcibly marking NER spans at inference time. During inference, if the current token is the left bound of a mention in the NER annotation layer, we make the model generate a mention start token `<m>`. We then allow the model to continue generating as normal, including the mention end token `</m>` and the cluster ID.[5]

The second approach takes a more conservative approach to augmenting predicted mentions. After running inference as usual, we add all mentions in the NER annotation

[4]We provide additional information on the types of NER labels in Appendix C.2.

[5]We also experimented with forcing the mention end token but found it did not lead to better results.

Model	Precision	Recall	F1
Non-Incremental Baseline	82.4	82.4	82.4
Full-Prefix Incremental	82.0	76.8	79.3
+ NER Forced Mention Start	80.5	78.6	79.6
+ NER Exact String Match	81.4	78.1	79.7
+ Pseudosingletons, 30K	81.7	78.2	79.9
+ Pseudosingletons, 60K	81.0	80.0	80.5

Table 4.6: CoNLL scores after augmenting inference with NER labels and pseudosingletons in the Full-Prefix setting. The full scores can be found in Appendix C.2.

layer that are either (1) exact string matches of each other or (2) an exact string match with a previously detected mention.

The results on the OntoNotes validation set are shown in Table 4.6. We find that improvements in F1 score are very slight. Although recall scores increase, precision scores across all three metrics decrease. The decrease in precision is particularly surprising, since we expect the method to only add high-quality exact match named entities.

We analyze errors made by the second approach on 20 documents in the OntoNotes validation set. We find that in 40% of cases, the mention in the NER layer occurs *within* the coreference layer mention; e.g. if the coreference layer contains post-modifiers such as "*Nelson Mandela*" vs. "*Nelson Mandela, who was an anti-apartheid activist*".

In another 40% of cases, we find the annotators miss marking two exact match named entities. This annotation error is particularly noticeable when the named entities occur far from each other.

The remaining 20% of errors occur due to inadvertently including named entities in copular relations, e.g. "My mother was Thelma Wahl". The OntoNotes annotation schema only includes the left side of copular relations, meaning named entities on the right side should not be included.

The success of prior models in predicting the noise in long-distance named entity coreference annotations suggests, to some degree, that they have learned a model of idiosyncrasies in OntoNotes annotations rather than to superior performance as a model of coreference itself.

4.6.6 Training with Pseudosingletons

Another potential solution to address OntoNotes' lack of singleton annotation is to augment the training dataset with predicted singleton mentions. We experiment with Toshniwal et al. (2021)'s approach with labelling *pseudosingletons* using a trained mention detector. Toshniwal et al. (2021) use the *longdoc* model based on the non-incremental SpanBERT encoder (Joshi et al., 2020) to score all text spans in each document in OntoNotes, based on which are likely to be mentions. Pseudosingletons are then selected from the top-scoring spans outside of gold mentions. Since the true number of singletons is unclear, they experiment with adding 30K, 60K or 120K pseudosingletons, and publicly release the annotations. Using their labels, we experiment with adding 30K and 60K pseudosingletons. In each experiment, we augment the OntoNotes training dataset with pseudosingletons, then train the Full-Prefix Incremental model as normal on the augmented dataset.

The results are shown in Table 4.6. Similar to the NER-augmented inference approaches discussed in the previous section, we find that adding pseudosingletons results in a small increase in F1, and gains in recall are offset by losses in precision. The best-performing setting, with 60K pseudosingletons, results in a 3.2% increase in recall but a 1.0% decrease in precision. Overall, augmenting OntoNotes with pseudosingletons improves over NER-augmented methods, but still underperforms compared to the non-incremental baseline.

As in the NER Exact String Match method, we analyze mention span errors in the 60K pseudosingleton experiment, finding many similar error cases. The pseudosingletons occasionally include slightly mismatched mention spans, for example, *Panama*, when the correct mention span is *Panama's*. As before, annotation error also affects the method's effectiveness. For example, annotators correctly label a named entity (*Rodrigo Miranda*) but fail to mark the pronouns *he* and *him* as co-referring. The pseudosingleton method then incorrectly labels these pronouns as singletons, leading to unwanted noise during training.

4.6.7 Error Samples

Figure 4.6 shows three examples of errors from the Model-Based Incremental systems on OntoNotes (top and middle) and LitBank (bottom).

The top example shows errors made by the Full-Prefix and Model-Based Incremental systems that are correctly predicted by the non-incremental baseline on OntoNotes.

her mother sat on **a school bus** for twelve hours to **Lafayette, Louisiana** – normally a three hour trip.

. . .

The owner of the nursing home Thelma Wahl was being taken to tells us by the time **{ the bus }$_8$** arrived . . .

. . .

but she was not alive when she got to the **Lafayette** area.

. . .

But last spring in a series of stories called State of Neglect they reported that **{ Louisiana's }$_{17}$** nursing homes are often poorly run . . .

The owner of the nursing home Thelma Wahl was being taken to tells **{ us }$_7$** by the time the bus arrived . . .

. . .

{ CNN }$_{10}$ made repeated attempts to contact administrators . . .

A wide and apparently an impervious boundary of { forests }$_7$ severed the possessions of the hostile provinces of France and England. The hardy colonist, and the trained European who fought at his side, frequently expended months in struggling against the rapids of the streams, or in effecting the rugged passes of { the mountains }$_{14}$, in quest of an opportunity to exhibit their courage in a more martial conflict.

. . .

Though the arts of peace were unknown to { this fatal region }$_{26}$, **{** { its }$_{26}$ **forests }$_7$** were alive with men; { its }$_{26}$ shades and glens rang with the sounds of martial music, and the echoes of **{** { its }$_{26}$ **mountains }$_{14}$** threw back the laugh. . .

Figure 4.6: Three error samples from OntoNotes and LitBank datasets. The bracket notation denotes a mention and clustering prediction, for example, *{ forests }$_7$* means the string "*forests*" belongs to cluster 7. Red, bolded text denotes errors from the corresponding model. For simplicity, we omit displaying the full set of predictions.
Top: Errors from the Full-Prefix Incremental model on OntoNotes. The model has failed to mark the first mentions of the entities and cannot revise its decisions later on.
Middle: An error from the Model-Based Incremental model on OntoNotes relating to deixis. The loss of context causes the model to miss linking "*CNN*" to the authorial "*us*".
Bottom: Errors from the Model-Based Incremental model on LitBank. The model erroneously links "*its forests*" and "*its mountains*" to previous mentions of "*forests*" and "*the mountains*", despite the mention "*this fatal region*" introducing a new context.

These errors exemplify the types of mistakes we investigate in Section 4.6.4. The model fails to predict mentions "*a school bus*", "*Louisiana*" and "*Lafayette, Louisiana*", as the incremental setting prevents it from simultaneously observing future references. In later chunks, it correctly detects the mentions "*the bus*" and "*Louisiana's*", but since it cannot revise the earlier missed mentions, it fails to correctly resolve them.

The middle example provides an error made by the Model-Based Incremental system which is correctly predicted by the Full-Prefix Incremental system on OntoNotes. Although in general, the predictions and errors between the two systems are the same, we find the removal of context can occasionally lead to incorrect predictions. This case involves deixis, as resolving the mention "*CNN*" requires understanding that the CNN journalists are also the speaker (i.e. "*us*"). However, since the context is removed after the mention "*us*" is detected, the model later identifies "*CNN*" as a separate entity, without realizing the story is being told from the perspective of journalists. In this case, adding speaker tags to the Model-Based Incremental representation may offer a simple solution.

The bottom example reveals two errors made by the Model-Based Incremental system that are correctly predicted by the Full-Prefix Incremental baseline on LitBank. The Model-Based system incorrectly links mentions "*its forests*" and "*its mountains*" to the previous mentions of "*forests*" and "*the mountains*". The correct interpretation should involve realizing that the mention "*this fatal region*" introduces a new context and that "*its forests*" and "*its mountains*" are syntactically linked to this context, and are therefore referring to a different set of forests and mountains. The Full-Prefix Incremental baseline likely avoids this error as the long context between these mentions helps to disambiguate them as separate entities; as this context is removed in the Model-Based approach, it mistakenly links them.

Modelling syntactic relationships between entities is a part of a more general task called Information Status (Halliday, 1967; Prince, 1981; Nissim et al., 2004; Markert et al., 2012), which includes related phenomena such as bridging anaphora (Clark, 1975) and split-antecedent reference (Eschenbach et al., 1989; Ingria and Stallard, 1989; Kamp and Reyle, 1993). Future work could explore how to integrate these considerations to constrain coreference predictions, and more generally how to model Information Status using the Model-Based representation, which is typically seen as a separate task (Hou, 2021).

4.7 Conclusion

In this chapter, we present a model-based, compressed representation for incremental seq2seq coreference resolution. Similar to Chapter 3, we aim to incrementally build a discourse model in order to efficiently and accurately compute coreference clusters.

Our proposed method stores and organizes tokens corresponding to entities in the text, while discarding other tokens outside of a small context window. On OntoNotes, the Model-based Incremental model scores 0.6 F1 below the Full-Prefix Incremental baseline's performance, while compressing the input by a ratio of 1.8. On LitBank, the Model-based Incremental model outperforms other SOTA approaches, demonstrating that compressing input representations can be successfully applied to the literary domain while reaching SOTA performance.

Incremental systems display a marked difference in performance relative to SOTA approaches on OntoNotes vs. LitBank datasets. In our analysis, we show the gap between non-incremental and incremental models on OntoNotes is dominated by named entities and definites. We hypothesize these cases are due to the lack of singleton annotation in OntoNotes, and we do not observe this pattern on the singleton-annotated LitBank dataset. Adding gold coreference links corresponding to exact/partial match named entities and definites improves performance; however, we find methods augmenting inference with NER and pseudosingletons struggle with annotation noise and mismatched spans.

Chapter 5

Exploration of Plan-Guided Summarization for Narrative Texts: the Case of Small Language Models

In this chapter, we switch a different topic, namely text summarization. In particular, we explore plan-guided summarization, which attempts to reduce hallucinations in summaries by grounding summaries to the source text. Although the task is not directly related to previous chapters on coreference resolution, using plans to guide summary writing can be seen as a form of discourse modelling, as the plans compactly represent important aspects of the text. In this chapter, we do not examine efficient computation like in previous chapters, but instead ask whether planning can reduce faithfulness errors in summarization. In particular, we are interested in narrative text summarization, as narrative texts' length and complexity mean they are often difficult to summarize faithfully. We hypothesize that **planning in SLMs can improve summarization in long document, narrative tasks.** Our discourse representation takes the form of a **narrative plan**, which models key events from the source text. We also implement an existing approach which models the discourse as a sequence of QA pairs. However, our results show that neither approach significantly improves on a baseline without planning. Human evaluation reveals that while plan-guided approaches are well grounded to their plan, plans contain hallucinations at the same rate as summaries, and plan-guided summaries are just as unfaithful as non-planning baselines. However, substituting an oracle, LLM-based plan markedly improves faithfulness and summary quality. The results serve as a cautionary tale to plan-guided approaches to summarization, especially for complex domains such as narrative texts.

5.1 Introduction

Modern summarization approaches based on language models generate increasingly fluent and useful summaries. However, both large language models (LLMs), and small language models (SLMs) of less than 4 billion parameters are prone to "hallucinations", where entities, dates, or assertions in predicted summaries do not faithfully reflect the source material (Kryscinski et al., 2019; Maynez et al., 2020; Lin et al., 2022; Ji et al., 2023; Wang et al., 2023). Such errors create trust and accountability issues, as users cannot rely on the outputs of the summarizer (Glikson and Woolley, 2020).

Plan-guided summarization is an approach involving conditionally generating summaries using a plan which guides summary content and ordering (Narayan et al., 2021, 2023; Huot et al., 2023). The approach has several motivations: one is to increase faithfulness in the summary, as the summary in grounded in a plan which is faithful to the source text. Another motivation is its modular design, as plans can be inspected and modified, resulting is a new summary which reflects the updated plan and is therefore more controllable. In this thesis, we focus on the former: the ability of plan-guided summarization to improve faithfulness.

Generally, plan-guided summarization works by fine-tuning a model to generate both plan and summary, with the expectation that the summary reflects planning content. Since it involves fine-tuning, plan-guided summarization is typically applied to SLMs, where full fine-tuning is more feasible compared to LLMs. Plan-guided summarization works have explored a diverse range of plan types; e.g., entity chains (Narayan et al., 2021), question-answer pairs (Narayan et al., 2023; Pagnoni et al., 2023), sentence fragments (Adams et al., 2023), semantic graphs (Hua et al., 2023), and salient noun phrases (Deutsch and Roth, 2023).

In this work, we investigate whether planning can reduce hallucination in summarizing long, narrative-based text with SLMs. Narrative texts are often much longer than common summarization domains such as news (Wang et al., 2022; Chen et al., 2022), and requires a deep understanding of states, events, and temporal and causal relations (Kryscinski et al., 2022; Kim et al., 2024). Many works have shown both SLMs and LLMs struggle at understanding long text (Levy et al., 2024; Liu et al., 2024), including narrative-based texts such as novels (Kim et al., 2024).

In Figure 5.1 we show an example output after fine-tuning Phi-3.5-mini (Abdin et al., 2024) on the SQuALITY dataset (Wang et al., 2022). While it correctly generates many aspects, the summary also contains factual errors, highlighted in red. It conflates

> ...The humanoids rally around their leader, who is wearing the same red sash and red headdress as the man. The humanoid leader marches over to Linden and Split's ship.
>
> Gravgak, the guard of the humanoid leader, confronts Linden and Split. He is worried that the sponge trees, which are camouflaging the warriors, will attack the humanoids again. He wants Linden and Split to repeat the "see-o-see-o" siren noises to keep the humanoids safe.
>
> When Linden and Split repeat the siren, the warriors run back into their hiding places. Tomboldo is the leader of the humanoids, and he is wearing the red sash and headdress. Linden gives Tomboldo a gift, a musical medallion that plays a tune when it is touched.
> ...

Figure 5.1: A predicted summary from Phi-3.5-mini (Abdin et al., 2024) after fine-tuning on SQuALITY (Wang et al., 2022). Red text marks hallucinated fragments.

the identity of characters (the leader and the man wearing the red sash and headdress are the same person), generates non-existent events (Linden and Split do not repeat the siren), and inserts correct events at the wrong time (Linden presents Tomboldo a gift much earlier in the story).

These aspects may be challenging to address for existing planning-based approaches, which typically model fine-grained details such as specific entities (Narayan et al., 2021, 2023), sentence or subsentence-level details (Adams et al., 2023; Pagnoni et al., 2023), or relationships between concepts and entities (Hua et al., 2023). While establishing key characters, dates and quantities is crucial for a summary, they do not reflect the overall narrative structure of a text.

To that end, we propose a *narrative planning* method based on *sub-events* (Daniel et al., 2003), corresponding to high-level events that outline the narrative structure of a text. Since manually annotating training plans is infeasible, we use Claude Sonnet 3.5 (v1)[1] (hereafter Sonnet 3.5) to generate synthetic training data in a single pass, leaving training and inference to the SLMs. We also experiment with QA-based planning methods (Narayan et al., 2023), and mixing both narrative plans with QA-based ones. For each method, we fine-tune transformer-based decoder models to generate both plans and gold summaries. An overview of our plan formulation is shown in Figure 5.2.

[1] https://claude.ai/

Contrary to earlier studies on plan-based summarization (Narayan et al., 2021, 2023; Hua et al., 2023), our findings indicate that neither fine-grained nor narrative-based planning improves performance on narrative-based text, with planning-based methods achieving scores comparable to baseline approaches. In order to further understand model performance, we also conduct a human evaluation of our models' outputs. We find that models, with or without planning, all hallucinate at comparable rates, resulting in similar summary quality. Although summaries are often well-grounded to their associated plans, the plans themselves may contain hallucinations, which subsequently lead to unfaithful summaries.

Analyzing synthetic plans from Sonnet 3.5 reveals that they are highly faithful to the source text, though on occasion miss relevant details from the source document. We also test the effect of replacing the models' predicted plans with the high quality plans from Sonnet 3.5 and find that coverage (+10%) and faithfulness (+6%) improve according to our manual evaluation. While one possible conclusion may be that only LLMs such as Sonnet 3.5 are reliable enough to be trusted as summarizers, SLMs are interesting in their own right as their smaller size allows them to be far more deployable and practical in low-resource settings. For this reason, methods that promise higher faithfulness in SLMs, such as plan-guided summarization, are also appealing. However, despite encouraging results in prior work, our findings suggest that there are limitations in planning-based approaches for complex, long document domains such as narrative text.

5.2 Related Work

Summarization with a planning step has taken many forms in prior work. Our most direct inspiration is Narayan et al. (2023), which we refer to as *Blueprint QA* throughout this paper. Blueprint QA uses question-answer pairs as a form of grounding for the predicted summary, with filters to ensure high-quality QA pairs. Their motivation is rooted in the "Questions Under Discussion" (QUD) theory of discourse processing (Ginzburg, 1994; Roberts, 2012). This theory presumes that a given discourse consists of assertions which can be implicitly described with an ordered set of questions and sub-questions. These questions, along with the subsequent answers in the text, specify the current structure of the text. Through the lens of this thesis, we can view Narayan et al. (2023)'s Blueprint QA plans as explicitly modelling discourse through a sequence of question-answers pairs.

In this chapter, we attempt to extend their discourse modelling approach and ask how planning can more flexibly model *events* in a narrative text. Narayan et al. (2023)'s QA plan formulation is considerably focused on entity-based questions, such as *who*, *what*, *when* and *where* questions. It formulates the answer portion of QA pairs as named entities (i.e. the output of an NER model), which may not flexibly capture abstractive aspects of narrative text. For example, a question-answer pair like "*Why did Omosla feel anguish? She felt Campbell had abandoned her.*" may be highly relevant for the summary but difficult to model using the Blueprint QA approach. Our proposed narrative approach is more flexible in this sense, as plan points are essentially free-form, as long as they are short sentences.

In this chapter, we implement a Blueprint QA baseline; however, our implementation should been seen as inspired from (Narayan et al., 2023) and is not an exact implementation. We describe their method and our implementation of it, including technical differences, more thoroughly in Section 5.3.

We also note that Narayan et al. (2023) present an *iterative* model that is roughly analogous to the incremental settings in previous chapters of this thesis. In this setting, the model generates one QA plan point and summary sentence at a time. The current plan point and summary sentence then serve as context for the next iteration. We did implement and experiment with this approach with our narrative plan formulation, but we found it did not generate coherent outputs, and we do not explore it further in this chapter.

Pagnoni et al. (2023) also presents plan-based summarization method based on QA pairs. Source-text sentences are converted into question-answer pairs as a pre-training objective. Sentence-level questions may be viewed as a form of higher-level planning; however, in this work we argue that narrative structure traverses sentence boundaries, and investigate generating plans across the entire document instead of a single sentence.

Deutsch and Roth (2023) train a model to mark salient NPs in the source document, then generate a summary conditioned on the augmented document. Adams et al. (2023) generate content plans from extractive elementary discourse units, then re-write and re-rank the candidate summaries. Although their model uses abstractive re-writing, the plans are ultimately based on extractive fragments which are unlikely to represent higher-level plans.

Perhaps closest to our work, in terms of motivation, is Hua et al. (2023) who present a distinct method using abstract meaning representation (AMR) graphs as a source of grounding. Their work explicitly captures high-level information, but is very involved,

requiring an AMR parser, a coreference resolution model and an additional module to align concepts and words, on top of the summarization model. Our approach presents a simpler, less involved approach to high-level planning where narrative structure is represented as short sentences instead of complex AMR graphs.

Planning is used outside of summarization, and most relevant to our work is that of Godbole et al. (2024). They use planning to generate both paragraph descriptions and QA pairs in order to generate biographies about individuals. However, their work focuses on using planning with retrieval to augment the model's parametric knowledge, while in our task, we are interested in knowing whether the model can understand and reason with long contexts in a narrative structure. Similarly, Shao et al. (2024) prompts LLMs to write Wikipedia articles by generating outlines, diverse perspectives and conversational QA-pairs as a plan. Their work similarly focuses on using LLMs to retrieve helpful documents for writing the article, rather than summarizing long narrative text. Chawla et al. (2024) also investigate planning techniques for generating knowledge-grounded dialogues, finding that content planning offers mixed results in improving dialogue quality.

Lastly, in relation to the incremental approaches of the two previous chapters, we note that incremental summarization methods also exist for processing long text. These approaches work by maintaining a structured representation as the text is read, which is then dynamically updated or augmented as new information is encountered (Chowdhury et al., 2024; Chang et al., 2024; Hwang et al., 2024, 2025). Incremental summarization approaches face distinct challenges, such maintaining coherence across text chunks (Chang et al., 2024), merging redundant information (Hwang et al., 2025) and adding excessive details to summaries (Hwang et al., 2024). For summarization domains with very long input length, using an intermediate representation such as plans or chain of thought remains the only viable alternative to the incremental summarization approach.

5.3 Method

In this work we experiment with an existing QA plan-based summarization method of Narayan et al. (2023) (our re-implementation) and our novel *narrative-based* planning methods that incorporate **coarse plans**. In this section we discuss the implementation details for each approach. Since summarization datasets do not come with plans, we generate them using specialized models or LLMs. We first detail in 5.3.1 how to create plans from summarization data, and provide statistics about each plan type in Table

		SQuALITY	SummScreen-FD
Blueprint QA	# tokens	103.6	25.6
	# QA pairs	5.5	2.4
Coarse Plans	# tokens	212.6	111.4
	# plan points	12.2	6.7
Coarse Plans + QA	# tokens	271.0	127.2
	# plan points (inc. QA)	16.1	7.9

Table 5.1: Plan statistics across the three plan formulations. We report the average number of tokens and average number of plan points per document across all of training, validation and test datasets. Note that the average number of QA pairs and plan points in SummScreen-FD is lower than SQuALITY as its summaries are considerably shorter (141 tokens/summary in SummScreen-FD vs. 591 in SQuALITY).

5.1. We then explain in 5.3.2 how we fine-tune SLMs to generate both plans and gold summaries.

5.3.1 Training Plans

5.3.1.1 Blueprint QA

In Narayan et al. (2023)'s Blueprint QA plan formulation, a question generation model overgenerates a sequence of questions relating to entities, times, and places in the document of interest, which are then filtered for quality. The overall goal is to generate a list of question-answer pairs which are highly accurate and relevant to the summary. We describe our implementation of Narayan et al. (2023)'s Blueprint QA approach, then specify the technical differences from their original implementation.

The first step is to extract meaningful answers from the summary. We use SpaCy (Honnibal et al., 2020) to extract named entities including dates, time expressions and quantities from the gold summary. These extracted entities serve as potential answers for the plan.

We then generate questions using MixQG[2] (Murakhovs'ka et al., 2022), a state-of-the-art question generation model. MixQG is an encoder-decoder model fine-tuned on 9 question generation datasets covering a wide variety of answer types, such as:

[2]We use the T5-3B variant.

- Yes/No, e.g. BoolQ (Clark et al., 2019)
- Multiple choice, e.g. MCTest (Richardson et al., 2013)
- Extractive, e.g. SQuAD (Rajpurkar et al., 2016)
- Abstractive, e.g. NarrativeQA (Kočiský et al., 2018)

Provided with an answer and a text passage as input, MixQG generates an appropriate question from the text. We pass each extracted entity from SpaCy along with its source sentence and two preceding sentences as the context to MixQG, which generates a question for each entity about the summary.

Lastly, we filter low quality QA pairs using Round-trip Consistency, Rheme and Coverage filters. These filters ensure the final set of QA pairs are accurate, informative and non-redundant. We describe each filter below:

- The Round-trip Consistency check (Alberti et al., 2019a) uses a question answering model to verify the extracted answer correctly answers the question from the QG model. If the question answering model output differs from the extracted answer, the QA pair is discarded.
- The Rheme filter promotes keeping QA pairs that describe new information in the text, as opposed to already known information. The summary is first deterministically chunked into propositions using key words such as punctuation, relative pronouns, prepositions and coordination. QA pairs whose answers do not match the rightmost token(s) of a proposition are filtered out. If multiple QA pairs match a single proposition, the one with the longest answer is kept. We notice in certain cases that the Rheme filters removes all QA pairs from the plan. In these cases, we do not apply it.
- Lastly, the Coverage filter removes overlapping QA pairs. The summary is tokenized into a bag of tokens and QA pairs are greedily matched by highest lexical overlap. The matched tokens are then removed and the process is continued until the bag is empty, there are no more QA pairs, or there is no more lexical overlap. Any remaining QA pairs are then discarded.

We note several important technical differences between our implementation and Narayan et al. (2023)'s original Blueprint QA formulation:

- Their base model is an encoder-decoder model with 3B parameters, namely LongT5 (XL) (Guo et al., 2022). In all our experiments, including the Blueprint QA setting, we use Phi-3.5-mini, a *decoder-only* model with 4B parameters (Abdin et al., 2024). We also fine-tune Phi-3.5-mini with LoRa (Hu et al., 2022), while Narayan et al. (2023) use standard fine-tuning. We describe the Phi-3.5-mini model in more detail in Section 5.4.2.1.

- Narayan et al. (2023)'s question generation model uses the T5-11B checkpoint (Raffel et al., 2020), fine-tuned on the SQuAD reading comprehension dataset (Rajpurkar et al., 2016). In contrast, we use MixQG (Murakhovs'ka et al., 2022), in particular the T5-3B variant. In comparison to Narayan et al. (2023)'s QG model, MixQG is smaller (3B vs. 11B parameters) but trained on more datasets (9 vs. 1).

- Further, Narayan et al. (2023) does not restrict the context length when inputting the summary to the question generation model. However, we find that passing the full summary has detrimental effects on SQuALITY, as it contains much longer summaries on average (591 tokens/summary, compared to roughly 100 tokens in the datasets used by Narayan et al. (2023)). We instead use the sentence containing the extracted entity and the previous two sentences as the summary context.

- Our question modelling model used in the Round-trip Consistency check is a pre-trained ELECTRA-Large model (Clark et al., 2020) fine-tuned on the SQuAD 2.0 dataset (Rajpurkar et al., 2018), the same used in QAEval (Deutsch et al., 2021). However, Narayan et al. (2023) rely on Alberti et al. (2019a)'s original formulation which uses BERT as their base model (Alberti et al., 2019b).

- Narayan et al. (2023) present QA pairs with the answer first followed by the question; they find this slightly outperforms presenting the question first. We opt to keep question-first ordering, following other work in high-level QA plan-based summarization (Pagnoni et al., 2023).

5.3.1.2 Coarse Plans

Since manually annotating sub-events is infeasible, we automatically generate the coarse plans containing these sub-events by prompting Sonnet 3.5, which is known for its strong text understanding and generation capabilities (Anthropic, 2024). We prompt

1. Generate coarse plans:
We prompt Claude 3.5 Sonnet to create synthetic plans in a single pass. Each plan point is linked to its source sentence.

1. Linden and Split land their ship. [1,2]
2. The crew witnesses an attack. [5,6]
3. The humanoids rally around their leader. [7]

2. Extract QA pairs:
For each plan point, we pass all named entities to a QG model to generate relevant questions.

Q: Who is the captain? A: Linden
Q: Who hits the siren? A: Split
Q: Who is the humanoids' leader? A: Tomboldo

3. Filter QA pairs:
We apply Blueprint QA filters to remove low-quality QA pairs, as well as identical QA pairs across plan points.

Figure 5.2: A visualization of the steps in creating training plans. First, coarse plans are generated with an LLM. Then, named entities are extracted from the relevant source sentences, and a question generation model generates a question for each entity. Lastly, low-quality questions are filtered out and the QA pairs are attached to the coarse plans. The figure shows one QA pair per plan point; in practice, the number of pairs may vary.

. . . Linden and Campbell think their ship is out of sight, and watch a ritual that the man is performing to the setting sun. The crowd of people continues to increase, and Linden notices that the landscape is moving: trees are shifting in the ground. He and Campbell stay in the ship and observe the various types of clothing and the ritual itself, as well as the moving trees which seemed to be moving to attack the people. They are indeed warriors starting an attack, and started swinging weapons. Linden tells Campbell to start the siren on their ship to scare away the attackers, and the first man they'd seen, presumably the leader, starts towards the ship. Once they are close enough, it is obvious that the humanoids don't have eyebrows or eye lashes. Captain Linden hands the leader a medallion that plays a song, as a token of friendship. Tomboldo, the leader, starts a round of introductions through a lot of gesturing. Linden hopes to learn about the Serpent River through the people to understand its cultural significance, and these people start to ask about the siren noises. The warriors attack again and panic ensues, pushing the humans to use weapons this time. Gravgak, the guard who had been escorting the humans, is knocked down. As Linden tries to tend to him, Gravgak knocks him out with his club. Linden is unconscious for a few weeks, and Vauna, Tomboldo's daughter, spends a lot of time by the Captian's side. . . .

1. Captain Linden and Campbell lead first manned expedition to planet.
2. They observe a silver river-like feature on the planet's surface.
3. Campbell spots humanoid beings through his telescope.
4. They witness people emerging from an underground city.
5. Linden and Campbell observe a ritual and moving trees.
6. Tree-like warriors attack the humanoid people.
7. Linden uses ship's siren to scare away attackers.
8. Humans make first contact with the planet's inhabitants.
9. Warriors attack again, prompting humans to use weapons.
10. Gravgak knocks Linden unconscious.
11. Linden recovers, forms connection with Vauna.
    Q: How long is Linden unconscious? A: a few weeks
12. Humans learn about the Benzendellas and their technology.
    Q: What does Linden not know about? A: the Serpent River
13. Gravgak apologizes for injuring Linden.
14. Tomboldo reveals the Serpent River is called Kao-Wagwattl.

Figure 5.3: An example fragment of a reference summary from SQuALITY (top), with an annotated plan from Sonnet 3.5 (numbered sentences) and QA pairs. The full example can be found in Appendix D.1.

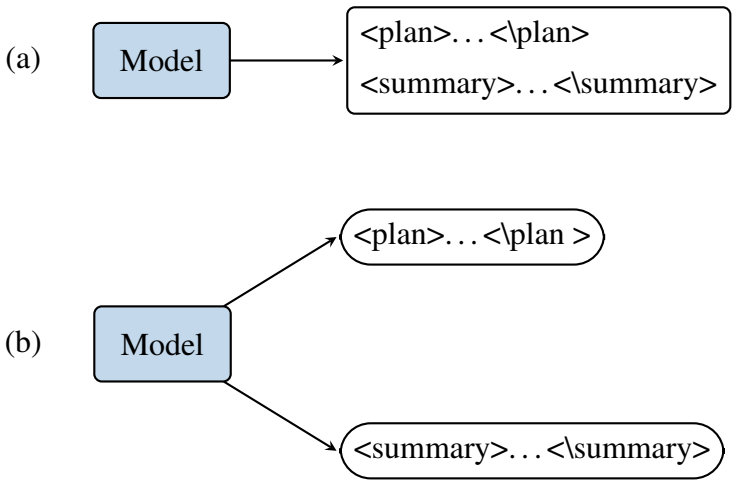


Figure 5.4: The two training methods we explore: (a) In the End-to-End (E2E) setting, the model generates both plans and summaries in a single decoder pass. (b) In the Multi-Task setting, the model separately learns to generate plans and summaries. At inference time, we pre-fill the decoder with the <summary> token to prompt the model to generate the summary.

the model to extract key sub-events from the text, without including specific details which are more likely to be hallucinated (Ji et al., 2023; Wang et al., 2023).[3] Note that we only prompt Sonnet 3.5 in a one-shot basis to generate coarse plans, and we do not employ it during training or inference.

Each plan consists of numbered *plan points*, where each plan point corresponds to one sub-event in the text. Our definition of a sub-event is primarily inspired from Daniel et al. (2003), but is also similar to the *atomic claims* defined in Gunjal and Durrett (2024). Each sub-event should progressively and succinctly describe key events from the story, with minimal details such as dates or locations. To encourage the LLM to output the right plan format, we provide in-context examples of the type of plan we expect (Brown et al., 2020). Examples of the types of plans we expect can be seen in the in-context examples in Appendix D.2.

We generate plans using temperature set to 1.0, nucleus sampling disabled ($top\_p = 1.0$), and limiting the output to 400 tokens. These settings encourage diverse plans and consistent lengths. Importantly, we prompt the model to construct the plan based on a gold summary s, instead of the source document d. This ensures the predicted plan matches the narrative of the gold summary. An example is shown in Figure 5.3 (without

[3] See Appendix D.2 for the exact prompt.

QA pairs).[4]

5.3.1.3 Coarse Plans + QA

We also experiment with a setting which combines the benefits of both the higher-level, event-focused planning and the lower-level, detail-focused planning. To this end, we combine coarse and fine-grained QA into a single plan. Question generation models typically require an answer and a context to generate an associated question. In our case, we do not directly input plan points to the question generation model, since plan points are designed to be free of the desired fine-grained details. Therefore, our first step is to link plan points back to the original summary sentences, which contain more details suitable for fine-grained question generation. The original source sentence that gave rise to the plan point contains relevant details abstracted away from the plan.

To facilitate this linking, we ask Sonnet 3.5 to provide a citation for each plan point, similar to Fierro et al. (2024). We number all sentences in the gold summary and ask the model to end each plan point with the relevant sentence number(s).[5]

Given the linked source sentences, we apply the Blueprint QA procedure as before, using the linked sentences and two preceding sentences as context for the QG model. After applying the Blueprint QA filters, each generated QA pair is then attached to the original plan point. We then additionally filter out identical QA pairs *across* plan points.

5.3.2 Training

We use two training methods to generate plans and summaries, following Narayan et al. (2023). We visualize the two settings in Figure 5.4.

End-to-End In this setting, the model produces both plans and summaries in a single pass. After processing document d, the model first produces the plan p, effectively modeling $\Pr(p|d)$. The decoder then generates the summary s based on both p and d, modeling $\Pr(s|d,p)$.We prefix the plan with the string <plan>, and similarly prefix the summary with <summary>. This allows the model to directly refer to both the plan and the source document via its attention mechanism when generating the summary. However, it comes at the cost of generating longer sequences.

[4]The full coarse plan is shown in Appendix D.1.
[5]The exact prompt is shown in Appendix D.2.

Multi-Task The end-to-end model may potentially struggle with extremely long generation when predicting both plan and summary in a single output. For this reason, we also experiment with training the model to generate plans and summaries separately in a multi-task setup. Each document is separately paired with the generated training plan and the gold summary, and the model learns to generate the plan and summary in separate tasks. Although the model is less likely to be grounded in the summary in this setting, avoiding long generation may produce higher quality summaries.

5.4 Experiments

5.4.1 Datasets

As described in Section 2.3.2, we choose two long document summarization tasks with a clear event-focused narrative structure: SQuALITY (Wang et al., 2022) covers short stories from Project Gutenburg, and SummScreen (Chen et al., 2022) is based on TV show transcripts. As mentioned previously, we use SummScreen's FullDreaming (FD) subset, following prior work (Hua et al., 2023; Narayan et al., 2023).

5.4.2 Model

5.4.2.1 Base architecture

We use the Phi-3.5-mini model in all experiments (Abdin et al., 2024), a 3.8B parameter decoder-only transformer architecture pre-trained on 3.3T tokens.[6] The model uses 3072 hidden dimension, 32 heads and 32 layers. The default context length during training is 4K tokens, but the authors extend it to 128K by interpolating position embeddings using LongRoPE (Ding et al., 2024). This feature allows us to directly input long documents for the summarization task, making it an ideal candidate for long document summarization with SLMs. Despite its compact size, Phi-3.5-mini performs comparably to other leading LLMs in a variety of reasoning tasks.

The exact training data sources are not publicly available, but they are described in Abdin et al. (2024) as "heavily filtered publicly available web data... from various open internet sources, as well as synthetic LLM-generated data". The Phi-3 blog[7] provides

[6]We also experimented with LongT5 (Guo et al., 2022), but got very poor summarization results across all datasets, which we omit here.

[7]https://news.microsoft.com/source/features/ai/the-phi-3-small-language-models-with-big-potential/

more information on the training data curation process: the researchers start with a publicly available (but undisclosed) set of data sources, then use an LLM to augment the dataset by synthesizing variations on the original data set. The resulting data is then repeatedly filtered and re-augmented by the LLM until the corpus reaches the desired size.

5.4.2.2 Training Details

In each experiment, the model is fine-tuned with a standard language modelling objective to maximize the likelihood of the labels. Depending on the task, the labels will either consist of the plan token labels (Multi-task), summary token labels (Multi-task), or plan+summary tokens labels (E2E). All models are trained with a learning rate of 0.001 and a batch size of 32 (per device batch size of 4). We fine-tune the model using LoRa[8] (Hu et al., 2022) for 100 epochs and checkpoint the model using the validation loss. We truncate input documents to 8K tokens. Since plans may be long, output length is a key consideration and we perform hyperparameter search over the output generation length with 512, 768 and 1024 tokens. Accordingly, we set the output length in all E2E settings to 768 and otherwise use 512 tokens. The E2E setting requires a longer generation size since it generates both plan and summary in a single decoder pass. During inference, we do not sample, and use beam search with 5 beams and a length penalty of 0.8.

Additionally, for both settings, we forbid repeated ngrams in the generated output for any ngram of length 5, where ngram size is counted as the number of subtoken units. We use this value after running a hyperparameter search over $\{4, .., 16\}$ during early experiments.

For training, we use a p4d.24xlarge instance available on Amazon Sagemaker which includes 8 NVIDIA A100 GPUs. It costs $32.77 per hour. The training time varied from 4 – 24 hours for different settings.

5.4.3 Evaluation

5.4.3.1 Automatic Metrics

As described in Section 2.3.3, we use four automatic metrics for our evaluation. We measure summary quality with ROUGE (Lin, 2004), which measures word-level overlap

[8] We use $r = 24, \alpha = 32$ and LoRa dropout of 0.05

with a reference summary. For SQuALITY, which includes 4 gold summaries per text, we compute ROUGE for each label and take the maximum of the four scores. We report ROUGE-1, -2 and -L with Porter stemming.[9]

We measure three faithfulness metrics: AlignScore (Zha et al., 2023), QAFactEval (Fabbri et al., 2022) and FineSurE (Song et al., 2024). AlignScore is an NLI-based metric, QAFactEval is a QA-based metric, and FineSurE is a LLM-based metric measuring both faithfulness and summary quality. For FineSurE, we use Claude Sonnet 3.5 (v1) as the LLM. Although using the same LLM for generating plans and evaluation is non-ideal as prior work has shown they are biased towards their outputs (Kim et al., 2024), we are restricted in our use of available LLMs.

5.4.3.2 Human Evaluation

Automatic evaluation metrics are known to be unreliable (Kryscinski et al., 2019; Kim et al., 2024), with different metrics capturing different dimensions of the outputs. Therefore, we also conduct human evaluation of 5 randomly selected documents from SQuALITY. The annotation is conducted by 2 authors of our published work (Grenander et al., 2025), following several discussions on best practices and establishing an evaluation rubric. It covers all settings: 8 models $\times$ 5 summaries $\times$ 4 evaluation metrics, for a total of 155 annotations (some metrics do not apply universally).[10]

After reading each document, we list important high-level events from the story. We measure **coverage** as the proportion of high-level events from the story that are present in the generated summary.

For each predicted summary, we extract *atomic claims* (Kim et al., 2024), as we find using a finer level of granularity allows for more consistent annotation. We aim to extract atomic claims that are indivisible minimal facts, which are context-independent and describe a property of an entity or a relationship between two entities.[11] We measure **faithfulness** as the proportion of the model's atomic claims that are supported by the text, and **conciseness** as the proportion of atomic claims that are present in at least one reference summaries.

We are also interested is how well the predicted summary follows the generated plan. For all settings generating plans, we compute **grounding** as the proportion of plan points that are present in the generated summary.

[9]We use the Hugging Face implementation from `https://huggingface.co/spaces/evaluate-metric/rouge`.

[10]Additional details can be found in Appendix D.5.

[11]Examples can be found in Appendix D.5.

We compute interannotator agreement by reusing one annotator's extracted facts for one story across the 8 model settings (i.e. 8 generated summaries in total). Using these extracted facts, we doubly annotate coverage, faithfulness, conciseness and grounding metrics. The overlapping annotations cover 264 atomic facts (for faithfulness and conciseness), 104 high-level events (for coverage) and 52 plan points (for grounding). We compute Cohen's kappa across all four evaluation dimensions and observe $\kappa = 0.823$, indicating strong agreement.

5.4.4 Compared Systems

5.4.4.1 Phi-3.5-mini

Our primary baseline is the Phi-3.5-mini model trained for summarization without using any plans. We fine-tune the model with the document and summary as input, with loss computed over the summary tokens. We prefix each example with a basic summarization prompt.[12]

5.4.4.2 Socratic

For SQuALITY we also report results from Pagnoni et al. (2023) using the BART-large model (Lewis et al., 2020). The model is pretrained with their **Socratic** objective on the Books3 corpus, from the Pile (Gao et al., 2020) before being fine-tuned on SQuALITY. The Socratic objective generates questions about sentences from the source text. The model is then pre-trained to generate question-answer pairs consisting of sentence-level questions and sentences from the source text.

5.4.4.3 Claude Sonnet 3.5 (v1)

We also report results obtained by directly prompting Claude Sonnet 3.5 (v1). We use a temperature of 0, max generation length of 512, and $top\_p = 1.0$. The full prompt is shared in Appendix D.4.[13] Since we do not have access to the information about Claude training data, we cannot be sure that the model has not seen the documents in SummScreen or SQuALITY. For this reason, we add Claude results mostly for reference, rather than as a comparative baseline.

[12]The prompt can be found in Appendix D.3.

[13]The final prompt we use for Claude summarization does not ask the model to generate plans. We tried prompt variants that asked the model to plan-and-summarize, but these worked worse than our final prompt.

5.4.4.4 Other Models

We include other contemporary models which perform strongly on SQuALITY and SummScreen-FD: LTRSum (Sotudeh and Goharian, 2024), the highest-scoring model we are able to find on SQuALITY, and BART-LS (Xiong et al., 2023), the state-of-the-art model on SummScreen-FD.

5.5 Results

Results on SQuALITY and SummScreen are shown in Table 5.2.

5.5.1 Summary Quality

In the Multi-Task setting, summary quality is generally comparable to the baseline method without planning. The Coarse Plans, Multi-Task model achieves the highest ROUGE scores overall on both SQuALITY and SummScreen, though improvements over the Phi-3.5 baseline are all under 1 F1 point. This finding suggests that the methods generate summaries with similar quality, as measured by ROUGE.

The same trend does not hold for E2E methods, which see a steep drop in ROUGE scores. This likely occurs due to the difficulty of generating much longer text in a single pass, as observed by Narayan et al. (2023).

Across different plan formulations, we observe little change in ROUGE performance. Instead, these methods perform similarly to the baseline on both datasets, suggesting that different plan formulations do not significantly affect summary quality.

The Claude 3.5 Sonnet baseline outperforms E2E methods but underperforms against other fine-tuned methods on both datasets, despite undoubtedly being much larger than Phi-3.5-mini. Fine-tuning likely confers advantages in terms of either summary quality or in-domain knowledge (e.g., the expected summary style).

Lastly, we note our methods outperform LTRSum (Sotudeh and Goharian, 2024) on SQuALITY and underperform compared to BART-LS (Xiong et al., 2023) on SummScreen. However, in both cases we feel the methods are not directly comparable. On SQuALITY, our Phi-3.5-mini base model is much larger (3.8B vs. 406M), meaning a performance boost is expected. On SummScreen, the BART-LS uses a 16K token context size, while we are limited to a 8K context window due to resource constraints.

Model	R-1	R-2	R-L	AS	QAFE	FSE
SQuALITY						
LTRSum (Sotudeh and Goharian, 2024)	46.11	14.68	24.23	-	-	-
Socratic (Pagnoni et al., 2023)	46.31	14.80	22.76	-	-	-
Claude 3.5 Sonnet	47.52	12.07	20.03	**59.76**	1.42	**91.8**
Phi-3.5-mini	51.47	16.44	23.36	53.22	1.83	66.3
Blueprint QA, E2E	40.82	13.14	20.23	57.30	1.75	63.4
Blueprint QA, Multi-Task	51.54	16.35	23.75	53.85	1.76	67.4
Coarse Plans, E2E	39.10	12.04	20.15	52.38	1.63	61.3
Coarse Plans, Multi-Task	**51.66**	**16.58**	**24.27**	52.25	1.78	66.1
Coarse Plans + QA, E2E	37.38	11.41	19.86	51.86	1.96	57.8
Coarse Plans + QA, Multi-Task	50.57	16.12	23.90	54.39	**2.21**	67.1
Pre-filled Claude Plans, E2E	48.57	16.26	25.51	50.04	1.97	71.9
SummScreen-FD						
BART-LS (Xiong et al., 2023)	39.1	10.7	33.5	-	-	-
Claude 3.5 Sonnet	28.87	7.27	14.91	**57.55**	1.69	**95.5**
Phi-3.5-mini	31.50	7.40	18.82	45.27	1.82	44.0
Blueprint QA, E2E	26.62	5.89	16.93	49.15	**1.88**	55.8
Blueprint QA, Multi-Task	31.45	6.92	18.54	46.23	1.77	45.7
Coarse Plans, E2E	29.34	6.01	17.55	46.49	1.66	44.3
Coarse Plans, Multi-Task	**31.98**	**7.55**	**19.02**	47.02	1.84	47.0
Coarse Plans + QA, E2E	28.83	5.91	17.19	46.16	1.66	40.3
Coarse Plans + QA, Multi-Task	31.58	7.30	18.62	45.93	1.85	46.2

Table 5.2: Results on SQuALITY and SummScreen. AS=AlignScore, QAFE=QAFactEval, FSE=FineSurE. Blueprint QA, E2E and Multi-Task, are our implementations of Narayan et al. (2023). Scores for Socratic (Pagnoni et al., 2023) are as reported in their paper; all other scores are from our own implementations. We bold the highest scores among the methods we evaluate, as other models are not directly comparable to ours (see the discussion in Section 5.5.1).

5.5.2 Faithfulness

We see mixed results across faithfulness metrics. Looking at AlignScore and QAFactEval, many methods surpass others in one metric or one dataset, but fail to replicate the result across datasets or across metrics. For example, while Claude 3.5 Sonnet achieves the highest AlignScore on both datasets, it scores lower than the Phi-3.5 baseline on QAFactEval. Similarly, Coarse Plans + QA, Multi-Task achieves the highest QAFactEval on SQuALITY, but scores lower than Blueprint QA, E2E on AlignScore across both datasets. These results lead us to question the reliability of these metrics, as they do not clearly suggest a better model.

On FineSurE, Sonnet 3.5 eclipses other models across both datasets. However, we are wary of this result for two reasons. First, AlignScore and QAFactEval do not corroborate massive faithfulness advantages over other methods. Secondly, previous work has found LLM auto-raters tend to favor their own outputs (Kim et al., 2024). Other settings achieve similar scores, except for the Blueprint QA, E2E setting on SummScreen.

In summary, the mixed results lead us to question the reliability of existing faithfulness metrics and whether any method is truly outperforming the baseline. This issue motivates our human evaluation in Section 5.5.3.

5.5.3 Human Evaluation

Human evaluation results are shown in Table 5.3. We note E2E coverage scores tend to be lower than their Multi-Task counterparts. We observe that the E2E model occasionally outputs a lengthy plan, leaving little space for the resulting summary before reaching the maximum token count. This behavior negatively affects coverage as the shortened summary cannot cover all relevant story points. In the Blueprint QA E2E model, we notice a particular failure case where the model's summary is a nearly identical copy of the QA pairs, resulting in summaries with especially low coverage.

We observe that faithfulness and conciseness scores are roughly equal across all settings. Faithfulness scores range from 68 to 77%, with the highest faithfulness score achieved by the Blueprint QA, Multi-Task setting. However, the highest scoring methods, Blueprint QA, Multi-Task, and Coarse Plans + QA, Multi-Task, score only 3.01% and 2.06% higher than the Phi-3.5-mini baseline, and we strongly suspect the difference is not meaningful. In order to validate our hypothesis, we count the number of unfaithful atomic facts across the three models. We find the Phi-3.5-mini

Model	Cov.	Faith.	Conc.	Ground.
Phi-3.5-mini	46.82	75.33	84.45	-
Blueprint QA, E2E	36.82	67.99	72.27	70.00
Blueprint QA, Multi-Task	56.36	**78.41**	82.69	14.00
Coarse Plans, E2E	43.15	74.09	84.95	91.68
Coarse Plans, Multi-Task	57.01	74.15	**88.62**	71.92
Coarse Plans + QA, E2E	37.71	71.26	87.05	**97.78**
Coarse Plans + QA, Multi-Task	**64.10**	77.39	82.81	64.79
Pre-filled Claude Coarse Plans, E2E	74.19	84.57	92.10	87.77

Table 5.3: Human evaluation results on SQuALITY. We bold the highest score in each column, excluding the "Pre-filled Claude Coarse Plans, E2E" oracle setting.

model generates 34 errors (out of 144 atomic facts), the Blueprint QA, Multi-Task model generates 32 errors (out of 146), and the Coarse Plans + QA, Multi-Task setting generates 33 errors (out of 147). The tiny variation in errors supports our conclusion that no model setting meaningfully outperforms the Phi-3.5-mini baseline. We also include annotated error samples in Section 5.6.3.

Although we do not report faithfulness scores on the Multi-Task plans, we note qualitatively that these plans contain hallucinations at roughly the same rate as the E2E plans. Similarly, conciseness scores range from 72 to 88%, though we note that the Blueprint QA E2E model scores more than 10 points lower than any other setting.

On grounding, we observe that E2E methods based on Coarse plans are remarkably tightly grounded to their plans, achieving above 90%. This effect contrasts with multi-task settings where grounding scores range from 14 to 71%. The effect can be readily explained by the different task setups: E2E models directly use their plans via the decoder's attention mechanism, whereas Multi-Task models only generate plans during training.

Dataset	Coverage	Faithfulness	Redundancy	Cit. Acc.
SQuALITY	84.44	98.26	0.87	91.26
SummScreen	98.67	98.63	0.0	100.0

Table 5.4: Analysis of coarse plans generated by Claude Sonnet 3.5 v1. Cov.=Coverage, Faith.=Faithfulness, Red.=Redundancy, and Cit. Acc.=Citation Accuracy.

5.6 Analysis

5.6.1 Claude Synthetic Plans

We would like to know if plans generated by Claude Sonnet 3.5 (used for training) correctly correspond to relevant sub-events in the source text. To this end, we manually inspect 20 coarse plans generated by Sonnet 3.5 on SummScreen and SQuALITY, covering 115 generated plan points on SQuALITY and 73 on SummScreen. Similar to our human evaluation, we analyze if each generated plan point is factual to the source summary (faithfulness), and whether each key fact in the reference summary is contained in the generated plan (coverage). We inspect each plan for redundant plan points, and report the percentage of redundant points among all generated points. Finally, we verify the accuracy of citations by checking if each citation provides a suitable source sentence for the associated plan point.

The results are shown in Table 5.4. In general, we find that Sonnet 3.5's plans are very high quality. Across both datasets, faithfulness scores are above 98%, and we find that Sonnet 3.5 rarely includes hallucinated details in the coarse plan.

Coverage performance is similarly high on SummScreen but lower on SQuALITY. We find that the summary length has a significant effect on Sonnet 3.5's ability to extract key events, and on longer summaries, Sonnet 3.5's plans tend to omit key details more often. On SummScreen, where summaries are generally shorter, this effect is less noticeable.

On both datasets, Sonnet 3.5's plans are highly non-redundant – we find only one case of redundant plan points, in SQuALITY. Likewise, the generated citations are highly accurate, scoring above 90% on SQuALITY and flawlessly on SummScreen.

5.6.2 Pre-filled Claude Plans in E2E Setting

Our results have shown that that E2E settings tend to be very grounded in their plans, but factual errors in planning ultimately lead to unfaithful summaries. On the other hand, Sonnet 3.5 plans are much higher in quality than the predicted plans across all of our tested settings. Putting the two together, we are interested to know whether substituting Sonnet 3.5 plans for the predicted ones could lead to better summaries overall. Although this implementation is an oracle setting, it can give insight into whether high-quality plans can lead to better summaries.

We run the Coarse Plans E2E settings as before (without QA) on SQuALITY, but replace their predicted plans with the Sonnet 3.5 plans. We then score the generated summaries as before.

The automatic evaluation results are shown in Table 5.2 (bottom row of the SQuALITY section). Using the pre-filled plans greatly improves ROUGE scores compared to the normal E2E settings. Although it is tempting to draw the conclusion that the summaries are therefore higher quality, we suspect this effect is mainly because E2E settings tend to overgenerate plans, crowding out their summaries. Since the plans are now pre-filled, E2E models are no longer at risk of generating excessively lengthy plans. Faithfulness scores on AlignScore and QAFactEval decrease slightly, though we again question the reliability of these metrics.

Our manual evaluation results are shown in Table 5.3. E2E with Sonnet 3.5 plans maintains good grounding to plans as before, although with a slight decrease compared to other E2E methods. We observe a notable increase in coverage (+10 over the next highest), as the Sonnet 3.5 plans' coverage is much higher than other plan-guided models. For the same reason, conciseness scores also increase, as Sonnet 3.5 plans are typically concise.

Faithfulness also slightly increases by 6.16% over the next highest score. We notice that although planning content is highly factual, the E2E model tends to follow the plan but then veer off into hallucinated generation. Overall, pre-filling plans increases scores across coverage, faithfulness and conciseness scores. Of course, the main challenge is generating very high quality plans.

5.6.3 Error Samples

Figures 5.5 and 5.6 show sample outputs from four of the model settings we evaluate in this chapter: the Phi-3.5-mini (non-planning) baseline, Coarse Plans E2E, Blueprint

QA E2E and Coarse Plans + QA Multi-Task. Red text in each output sample represents unfaithful facts. We note that all four models hallucinate at roughly equal rates, and plan points, whether QA pairs or Coarse plans, also include unfaithful content. Moreover, although the Blueprint QA, Multi-Task and Coarse Plans + QA, Multi-Task models achieve the highest faithfulness scores in our human evaluation results, we note that summaries are not qualitatively different, as discussed in Section 5.5.3.

We categorize faithfulness errors roughly as follows:

- **Entity errors:** The model generates a correct event but substitutes an incorrect character. For example, in "*the leader of the people, Tombold*" (Coarse Plans + QA, Multi-Task), the character is actually named "*Tomboldo*". In "*The natives nurse Campbell back to health*" (Coarse Plans, E2E), it is actually "*Linden*" who is nursed back to health.

- **Non-existent events:** The model hallucinates an incorrect event. For example, "*Gravgak is killed in the process*" (Coarse Plans, E2E) does not happen in the story.

- **Temporal errors:** The model generates events that do happen in the story, but in the wrong order. For example, "*Linden gives Tomboldo a gift*" (Phi-3.5-mini) does occur in the story but at a much earlier point in time.

We note that these errors are also included in the long document summarization error taxonomy presented in Kim et al. (2024).

Phi-3.5-mini (No Plan)

Captain Linden and his companion, Split, land their ship on the surface of an alien planet. The planet is inhabited by humanoid creatures that resemble the humans of Earth. The planet is home to the Serpent River, a long rope-like structure that crawls across the surface of the planet. Split and Linden are on a mission to learn more about the Serpent River and the planet's inhabitants. Linden and Split observe two humanoids emerge from the ground. The two humanoids are a man and a woman. The man is wearing a red sash and a red headdress. The woman is wearing white fur ornaments on her elbows. The humanoids are performing a ritual as the sun sets. Split and Linden observe the humanoids from their ship. They notice that the humanoids are being attacked by a group of warriors. The warriors are wielding crude weapons and swinging them at the humanoids. Linden and Split are worried that they will not be able to warn the humanoids in time. The humanoids rally around their leader, who is wearing the same red sash and red headdress as the man. The humanoid leader marches over to Linden and Split's ship. Gravgak, the guard of the humanoid leader, confronts Linden and Split. He is worried that the sponge trees, which are camouflaging the warriors, will attack the humanoids again. He wants Linden and Split to repeat the "see-o-see-o" siren noises to keep the humanoids safe. When Linden and Split repeat the siren, the warriors run back into their hiding places. Tomboldo is the leader of the humanoids, and he is wearing the red sash and headdress. Linden gives Tomboldo a gift, a musical medallion that plays a tune when it is touched. Tomboldo accepts the gift, and Linden and Split learn that Tomboldo's name is "Tomboldo." They are invited to accompany Tomboldo back to his cave, where they will be safe from the warriors.

Coarse Plans, E2E

1. Captain Linden and Split Campbell land on a strange planet.
2. They observe a man-like creature on the planet.
3. A group of natives gather for a sunset ritual.
4. The natives are attacked by savage warriors.
5. Campbell and Linden intervene to save the natives.
6. The natives invite Campbell and Linden to their city.
7. Campbell is injured during the attack.
8. Vauna, the leader's daughter, nurses Campbell back to health.
9. Campbell learns to speak Benzendella.
10. The natives plan to move back to another part of the world.

Captain Linden and his assistant, Split Campbell, land their ship on a strange planet in order to learn more about it. As they observe the planet from their ship, they see a long, silvery, serpent-like object crawling across the planet's surface. Linden and Campbell are the first humans to land on the planet, and they have been sent on this mission by the EGGWE (Earth Galaxy Goodwill Expeditions). They have been given two pieces of information about the planet: that it is inhabited by humanoid creatures, and that there is a long, cylindrical "rope" that crawls across the surface of the planet. Linden orders Campbell to take a closer look at the "rope" while the sun is still shining on it. Campbell does so, and he reports back to Linden that the object is a living creature that is upright and wearing clothes. Linden is excited to see a human-like creature on another planet, and he orders Campbell to get ready to meet the creature. Linden and Campbell observe a group of natives who are gathering in a clearing to watch the sunset. The leader of the group is wearing a red sash and a red headdress, and the other members of the group are all handsome and muscular. They are all wearing white fur ornaments on their elbows, which Linden and Campbell infer are for protection. As they watch the natives, Linden and Campbell notice that the trees around them are moving, and they realize that the natives are not aware that they are being watched by humans from another world. Suddenly, a group of savage warriors appear and attack the natives. The natives try to defend themselves, but they are no match for the warriors, who are armed with crude clubs and whips. The natives rally around their leader, and Linden and Campbell decide to intervene to save them. They use a siren on their ship to scare off the warriors, and they descend from the ship to join the natives in the clearing. The natives invite Linden and Campbell to accompany them back to their city, where they can learn more about the planet and the "serpent river" that the natives worship. They are escorted by Gravgak, a tall, muscular guard who is responsible for keeping an eye on the moving trees. During the journey back to the city, Campbell is injured when Gravgak accidentally hits him with a club. Linden uses a capsule bomb to stop the attackers, but Gravgak is killed in the process. The natives nurse Campbell back to health.

Figure 5.5: Sample outputs from the No Planning baseline and Coarse Plan E2E models on SQuALITY. Red text indicates factual errors in the output.

Blueprint QA, Multi-Task

Q: What is the name of the planet that Campbell and Linden land on? A: Serpent River
Q: Who is Campbell and Linden's sponsor? A: EGGWE

Captain Linden and his crew, Split and Campbell, are on a mission to explore a new planet. They land on the planet and see a long serpent-like thing crawling over the planet's surface. They are sent to learn the answers to what it is. They land at a safe distance from it, and Split records all observable phenomena. They see a man and a woman walking upright and wearing clothing. The man is the leader of the group, and the woman is the leader's daughter, Vauna. Campbell and Linden go out to meet them, and they see a crowd of people gathering around the man and the woman. They notice that the trees are moving, and they realize that the people are being attacked by a group of warriors. The people rally around their leader, and Campbell and Linden use their ship's siren to scare off the attackers. The leader invites Campbell and Linden to accompany them back to their cave, where they can learn more about the Serpent River and the people's relationship with it. Linden and Campbell meet Gravgak, the leader's guard, and learn more about the people and the Serpent River. Linden and Vauna fall in love, but Gravgak tries to kill Linden. Linden wakes up from a long sleep, and Gravgak apologizes to him. Tomboldo tells Linden that soon they will move back to another part of the world, and Linden will be well enough to travel with them.

Coarse Plans + QA, Multi-Task

1. Captain Linden and Split Campbell land on an unknown planet.
2. They observe a mysterious rope-like structure on the planet.
   Q: What is the name of the mysterious rope? A: Serpent River
3. They spot two humanoid beings on the planet.
4. The beings are performing a sunset ritual.
5. A group of savage warriors attack the beings.
6. Campbell and Linden use a siren to scare off the attackers.
7. The beings invite the explorers to their city.
8. The explorers learn about the Serpent River and its significance.
9. Linden falls unconscious and wakes up with Vauna.
10. The beings plan to move back to another part of the world.

Captain Linden and his assistant, Split, are on a mission to explore a new planet. The planet is inhabited by a humanoid race that resembles the humans of Earth. They land on the planet and see a long serpent-like object crawling over the planet's surface. The object is moving slowly and has been moving for centuries. They land at a safe distance away from the object. Split is observing the object through a telescope while Linden is writing a report. Linden asks Split to look at the object through the telescope. Split confirms that the object is moving. Linden is excited to see a human-like creature on the planet. He orders Split to get ready to meet the creature. They observe the creature from a distance and see that it is a man and a woman. They also see a group of people gathering in a clearing. Linden notices that the people are wearing fur ornaments on their elbows, which he assumes are for protection. He also notices that the trees are moving towards the people. Linden realizes that the trees are camouflaging for a group of warriors. The warriors charge towards the people with crude weapons. The people rally around their leader. Linden and Split decide to descend from the ship to help the people. They meet the leader of the people, Tombold, and offer him a gift. Tombold accepts the gift and introduces Linden and Split to the rest of his group. They learn that Gravgak is the guard of Tombold's group and that they are going to take them back to their cave. As they walk back to the cave, Linden and Split realize that they are being attacked by the warriors. They throw fire bombs at the warriors, injuring Gravgak in the process. Tombold apologizes to Linden for Gravgak's injury, but Gravgak refuses to believe that the attack was accidental. Tombold tells Linden that they will move back to another part of the world soon.

Figure 5.6: Sample outputs from the Blueprint QA, Multi-Task and Coarse Plan + QA, Multi-Task models on SQuALITY. Red text indicates factual errors in the output.

5.7 Conclusion

In this chapter, we explore a different approach to building a discourse model, for the summarization task. We formulate the discourse model as a "plan" for generating summaries, and investigate several plan formulations for narrative summarization. We explore lower-level plans with targeted QA pairs, higher-level plans that reflect the narrative structure of the text, and a combination of both. Our high-level planning method involves prompting an LLM to produce training plans, followed by fine-tuning an SLM. Our results show that, contrary to prior works, planning methods do not offer a considerable improvement in summary quality or faithfulness. In our manual evaluation, we find that while E2E methods are tightly grounded to summaries, all settings hallucinate at similar rates, including in the generated plans. Replacing E2E plans with high-quality Sonnet 3.5 plans improves summary quality and partially mitigates faithfulness issues. Our work illustrates the difficulty of plan-guided summarization in narrative text with SLMs.

Chapter 6

Conclusion

6.1 Summary of Findings

In this thesis, we are interested in how we can model discourse in order to improve language processing in coreference resolution and summarization tasks. We argue that maintaining a discourse model can improve computational efficiency, maintain high accuracy, and in summarization, increase summary quality and faithfulness to the source text.

In Chapter 3, we examine encoder-based coreference resolution systems. Many existing works involve computing scores for every span in the document, incurring a high computation cost. We propose a shift-reduce framework which runs in linear complexity with respect to the tokens in the document. The model incrementally builds a memory representation of the entities encountered so far, representing entities with the average of their mentions' hidden representations. In order to fairly assess against other systems, we restrict compared systems to also process text incrementally. We show that on OntoNotes (Weischedel et al., 2013; Pradhan et al., 2012), our incremental system performs strongly against other incremental ones. The performance advantage improves further on the CODI-CRAC 2021 corpus (Khosla et al., 2021), perhaps reflecting that our incremental formulation is more suitable for inherently incremental tasks such as dialogue interaction. If we relax the incrementality restriction in the base encoder, our system achieves comparable results to state-of-the-art methods. Analyzing the performance difference between the sentence-incremental model and part-incremental model reveals a heavy dependence on the caching mechanism of the base model, XLNet. The performance difference between our sentence-incremental model and state-of-the-art methods can be explained by XLNet's inability to effectively use its caching

mechanism in an incremental setup.

In Chapter 4, we explore newer seq2seq approaches to coreference resolution systems (Bohnet et al., 2023; Zhang et al., 2023). Noting that existing works either do not handle incremental processing (Zhang et al., 2023), or re-input the entire preceding text for each new sentence (Bohnet et al., 2023), we instead propose a Model-based Incremental system which incrementally builds up the discourse model as a list of entities ordered by mention recency. While we find the Model-based Incremental system scores very similarly to a Full-Prefix Incremental baseline, both systems suffer a performance gap on OntoNotes compared to non-incremental systems. Significantly, on LitBank (Bamman et al., 2020), which includes singleton mentions, we do not observe any performance drop, and instead the Model-based system improves over state-of-the-art methods. We perform an in-depth analysis between the non-incremental baseline and Full-Prefix Incremental system, finding that the incremental system struggles with straightforward proper names and definite noun phrases, even if they have an exact or partial string match to their antecedent. We hypothesize that this result is caused by the lack of singleton annotation in OntoNotes, particularly as the performance drop is not present on LitBank. Lastly, we explore adding NER labels to OntoNotes as a proxy for singletons, but find that span mismatches between NER and coreference annotation layers, as well as annotation errors, hinder this method from further improvement on OntoNotes.

In Chapter 5, we switch tasks and investigate methods for automatic summarization. We investigate whether plan-guided summarization in SLMs can reduce hallucinations in long, narrative-based text. We investigate several plan formulations, include existing QA-based plans, abstractive sub-event plans, and a mixture of both. Surprisingly, we find that plans do not improve faithfulness or summary quality according to automatic metrics, despite prior work showing plans can improve summary faithfulness. We conduct a human evaluation, finding that both plan-based and baseline summarization methods hallucinate on narrative texts, including in their plans. However, we find that one training method, the E2E method (Narayan et al., 2023), produces summaries that are well-grounded to their plans. We perform an oracle experiment, where the SLM-generated plan is replaced with a much higher-quality LLM-based plan. We find the resulting summaries are much more faithful to the source texts, and also improve on coverage and conciseness scores.

Overall, in this thesis we argue that discourse modelling is an effective strategy for improving language processing in tasks such as coreference resolution and summariza-

tion. In Chapters 3 and 4, discourse modelling improve computational efficiency while maintaining accuracy. However, we find that incremental settings are more challenging than non-incremental ones, and in both chapters, we see performance gaps between incremental and non-incremental processing. In Chapter 3, this gap is primarily caused by deficiencies in the encoder's caching mechanism, while in Chapter 4, dataset annotation artifacts (i.e. singletons) affect the model's performance. On the other hand, in Chapter 5 we find that abstractive discourse modelling, i.e. the narrative-based plans, is much more challenging. Here, although we find SLMs are not able to benefit from the narrative-based plans, substituting in high-quality, LLM-based plans results in better faithfulness and summary quality.

6.2 Future Work

6.2.1 Extending the Discourse Model

In Chapters 3 and 4, we represent the discourse model using entities in the text, which in turn are represented as an average of their mentions' hidden representations in Chapter 3 and a concatenation of their mentions in Chapter 4. This is a rough approximation of the discourse model proposed in File Change Semantics and Discourse Representation Theory. However, the theories involve storing more information about entities than we implement in this thesis; for example, their state, attributes and actions, the relationships between entities, etc.

In Chapter 4, we find an explicit case where the loss of context causes the Model-based Incremental model to miss linking two mentions, namely the deictic "*we*" with its speaker, "*CNN*". In this case, adding speaker tags in the OntoNotes annotation layer should be straightforward to implement. However, in other cases, discarding tokens relevant to the context could be more harmful. For example, consider the following copular expression from OntoNotes:

My mother was Thelma Wahl.

All coreference datasets we use in this thesis only annotate the left side of copular expressions as mentions. In this case, the system would add "*My mother*" to the discourse model, and discard her name "*Thelma Wahl*". But clearly *Thelma Wahl*'s name is an important attribute to consider and should belong to the discourse model. However, determining exactly which attributes should be extracted from the text and how is not clear.

Furthermore, constructing complex discourse models may introduce unwanted noise. In Chapter 5, the discourse model is formulated as an abstractive plan composed of sub-events. The complicated and abstractive nature of this formulation introduces hallucinations into the plan, as there is no guarantee that the generated plan is faithful to the source text. Any attempt at enriching the discourse model should be wary of the risks of adding noise instead of positive signal.

Lastly, extending our work on discourse modelling may not only involve adding more information, but also selectively filtering out unnecessary information. McCoy and Strube (1999) find that writers often use definite NPs where a pronoun would be allowed, since the context was unambiguous in what the pronoun was referring to. They show that writers use definite NPs to "refresh" a salient entity, implying that pronouns do not keep the entity salient in the discourse model. This may imply that mentions with different syntactic categories should be handled differently in the discourse model, with the understanding that pronouns are semantically "empty", apart from grammatical gender. For example, it may be more suitable to omit pronouns from the discourse model, or treat proper names with more weight.

6.2.2 Bridging and Split-Antecedent Reference

In this thesis, we do not address other types of anaphora such as bridging (Clark, 1975; Asher and Lascarides, 1998; Poesio et al., 2004a) or split-antecedent resolution (Eschenbach et al., 1989; Ingria and Stallard, 1989; Kamp and Reyle, 1993). It is interesting to consider how our model may be extended to accommodate such phenomena. To give an example, consider the following text:

We had a picnic. The beer was warm.

After encountering "*a picnic*", we would like to not only update the discourse model with *a picnic*, but also recognize that *picnic*-related concepts such as *the beer* and *the food* may emerge in future utterances. Adding all the related concepts to the discourse model may be infeasible; instead we envision a separate module able to recognize related concepts given "*a picnic*". It should then predict the bridging relation.

Similarly, for split-antecedent reference, when the system encounters a plural referring expression, such as *Tom, Jerry, Sally and Jane*, we would allow it to recognize possible related groupings, such as *the boys* or *the girls*. However, determining which groupings are relevant or plausible may not be straightforward. A mention such as "*the green cup, the red cup and the blue cup*" may not contain any relevant groups.

While both phenomena may seem highly specific, bridging and split-antecedent references occur regularly among speakers of any language. Both tasks are central to many NLU applications, such as dialogue interpretation.

Bibliography

Abdin, M., Aneja, J., Awadalla, H., Awadallah, A., Awan, A. A., Bach, N., Bahree, A., Bakhtiari, A., Bao, J., Behl, H., Benhaim, A., Bilenko, M., Bjorck, J., Bubeck, S., Cai, M., Cai, Q., Chaudhary, V., Chen, D., Chen, D., Chen, W., Chen, Y.-C., Chen, Y.-L., Cheng, H., Chopra, P., Dai, X., Dixon, M., Eldan, R., Fragoso, V., Gao, J., Gao, M., Gao, M., Garg, A., Giorno, A. D., Goswami, A., Gunasekar, S., Haider, E., Hao, J., Hewett, R. J., Hu, W., Huynh, J., Iter, D., Jacobs, S. A., Javaheripi, M., Jin, X., Karampatziakis, N., Kauffmann, P., Khademi, M., Kim, D., Kim, Y. J., Kurilenko, L., Lee, J. R., Lee, Y. T., Li, Y., Li, Y., Liang, C., Liden, L., Lin, X., Lin, Z., Liu, C., Liu, L., Liu, M., Liu, W., Liu, X., Luo, C., Madan, P., Mahmoudzadeh, A., Majercak, D., Mazzola, M., Mendes, C. C. T., Mitra, A., Modi, H., Nguyen, A., Norick, B., Patra, B., Perez-Becker, D., Portet, T., Pryzant, R., Qin, H., Radmilac, M., Ren, L., de Rosa, G., Rosset, C., Roy, S., Ruwase, O., Saarikivi, O., Saied, A., Salim, A., Santacroce, M., Shah, S., Shang, N., Sharma, H., Shen, Y., Shukla, S., Song, X., Tanaka, M., Tupini, A., Vaddamanu, P., Wang, C., Wang, G., Wang, L., Wang, S., Wang, X., Wang, Y., Ward, R., Wen, W., Witte, P., Wu, H., Wu, X., Wyatt, M., Xiao, B., Xu, C., Xu, J., Xu, W., Xue, J., Yadav, S., Yang, F., Yang, J., Yang, Y., Yang, Z., Yu, D., Yuan, L., Zhang, C., Zhang, C., Zhang, J., Zhang, L. L., Zhang, Y., Zhang, Y., Zhang, Y., and Zhou, X. (2024). Phi-3 technical report: A highly capable language model locally on your phone.

Adams, G., Fabbri, A., Ladhak, F., Elhadad, N., and McKeown, K. (2023). Generating EDU extracts for plan-guided summary re-ranking. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2680–2697, Toronto, Canada. Association for Computational Linguistics.

Ahmed, S. R., Nath, A., Martin, J. H., and Krishnaswamy, N. (2023). $2 * n$ is better than n^2: Decomposing event coreference resolution into two tractable problems. In

Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, Findings of the Association for Computational Linguistics: ACL 2023, pages 1569–1583, Toronto, Canada. Association for Computational Linguistics.

Alberti, C., Andor, D., Pitler, E., Devlin, J., and Collins, M. (2019a). Synthetic QA corpora generation with roundtrip consistency. In Korhonen, A., Traum, D., and Màrquez, L., editors, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.

Alberti, C., Lee, K., and Collins, M. (2019b). A bert baseline for the natural questions.

Allaway, E., Wang, S., and Ballesteros, M. (2021). Sequential cross-document coreference resolution. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4659–4671, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Altmann, G. and Steedman, M. (1988). Interaction with context during human sentence processing. Cognition, 30(3):191–238.

Andreas, J., Bufe, J., Burkett, D., Chen, C., Clausman, J., Crawford, J., Crim, K., DeLoach, J., Dorner, L., Eisner, J., Fang, H., Guo, A., Hall, D., Hayes, K., Hill, K., Ho, D., Iwaszuk, W., Jha, S., Klein, D., Krishnamurthy, J., Lanman, T., Liang, P., Lin, C. H., Lintsbakh, I., McGovern, A., Nisnevich, A., Pauls, A., Petters, D., Read, B., Roth, D., Roy, S., Rusak, J., Short, B., Slomin, D., Snyder, B., Striplin, S., Su, Y., Tellman, Z., Thomson, S., Vorobev, A., Witoszko, I., Wolfe, J., Wray, A., Zhang, Y., and Zotov, A. (2020). Task-oriented dialogue as dataflow synthesis. Transactions of the Association for Computational Linguistics, 8:556–571.

Anthropic (2024). Claude 3.5 sonnet model card addendum. Accessed on October 3, 2024.

Asher, N. and Lascarides, A. (1998). Bridging. Journal of Semantics, 15(1):83–113.

Asher, N. and Lascarides, A. (2003). Logics of Conversation. Cambridge University Press, Cambridge.

Bagga, A. and Baldwin, B. (1998). Algorithms for scoring coreference chains. In The first international conference on language resources and evaluation workshop on linguistics coreference, volume 1, pages 563–566. Citeseer.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In Bengio, Y. and LeCun, Y., editors, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Bamman, D., Lewke, O., and Mansoor, A. (2020). An annotated dataset of coreference in English literature. In Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, Proceedings of the Twelfth Language Resources and Evaluation Conference, pages 44–54, Marseille, France. European Language Resources Association.

Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer. arXiv:2004.05150.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2):157–166.

Berg-Kirkpatrick, T., Burkett, D., and Klein, D. (2012). An empirical investigation of statistical significance in NLP. In Tsujii, J., Henderson, J., and Paşca, M., editors, Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 995–1005, Jeju Island, Korea. Association for Computational Linguistics.

Bhandari, M., Gour, P. N., Ashfaq, A., Liu, P., and Neubig, G. (2020). Re-evaluating evaluation in text summarization. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 9347–9359, Online. Association for Computational Linguistics.

Bohnet, B., Alberti, C., and Collins, M. (2023). Coreference resolution through a seq2seq transition-based system. Transactions of the Association for Computational Linguistics, 11:212–226.

Bosch, P. (1983). Agreement and Anaphora. Academic Press, New York.

Brennan, S. E., Friedman, M. W., and Pollard, C. J. (1987). A centering approach to pronouns. In 25th Annual Meeting of the Association for Computational Linguistics, pages 155–162, Stanford, California, USA. Association for Computational Linguistics.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, Advances in Neural Information Processing Systems, volume 33, pages 1877–1901. Curran Associates, Inc.

Carletta, J. (2006). Announcing the ami meeting corpus. In The ELRA Newsletter 11(1), pages 3 – 5.

Chai, H. and Strube, M. (2022). Incorporating centering theory into neural coreference resolution. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V., editors, Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2996–3002, Seattle, United States. Association for Computational Linguistics.

Chang, Y., Lo, K., Goyal, T., and Iyyer, M. (2024). Booookscore: A systematic exploration of book-length summarization in the era of LLMs. In The Twelfth International Conference on Learning Representations.

Chawla, K., Rashkin, H., Tomar, G. S., and Reitter, D. (2024). Investigating content planning for navigating trade-offs in knowledge-grounded dialogue. In Graham, Y. and Purver, M., editors, Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2316–2335, St. Julian's, Malta. Association for Computational Linguistics.

Chen, A., Stanovsky, G., Singh, S., and Gardner, M. (2020). MOCHA: A dataset for training and evaluating generative reading comprehension metrics. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6521–6532, Online. Association for Computational Linguistics.

Chen, M., Chu, Z., Wiseman, S., and Gimpel, K. (2022). SummScreen: A dataset for abstractive screenplay summarization. In Muresan, S., Nakov, P., and Villavicencio, A., editors, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 8602–8615, Dublin, Ireland. Association for Computational Linguistics.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Moschitti, A., Pang, B., and Daelemans, W., editors, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Chowdhury, S. B. R., Monath, N., Dubey, K. A., Zaheer, M., McCallum, A., Ahmed, A., and Chaturvedi, S. (2024). Incremental extractive opinion summarization using cover trees. Transactions on Machine Learning Research.

Christiansen, M. H. and Chater, N. (2016). The now-or-never bottleneck: A fundamental constraint on language. Behavioral and Brain Sciences, 39:e62.

Chun, J. and Xue, N. (2024). Uniform meaning representation parsing as a pipelined approach. In Ustalov, D., Gao, Y., Panchenko, A., Tutubalina, E., Nikishina, I., Ramesh, A., Sakhovskiy, A., Usbeck, R., Penn, G., and Valentino, M., editors, Proceedings of TextGraphs-17: Graph-based Methods for Natural Language Processing, pages 40–52, Bangkok, Thailand. Association for Computational Linguistics.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. (2019). BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Burstein, J., Doran, C., and Solorio, T., editors, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Clark, H. H. (1975). Bridging. In Nash-Webber, B. and Schank, R., editors, Theoretical Issues in Natural Language Processing.

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. In International Conference on Learning Representations.

Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., and Goharian, N. (2018). A discourse-aware attention model for abstractive summarization of long documents. In Walker, M., Ji, H., and Stent, A., editors, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., Marris, L., Petulla, S., Gaffney, C., Aharoni, A., Lintz, N., Pais, T. C., Jacobsson, H., Szpektor, I., Jiang, N.-J., Haridasan, K., Omran, A., Saunshi, N., Bahri, D., Mishra, G., Chu, E., Boyd, T., Hekman, B., Parisi, A., Zhang, C., Kawintiranon, K., Bedrax-Weiss, T., Wang, O., Xu, Y., Purkiss, O., Mendlovic, U., Deutel, I., Nguyen, N., Langley, A., Korn, F., Rossazza, L., Ramé, A., Waghmare, S., Miller, H., Byrd, N., Sheshan, A., Hadsell, R., Bhardwaj, S., Janus, P., Rissa, T., Horgan, D., Abdagic, A., Belenki, L., Allingham, J., Singh, A., Guidroz, T., Srinivasan, S., Schmit, H., Chiafullo, K., Elisseeff, A., Jha, N., Kolhar, P., Berrada, L., Ding, F., Si, X., Mallick, S. B., Och, F., Erell, S., Ni, E., Latkar, T., Yang, S., Sirkovic, P., Feng, Z., Leland, R., Hornung, R., Wu, G., Blundell, C., Alvari, H., Huang, P.-S., Yip, C., Deur, S., Liu, L., Surita, G., Duque, P., Damen, D., Jia, J., Guez, A., Mircea, M., Sinha, A., Magni, A., Stradomski, P., Marian, T., Galić, V., Chen, W., Husain, H., Singhal, A., Grewe, D., Aubet, F.-X., Song, S., Blanco, L., Rechis, L., Ho, L., Munoz, R., Zheng, K., Hamrick, J., Mather, K., Taitelbaum, H., Rutherford, E., Lei, Y., Chen, K., Shukla, A., Moreira, E., Doi, E., Isik, B., Shabat, N., Rogozińska, D., Kolipaka, K., Chang, J., Vušak, E., Venkatachary, S., Noghabi, S., Bharti, T., Jun, Y., Zaks, A., Green, S., Challagundla, J., Wong, W., Mohammad, M., Hirsch, D., Cheng, Y., Naim, I., Proleev, L., Vincent, D., Singh, A., Krikun, M., Krishnan, D., Ghahramani, Z., Atias, A., Aggarwal, R., Kirov, C., Vytiniotis, D., Koh, C., Chronopoulou, A., Dogra, P., Ion, V.-D., Tyen, G., Lee, J., Weissenberger, F., Strohman, T., Balakrishna, A., Rae, J., Velic, M., de Liedekerke, R., Elyada, O., Yuan, W., Liu, C., Shani, L., Kishchenko, S., Alessio, B., Li, Y., Song, R., Kwei, S., Jankowski, O., Pappu, A., Namiki, Y., Ma, Y., Tripuraneni, N., Cherry, C., Ikonomidis, M., Ling, Y.-C., Ji, C., Westberg, B., Wright, A., Yu, D., Parkinson, D., Ramaswamy, S., Connor, J., Yeganeh, S. H., Grover, S., Kenwright, G., Litchev, L., Apps, C., Tomala, A., Halim, F., Castro-Ros, A., Li, Z., Boral, A., Sho, P., Yarom, M., Malmi, E., Klinghoffer, D., Lin, R., Ansell, A., S, P. K., Zhao,

S., Zuo, S., Santoro, A., Cheng, H.-T., Demmessie, S., Liu, Y., Brichtova, N., Culp, A., Braun, N., Graur, D., Ng, W., Mehta, N., Phillips, A., Sundberg, P., Godbole, V., Liu, F., Katariya, Y., Rim, D., Seyedhosseini, M., Ammirati, S., Valfridsson, J., Malihi, M., Knight, T., Toor, A., Lampe, T., Ittycheriah, A., Chiang, L., Yeung, C., Fréchette, A., Rao, J., Wang, H., Srivastava, H., Zhang, R., Rhodes, R., Brand, A., Weesner, D., Figotin, I., Gimeno, F., Fellinger, R., Marcenac, P., Leal, J., Marcus, E., Cotruta, V., Cabrera, R., Luo, S., Garrette, D., Axelrod, V., Baltateanu, S., Barker, D., Chen, D., Toma, H., Ingram, B., Riesa, J., Kulkarni, C., Zhang, Y., Liu, H., Wang, C., Polacek, M., Wu, W., Hui, K., Reyes, A. N., Su, Y., Barnes, M., Malhi, I., Siddiqui, A., Feng, Q., Damaschin, M., Pighin, D., Steiner, A., Yang, S., Boppana, R. S., Ivanov, S., Kandoor, A., Shah, A., Mujika, A., Huang, D., Choquette-Choo, C. A., Patel, M., Yu, T., Creswell, T., Jerry, Liu, Barros, C., Razeghi, Y., Roy, A., Culliton, P., Xiong, B., Pan, J., Strohmann, T., Powell, T., Seal, B., DeCarlo, D., Shyam, P., Katircioglu, K., Wang, X., Hardin, C., Odisho, I., Broder, J., Chang, O., Nair, A., Shtefan, A., O'Brien, M., Agarwal, M., Potluri, S., Goyal, S., Jhindal, A., Thakur, S., Stuken, Y., Lyon, J., Toutanova, K., Feng, F., Wu, A., Horn, B., Wang, A., Cullum, A., Taubman, G., Shrivastava, D., Shi, C., Tomlinson, H., Patel, R., Tu, T., Oflazer, A. M., Pongetti, F., Yang, M., Taïga, A. A., Perot, V., Pierse, N. W., Han, F., Drori, Y., Iturrate, I., Chakrabarti, A., Yeung, L., Dopson, D., ting Chen, Y., Kulshreshtha, A., Guo, T., Pham, P., Schuster, T., Chen, J., Polozov, A., Xing, J., Zhou, H., Kacham, P., Kukliansky, D., Miech, A., Yaroshenko, S., Chi, E., Douglas, S., Fei, H., Blondel, M., Myla, P., Madmoni, L., Wu, X., Keysers, D., Kjems, K., Albuquerque, I., Yu, L., D'sa, J., Plantan, M., Ionescu, V., Elias, J. S., Gupta, A., Vuyyuru, M. R., Alcober, F., Zhou, T., Ji, K., Hartmann, F., Puttagunta, S., Song, H., Amid, E., Stefanoiu, A., Lee, A., Pucciarelli, P., Wang, E., Raul, A., Petrov, S., Tian, I., Anklin, V., Nti, N., Gomes, V., Schumacher, M., Vesom, G., Panagopoulos, A., Bousmalis, K., Andor, D., Jacob, J., Zhang, Y., Rosgen, B., Kecman, M., Tung, M., Belias, A., Goodman, N., Covington, P., Wieder, B., Saxena, N., Davoodi, E., Huang, M., Maddineni, S., Roulet, V., Campbell-Ajala, F., Sessa, P. G., Xintian, Wu, Lai, G., Collins, P., Haig, A., Sakenas, V., Xu, X., Giustina, M., Shafey, L. E., Charoenpanit, P., Garg, S., Ainslie, J., Severson, B., Arenas, M. G., Pathak, S., Rajayogam, S., Feng, J., Bakker, M., Li, S., Wichers, N., Rogers, J., Geng, X., Li, Y., Jagerman, R., Jia, C., Olmert, N., Sharon, D., Mauger, M., Mariserla, S., Ma, H., Mohabey, M., Kim, K., Andreev, A., Pollom, S., Love, J., Jain, V., Agrawal, P., Schroecker, Y., Fortin, A., Warmuth, M., Liu, J., Leach, A., Blok, I., Girirajan, G. P., Aharoni, R., Uria, B.,

Sozanschi, A., Goldberg, D., Ionita, L., Ribeiro, M. T., Zlocha, M., Birodkar, V., Lachgar, S., Yuan, L., Choudhury, H., Ginsberg, M., Zheng, F., Dibb, G., Graves, E., Lokhande, S., Rasskin, G., Muraru, G.-C., Quick, C., Tata, S., Sermanet, P., Chawla, A., Karo, I., Wang, Y., Zhang, S., Keller, O., Dragan, A., Su, G., Chou, I., Liu, X., Tao, Y., Prabhakara, S., Wilson, M., Liu, R., Wang, S., Evans, G., Du, D., Castaño, A., Prasad, G., Mahdy, M. E., Gerlach, S., Reid, M., Kahn, J., Zait, A., Pillai, T. S., Ulrich, T., Wang, G., Wassenberg, J., Farkash, E., Yalasangi, K., Wang, C., Bauza, M., Bucher, S., Liu, T., Yan, J., Leung, G., Sindhwani, V., Barnes, P., Singh, A., Jurin, I., Chang, J., Bhumihar, N. K., Eiger, S., Citovsky, G., Withbroe, B., Li, Z., Xue, S., Santo, N. D., Stoyanov, G., Raimond, Y., Zheng, S., Gao, Y., Listík, V., Kwasiborski, S., Saputro, R., Ozturel, A., Mallya, G., Majmundar, K., West, R., Caron, P., Wei, J., Castrejon, L., Vikram, S., Ramachandran, D., Dhawan, N., Park, J., Smoot, S., van den Driessche, G., Blau, Y., Malik, C., Liang, W., Hirsch, R., dos Santos, C. N., Weinstein, E., van den Oord, A., Lall, S., FitzGerald, N., Jiang, Z., Yang, X., Webster, D., Elqursh, A., Pope, A., Rotival, G., Raposo, D., Zhu, W., Dean, J., Alabed, S., Tran, D., Gupta, A., Gleicher, Z., Austin, J., Rosseel, E., Umekar, M., Das, D., Sun, Y., Chen, K., Misiunas, K., Zhou, X., Di, Y., Loo, A., Newlan, J., Li, B., Ramasesh, V., Xu, Y., Chen, A., Gandhe, S., Soricut, R., Gupta, N., Hu, S., El-Sayed, S., Garcia, X., Brusilovsky, I., Chen, P.-C., Bolt, A., Huang, L., Gurney, A., Zhang, Z., Pritzel, A., Wilkiewicz, J., Seybold, B., Shamanna, B. K., Fischer, F., Dean, J., Gill, K., Mcilroy, R., Bhowmick, A., Selier, J., Yang, A., Cheng, D., Magay, V., Tan, J., Varma, D., Walder, C., Kocisky, T., Nakashima, R., Natsev, P., Kwong, M., Gog, I., Zhang, C., Dieleman, S., Jimma, T., Ryabtsev, A., Brahma, S., Steiner, D., Du, D., Žužul, A., Žanić, M., Raghavachari, M., Gierke, W., Zheng, Z., Petrova, D., Dauphin, Y., Liu, Y., Kessler, I., Hand, S., Duvarney, C., Kim, S., Lee, H., Hussenot, L., Hui, J., Smith, J., Jain, D., Xia, J., Tomar, G. S., Amiri, K., Phan, D., Fuchs, F., Weyand, T., Tomasev, N., Cordell, A., Liu, X., Mallinson, J., Joshi, P., Crawford, A., Suggala, A., Chien, S., Fernando, N., Sanchez-Vargas, M., Williams, D., Crone, P., Luo, X., Karpov, I., Shan, J., Thurk, T., Strudel, R., Voigtlaender, P., Patil, P., Dozat, T., Khodaei, A., Singla, S., Ambroszczyk, P., Wu, Q., Chang, Y., Roark, B., Hegde, C., Ding, T., Filos, A., Wu, Z., Pinto, A. S., Liu, S., Khanna, S., Pandey, A., Mcloughlin, S., Li, Q., Haves, S., Zhou, A., Buchatskaya, E., Leal, I., de Boursac, P., Akazawa, N., Anderson, N., Chen, T., Somandepalli, K., Liang, C., Goenka, S., Winkler, S., Grushetsky, A., Ding, Y., Smith, J., Ye, F., Pont-Tuset, J., Li, E., Li, R., Golany, T., Wegner, D., Jiang, T., Barak, O., Shangguan, Y., Vértes, E., Wong, R., Bornschein,

J., Tudor, A., Bevilacqua, M., Schaul, T., Rawat, A. S., Zhao, Y., Axiotis, K., Meng, L., McLean, C., Lai, J., Beattie, J., Kushman, N., Liu, Y., Kutzman, B., Lang, F., Ye, J., Netrapalli, P., Mishra, P., Khan, M., Goel, M., Willoughby, R., Tian, D., Zhuang, H., Chen, J., Tsai, Z., Kementsietsidis, T., Khare, A., Keeling, J., Xu, K., Waters, N., Altché, F., Popat, A., Mittal, B., Saxton, D., Badawy, D. E., Mathieu, M., Zheng, Z., Zhou, H., Ranka, N., Shin, R., Duan, Q., Salimans, T., Mihailescu, I., Shaham, U., Chang, M.-W., Assael, Y., Dikkala, N., Izzard, M., Cohen-Addad, V., Graves, C., Feinberg, V., Chung, G., Strouse, D., Karmon, D., Sharifzadeh, S., Ashwood, Z., Pham, K., Blanton, J., Vasiloff, A., Barber, J., Geller, M., Zhou, A., Zubach, F., Huang, T.-K., Zhang, L., Gupta, H., Young, M., Proskurnia, J., Votel, R., Gabeur, V., Barcik, G., Tripathi, A., Yu, H., Yan, G., Changpinyo, B., Pavetić, F., Coyle, A., Fujii, Y., Mendez, J. G., Zhou, T., Rajamani, H., Hechtman, B., Cao, E., Juan, D.-C., Tan, Y.-X., Dalibard, V., Du, Y., Clay, N., Yao, K., Jia, W., Vijaykumar, D., Zhou, Y., Bai, X., Hung, W.-C., Pecht, S., Todorov, G., Khadke, N., Gupta, P., Lahoti, P., Autef, A., Duddu, K., Lee-Thorp, J., Bykovsky, A., Misiunas, T., Flennerhag, S., Thangaraj, S., McGiffin, J., Nado, Z., Kunesch, M., Noever, A., Hertz, A., Liang, M., Stone, V., Palmer, E., Daruki, S., Pramanik, A., Põder, S., Kyker, A., Khan, M., Sluzhaev, E., Ritter, M., Ruderman, A., Zhou, W., Nagpal, C., Vodrahalli, K., Necula, G., Barham, P., Pavlick, E., Hartford, J., Shafran, I., Zhao, L., Mikuła, M., Eccles, T., Shimokawa, H., Garg, K., Vilnis, L., Chen, H., Shumailov, I., Lee, K.-H., Abdelhamed, A., Xie, M., Cohen, V., Hlavnova, E., Malkin, D., Sitawarin, C., Lottes, J., Coquinot, P., Yu, T., Kumar, S., Zhang, J., Mahendru, A., Ahmed, Z., Martens, J., Chen, T., Boag, A., Peng, D., Devin, C., Klimovskiy, A., Phuong, M., Vainstein, D., Xie, J., Ramabhadran, B., Howard, N., Yu, X., Goswami, G., Cui, J., Shleifer, S., Pinto, M., Yeh, C.-K., Yang, M.-H., Javanmardi, S., Ethier, D., Lee, C., Orbay, J., Kotecha, S., Bromberg, C., Shaw, P., Thornton, J., Rosenthal, A. G., Gu, S., Thomas, M., Gemp, I., Ayyar, A., Ushio, A., Selvan, A., Wee, J., Liu, C., Majzoubi, M., Yu, W., Abernethy, J., Liechty, T., Pan, R., Nguyen, H., Qiong, Hu, Perrin, S., Arora, A., Pitler, E., Wang, W., Shivakumar, K., Prost, F., Limonchik, B., Wang, J., Gao, Y., Cour, T., Buch, S., Gui, H., Ivanova, M., Neubeck, P., Chan, K., Kim, L., Chen, H., Goyal, N., Chung, D.-W., Liu, L., Su, Y., Petrushkina, A., Shen, J., Joulin, A., Xu, Y., Lin, S. X., Kulizhskaya, Y., Chelba, C., Vasudevan, S., Collins, E., Bashlovkina, V., Lu, T., Fritz, D., Park, J., Zhou, Y., Su, C., Tanburn, R., Sushkov, M., Rasquinha, M., Li, J., Prendki, J., Li, Y., LV, P., Sharma, S., Fitoussi, H., Huang, H., Dai, A., Dao, P., Burrows, M., Prior, H., Qin, D., Pundak, G., Sjoesund, L. L., Khurshudov,

A., Zhu, Z., Webson, A., Kemp, E., Tan, T., Agrawal, S., Sargsyan, S., Cheng, L., Stephan, J., Kwiatkowski, T., Reid, D., Byravan, A., Michaely, A. H., Heess, N., Zhou, L., Goenka, S., Carpenter, V., Levskaya, A., Wang, B., Roberts, R., Leblond, R., Chikkerur, S., Ginzburg, S., Chang, M., Riachi, R., Chuqiao, Xu, Borsos, Z., Pliskin, M., Pawar, J., Lustman, M., Kirkwood, H., Anand, A., Chaudhary, A., Kalb, N., Milan, K., Augenstein, S., Goldie, A., Prince, L., Raman, K., Sun, Y., Xia, V., Cohen, A., Huo, Z., Camp, J., Ellis, S., Zilka, L., Torres, D. V., Patel, L., Arora, S., Chan, B., Adler, J., Ayoub, K., Liang, J., Jamil, F., Jiang, J., Baumgartner, S., Sun, H., Karov, Y., Akulov, Y., Zheng, H., Cai, I., Fantacci, C., Rubin, J., Acha, A. R., Wang, M., D'Souza, N., Sathyanarayana, R., Dai, S., Rowe, S., Simanovsky, A., Goldman, O., Kuang, Y., Pan, X., Rosenberg, A., Rojas-Esponda, T., Dutta, P., Zeng, A., Jurenka, I., Farquhar, G., Bansal, Y., Iqbal, S., Roelofs, B., Joung, G.-Y., Beak, P., Ryu, C., Poplin, R., Wu, Y., Alayrac, J.-B., Buthpitiya, S., Ronneberger, O., Habtegebriel, C., Li, W., Cavallaro, P., Wei, A., Bensky, G., Denk, T., Ganapathy, H., Stanway, J., Joshi, P., Bertolini, F., Lo, J., Ma, O., Charles, Z., Sampemane, G., Sahni, H., Chen, X., Askham, H., Gaddy, D., Young, P., Tan, J., Eyal, M., Bražinskas, A., Zhong, L., Wu, Z., Epstein, M., Bailey, K., Hard, A., Lee, K., Goldshtein, S., Ruiz, A., Badawi, M., Lochbrunner, M., Kearns, J., Brown, A., Pardo, F., Weber, T., Yang, H., Jiang, P.-P., Akin, B., Fu, Z., Wainwright, M., Zou, C., Gaba, M., Manzagol, P.-A., Kan, W., Song, Y., Zainullina, K., Lin, R., Ko, J., Deshmukh, S., Jindal, A., Svensson, J., Tyam, D., Zhao, H., Kaeser-Chen, C., Baird, S., Moradi, P., Hall, J., Guo, Q., Tsang, V., Liang, B., Pereira, F., Ganesh, S., Korotkov, I., Adamek, J., Thiagarajan, S., Tran, V., Chen, C., Tar, C., Jain, S., Dasgupta, I., Bilal, T., Reitter, D., Zhao, K., Vezzani, G., Gehman, Y., Mehta, P., Beltrone, L., Dotiwalla, X., Guadarrama, S., Abbas, Z., Karp, S., Georgiev, P., Ferng, C.-S., Brockschmidt, M., Peng, L., Hirnschall, C., Verma, V., Bi, Y., Xiao, Y., Dabush, A., Xu, K., Wallis, P., Parker, R., Wang, Q., Xu, Y., Safarli, I., Tewari, D., Zhang, Y., Kim, S., Gesmundo, A., Thomas, M., Levi, S., Chowdhury, A., Rao, K., Garst, P., Conway-Rahman, S., Ran, H., McKinney, K., Xiao, Z., Yu, W., Agrawal, R., Stjerngren, A., Ionescu, C., Chen, J., Sharma, V., Chiu, J., Liu, F., Franko, K., Sanford, C., Cai, X., Michel, P., Ganapathy, S., Labanowski, J., Garrett, Z., Vargas, B., Sun, S., Gale, B., Buschmann, T., Desjardins, G., Ghelani, N., Jain, P., Verma, M., Asawaroengchai, C., Eisenschlos, J., Harlalka, J., Kazawa, H., Metzler, D., Howland, J., Jian, Y., Ades, J., Shah, V., Gangwani, T., Lee, S., Ring, R., Hernandez, S. M., Reich, D., Sinha, A., Sathe, A., Kovac, J., Gill, A., Kannan, A., D'olimpio, A., Sevenich, M., Whang, J., Kim, B., Sim, K. C., Chen, J., Zhang,

J., Lall, S., Matias, Y., Jia, B., Friesen, A., Nasso, S., Thapliyal, A., Perozzi, B., Yu, T., Shekhawat, A., Huda, S., Grabowski, P., Wang, E., Sreevatsa, A., Dib, H., Hassen, M., Schuh, P., Milutinovic, V., Welty, C., Quinn, M., Shah, A., Wang, B., Barth-Maron, G., Frye, J., Axelsson, N., Zhu, T., Ma, Y., Giannoumis, I., Sedghi, H., Ye, C., Luan, Y., Aydin, K., Chandra, B., Sampathkumar, V., Huang, R., Lavrenko, V., Eleryan, A., Hong, Z., Hansen, S., Carthy, S. M., Samanta, B., Ćevid, D., Wang, X., Li, F., Voznesensky, M., Hoffman, M., Terzis, A., Sehwag, V., Fidel, G., He, L., Cai, M., He, Y., Feng, A., Nikoltchev, M., Phatale, S., Chase, J., Lawton, R., Zhang, M., Ouyang, T., Tragut, M., Manshadi, M. H., Narayanan, A., Shen, J., Gao, X., Bolukbasi, T., Roy, N., Li, X., Golovin, D., Panait, L., Qin, Z., Han, G., Anthony, T., Kudugunta, S., Patraucean, V., Ray, A., Chen, X., Yang, X., Bhatia, T., Talluri, P., Morris, A., Ražnatović, A., Brownfield, B., An, J., Peng, S., Kane, P., Zheng, C., Duduta, N., Kessinger, J., Noraky, J., Liu, S., Rong, K., Veličković, P., Rush, K., Goldin, A., Wei, F., Garlapati, S. M. R., Pantofaru, C., Kwon, O., Ni, J., Noland, E., Trapani, J. D., Beaufays, F., Roy, A. G., Chow, Y., Turker, A., Cideron, G., Mei, L., Clark, J., Dou, Q., Bošnjak, M., Leith, R., Du, Y., Yazdanbakhsh, A., Nasr, M., Kwak, C., Sheth, S. S., Kaskasoli, A., Anand, A., Lakshminarayanan, B., Jerome, S., Bieber, D., Chu, C.-T., Senges, A., Shen, T., Sridhar, M., Ndebele, N., Beyret, B., Mohamed, S., Chen, M., Freitag, M., Guo, J., Liu, L., Roit, P., Chen, H., Yan, S., Stone, T., Co-Reyes, J., Cole, J., Scellato, S., Azizi, S., Hashemi, H., Jin, A., Iyer, A., Valentine, M., György, A., Ahuja, A., Diaz, D. H., Lee, C.-Y., Clement, N., Kong, W., Garmon, D., Watts, I., Bhatia, K., Gupta, K., Miecnikowski, M., Vallet, H., Taly, A., Loper, E., Joshi, S., Atwood, J., Chick, J., Collier, M., Iliopoulos, F., Trostle, R., Gunel, B., Leal-Cavazos, R., Hrafnkelsson, A. M., Guzman, M., Ju, X., Forbes, A., Emond, J., Chauhan, K., Caine, B., Xiao, L., Zeng, W., Moufarek, A., Murphy, D., Meng, M., Gupta, N., Riedel, F., Das, A., Lawal, E., Narayan, S., Sosea, T., Swirhun, J., Friso, L., Neyshabur, B., Lu, J., Girgin, S., Wunder, M., Yvinec, E., Pyne, A., Carbune, V., Rijhwani, S., Guo, Y., Doshi, T., Briukhov, A., Bain, M., Hitron, A., Wang, X., Gupta, A., Chen, K., Du, C., Zhang, W., Shah, D., Akula, A., Dylla, M., Kachra, A., Kuo, W., Zou, T., Wang, L., Xu, L., Zhu, J., Snyder, J., Menon, S., Firat, O., Mordatch, I., Yuan, Y., Ponomareva, N., Blevins, R., Moore, L., Wang, W., Chen, P., Scholz, M., Dwornik, A., Lin, J., Li, S., Antognini, D., I, T., Song, X., Miller, M., Kalra, U., Raveret, A., Akerlund, O., Wu, F., Nystrom, A., Godbole, N., Liu, T., DeBalsi, H., Zhao, J., Liu, B., Caciularu, A., Lax, L., Khandelwal, U., Langston, V., Bailey, E., Lattanzi, S., Wang, Y., Kovelamudi, N., Mondal, S.,

Guruganesh, G., Hua, N., Roval, O., Wesołowski, P., Ingale, R., Halcrow, J., Sohn, T., Angermueller, C., Raad, B., Stickgold, E., Lu, E., Kosik, A., Xie, J., Lillicrap, T., Huang, A., Zhang, L. L., Paulus, D., Farabet, C., Wertheim, A., Wang, B., Joshi, R., ling Ko, C., Wu, Y., Agrawal, S., Lin, L., Sheng, X., Sung, P., Breland-King, T., Butterfield, C., Gawde, S., Singh, S., Zhang, Q., Apte, R., Shetty, S., Hutter, A., Li, T., Salesky, E., Lebron, F., Kanerva, J., Paganini, M., Nguyen, A., Vallu, R., Peter, J.-T., Velury, S., Kao, D., Hoover, J., Bortsova, A., Bishop, C., Jakobovits, S., Agostini, A., Agarwal, A., Liu, C., Kwong, C., Tavakkol, S., Bica, I., Greve, A., GP, A., Marcus, J., Hou, L., Duerig, T., Moroshko, R., Lacey, D., Davis, A., Amelot, J., Wang, G., Kim, F., Strinopoulos, T., Wan, H., Lan, C. L., Krishnan, S., Tang, H., Humphreys, P., Bai, J., Shtacher, I. H., Machado, D., Pang, C., Burke, K., Liu, D., Aravamudhan, R., Song, Y., Hirst, E., Singh, A., Jou, B., Bai, L., Piccinno, F., Fu, C. K., Alazard, R., Meiri, B., Winter, D., Chen, C., Zhang, M., Heitkaemper, J., Lambert, J., Lee, J., Frömmgen, A., Rogulenko, S., Nair, P., Niemczyk, P., Bulyenov, A., Xu, B., Shemtov, H., Zadimoghaddam, M., Toropov, S., Wirth, M., Dai, H., Gollapudi, S., Zheng, D., Kurakin, A., Lee, C., Bullard, K., Serrano, N., Balazevic, I., Li, Y., Schalkwyk, J., Murphy, M., Zhang, M., Sequeira, K., Datta, R., Agrawal, N., Sutton, C., Attaluri, N., Chiang, M., Farhan, W., Thornton, G., Lin, K., Choma, T., Nguyen, H., Dasgupta, K., Robinson, D., Comşa, I., Riley, M., Pillai, A., Mustafa, B., Golan, B., Zandieh, A., Lespiau, J.-B., Porter, B., Ross, D., Rajayogam, S., Agarwal, M., Venugopalan, S., Shahriari, B., Yan, Q., Xu, H., Tobin, T., Dubov, P., Shi, H., Recasens, A., Kovsharov, A., Borgeaud, S., Dery, L., Vasanth, S., Gribovskaya, E., Qiu, L., Mahdieh, M., Skut, W., Nielsen, E., Zheng, C., Yu, A., Bostock, C. G., Gupta, S., Archer, A., Rawles, C., Davies, E., Svyatkovskiy, A., Tsai, T., Halpern, Y., Reisswig, C., Wydrowski, B., Chang, B., Puigcerver, J., Taege, M. H., Li, J., Schnider, E., Li, X., Dena, D., Xu, Y., Telang, U., Shi, T., Zen, H., Kastner, K., Ko, Y., Subramaniam, N., Kumar, A., Blois, P., Dai, Z., Wieting, J., Lu, Y., Zeldes, Y., Xie, T., Hauth, A., Ţifrea, A., Li, Y., El-Husseini, S., Abolafia, D., Zhou, H., Ding, W., Ghalebikesabi, S., Guía, C., Maksai, A., Ágoston Weisz, Arik, S., Sukhanov, N., Świetlik, A., Jia, X., Yu, L., Wang, W., Brand, M., Bloxwich, D., Kirmani, S., Chen, Z., Go, A., Sprechmann, P., Kannen, N., Carin, A., Sandhu, P., Edkins, I., Nooteboom, L., Gupta, J., Maggiore, L., Azizi, J., Pritch, Y., Yin, P., Gupta, M., Tarlow, D., Smith, D., Ivanov, D., Babaeizadeh, M., Goel, A., Kambala, S., Chu, G., Kastelic, M., Liu, M., Soltau, H., Stone, A., Agrawal, S., Kim, M., Soparkar, K., Tadepalli, S., Bunyan, O., Soh, R., Kannan, A., Kim, D., Chen, B. J., Halumi, A., Roy,

S., Wang, Y., Sercinoglu, O., Gibson, G., Bhatnagar, S., Sano, M., von Dincklage, D., Ren, Q., Mitrevski, B., Olšák, M., She, J., Doersch, C., Jilei, Wang, Liu, B., Tan, Q., Yakar, T., Warkentin, T., Ramirez, A., Lebsack, C., Dillon, J., Mathews, R., Cobley, T., Wu, Z., Chen, Z., Simon, J., Nath, S., Sainath, T., Bendebury, A., Julian, R., Mankalale, B., Ćurko, D., Zacchello, P., Brown, A. R., Sodhia, K., Howard, H., Caelles, S., Gupta, A., Evans, G., Bulanova, A., Katzen, L., Goldenberg, R., Tsitsulin, A., Stanton, J., Schillings, B., Kovalev, V., Fry, C., Shah, R., Lin, K., Upadhyay, S., Li, C., Radpour, S., Maggioni, M., Xiong, J., Haas, L., Brennan, J., Kamath, A., Savinov, N., Nagrani, A., Yacovone, T., Kappedal, R., Andriopoulos, K., Lao, L., Li, Y., Rozhdestvenskiy, G., Hashimoto, K., Audibert, A., Austin, S., Rodriguez, D., Ruoss, A., Honke, G., Karkhanis, D., Xiong, X., Wei, Q., Huang, J., Leng, Z., Premachandran, V., Bileschi, S., Evangelopoulos, G., Mensink, T., Pavagadhi, J., Teplyashin, D., Chang, P., Xue, L., Tanzer, G., Goldman, S., Patel, K., Li, S., Wiesner, J., Zheng, I., Stewart-Binks, I., Han, J., Li, Z., Luo, L., Lenc, K., Lučić, M., Xue, F., Mullins, R., Guseynov, A., Chang, C.-C., Galatzer-Levy, I., Zhang, A., Bingham, G., Hu, G., Hartman, A., Ma, Y., Griffith, J., Irpan, A., Radebaugh, C., Yue, S., Fan, L., Ungureanu, V., Sorokin, C., Teufel, H., Li, P., Anil, R., Paparas, D., Wang, T., Lin, C.-C., Peng, H., Shum, M., Petrovic, G., Brady, D., Nguyen, R., Macherey, K., Li, Z., Singh, H., Yenugula, M., Iinuma, M., Chen, X., Kopparapu, K., Stern, A., Dave, S., Thekkath, C., Perot, F., Kumar, A., Li, F., Xiao, Y., Bilotti, M., Bateni, M. H., Noble, I., Lee, L., Vázquez-Reina, A., Salazar, J., Yang, X., Wang, B., Gruzewska, E., Rao, A., Raghuram, S., Xu, Z., Ben-David, E., Mei, J., Dalmia, S., Zhang, Z., Liu, Y., Bansal, G., Pankov, H., Schwarcz, S., Burns, A., Chan, C., Sanghai, S., Liang, R., Liang, E., He, A., Stuart, A., Narayanan, A., Zhu, Y., Frank, C., Fatemi, B., Sabne, A., Lang, O., Bhattacharya, I., Settle, S., Wang, M., McMahan, B., Tacchetti, A., Soares, L. B., Hadian, M., Cabi, S., Chung, T., Putikhin, N., Li, G., Chen, J., Tarango, A., Michalewski, H., Kazemi, M., Masoom, H., Sheftel, H., Shivanna, R., Vadali, A., Comanescu, R., Reid, D., Moore, J., Neelakantan, A., Sander, M., Herzig, J., Rosenberg, A., Dehghani, M., Choi, J., Fink, M., Hayes, R., Ge, E., Weng, S., Ho, C.-H., Karro, J., Krishna, K., Thiet, L. N., Skerry-Ryan, A., Eppens, D., Andreetto, M., Sarma, N., Bonacina, S., Ayan, B. K., Nawhal, M., Shan, Z., Dusenberry, M., Thakoor, S., Gubbi, S., Nguyen, D. D., Tsarfaty, R., Albanie, S., Mitrović, J., Gandhi, M., Chen, B.-J., Epasto, A., Stephanov, G., Jin, Y., Gehman, S., Amini, A., Weber, J., Behbahani, F., Xu, S., Allamanis, M., Chen, X., Ott, M., Sha, C., Jastrzebski, M., Qi, H., Greene, D., Wu, X., Toki, A., Vlasic, D., Shapiro, J., Kotikalapudi, R.,

Shen, Z., Saeki, T., Xie, S., Cassirer, A., Bharadwaj, S., Kiyono, T., Bhojanapalli, S., Rosenfeld, E., Ritter, S., Mao, J., Oliveira, J. G., Egyed, Z., Bandemer, B., Parisotto, E., Kinoshita, K., Pluto, J., Maniatis, P., Li, S., Guo, Y., Ghiasi, G., Tarbouriech, J., Chatterjee, S., Jin, J., Katrina, Xu, Palomaki, J., Arnold, S., Sewak, M., Piccinini, F., Sharma, M., Albrecht, B., Purser-haskell, S., Vaswani, A., Chen, C., Wisniewski, M., Cao, Q., Aslanides, J., Phu, N. M., Sieb, M., Agubuzu, L., Zheng, A., Sohn, D., Selvi, M., Andreassen, A., Subudhi, K., Eruvbetine, P., Woodman, O., Mery, T., Krause, S., Ren, X., Ma, X., Luo, J., Chen, D., Fan, W., Griffiths, H., Schuler, C., Li, A., Zhang, S., Sarr, J.-M., Luo, S., Patana, R., Watson, M., Naboulsi, D., Collins, M., Sidhwani, S., Hoogeboom, E., Silver, S., Caveness, E., Zhao, X., Rodriguez, M., Deines, M., Bai, L., Griffin, P., Tagliasacchi, M., Xue, E., Babbula, S. R., Pang, B., Ding, N., Shen, G., Peake, E., Crocker, R., Raghvendra, S. S., Swisher, D., Han, W., Singh, R., Wu, L., Pchelin, V., Munkhdalai, T., Alon, D., Bacon, G., Robles, E., Bulian, J., Johnson, M., Powell, G., Ferreira, F. T., Li, Y., Benzing, F., Velimirović, M., Soyer, H., Kong, W., Tony, Nguyên, Yang, Z., Liu, J., van Amersfoort, J., Gillick, D., Sun, B., Rauschmayr, N., Zhang, K., Zhan, S., Zhou, T., Frolov, A., Yang, C., Vnukov, D., Rouillard, L., Li, H., Mandhane, A., Fallen, N., Venkataraman, R., Hu, C. H., Brennan, J., Lee, J., Chang, J., Sundermeyer, M., Pan, Z., Ke, R., Tong, S., Fabrikant, A., Bono, W., Gu, J., Foley, R., Mao, Y., Delakis, M., Bhaswar, D., Frostig, R., Li, N., Zipori, A., Hope, C., Kozlova, O., Mishra, S., Djolonga, J., Schiff, C., Merey, M. A., Briakou, E., Morgan, P., Wan, A., Hassidim, A., Skerry-Ryan, R., Sengupta, K., Jasarevic, M., Kallakuri, P., Kunkle, P., Brennan, H., Lieber, T., Mansoor, H., Walker, J., Zhang, B., Xie, A., Žužić, G., Chukwuka, A., Druinsky, A., Cho, D., Yao, R., Naeem, F., Butt, S., Kim, E., Jia, Z., Jordan, M., Lelkes, A., Kurzeja, M., Wang, S., Zhao, J., Over, A., Chakladar, A., Prasetya, M., Jha, N., Ganapathy, S., Cong, Y., Shroff, P., Saroufim, C., Miryoosefi, S., Hammad, M., Nasir, T., Xi, W., Gao, Y., Maeng, Y., Hora, B., Cheng, C.-Y., Haghani, P., Lewenberg, Y., Lu, C., Matysiak, M., Raisinghani, N., Wang, H., Baugher, L., Sukthankar, R., Giang, M., Schultz, J., Fiedel, N., Chen, M., Lee, C.-C., Dey, T., Zheng, H., Paul, S., Smith, C., Ly, A., Wang, Y., Bansal, R., Perz, B., Ricco, S., Blank, S., Keshava, V., Sharma, D., Chow, M., Lad, K., Jalan, K., Osindero, S., Swanson, C., Scott, J., Ilić, A., Li, X., Jonnalagadda, S. R., Soudagar, A. S., Xiong, Y., Batsaikhan, B.-O., Jarrett, D., Kumar, N., Shah, M., Lawlor, M., Waters, A., Graham, M., May, R., Ramos, S., Lefdal, S., Cankara, Z., Cano, N., O'Donoghue, B., Borovik, J., Liu, F., Grimstad, J., Alnahlawi, M., Tsihlas, K., Hudson, T., Grigorev, N., Jia, Y., Huang, T., Igwe, T. P.,

Lebedev, S., Tang, X., Krivokon, I., Garcia, F., Tan, M., Jia, E., Stys, P., Vashishth, S., Liang, Y., Venkatraman, B., Gu, C., Kementsietsidis, A., Zhu, C., Jung, J., Bai, Y., Hosseini, M. J., Ahmed, F., Gupta, A., Yuan, X., Ashraf, S., Nigam, S., Vasudevan, G., Awasthi, P., Gilady, A. M., Mariet, Z., Eskander, R., Li, H., Hu, H., Garrido, G., Schlattner, P., Zhang, G., Saxena, R., Dević, P., Muralidharan, K., Murthy, A., Zhou, Y., Choi, M., Wongpanich, A., Wang, Z., Shah, P., Xu, Y., Huang, Y., Spencer, S., Chen, A., Cohan, J., Wang, J., Tompson, J., Wu, J., Haroun, R., Li, H., Huergo, B., Yang, F., Yin, T., Wendt, J., Bendersky, M., Chaabouni, R., Snaider, J., Ferret, J., Jindal, A., Thompson, T., Xue, A., Bishop, W., Phal, S. M., Sharma, A., Sung, Y., Radhakrishnan, P., Shomrat, M., Ingle, R., Vij, R., Gilmer, J., Istin, M. D., Sobell, S., Lu, Y., Nottage, E., Sadigh, D., Willcock, J., Zhang, T., Xu, S., Brown, S., Lee, K., Wang, G., Zhu, Y., Tay, Y., Kim, C., Gutierrez, A., Sharma, A., Xian, Y., Seo, S., Cui, C., Pochernina, E., Baetu, C., Jastrzębski, K., Ly, M., Elhawaty, M., Suh, D., Sezener, E., Wang, P., Yuen, N., Tucker, G., Cai, J., Yang, Z., Wang, C., Muzio, A., Qian, H., Yoo, J., Lockhart, D., McKee, K. R., Guo, M., Mehrotra, M., Mendonça, A., Mehta, S. V., Ben, S., Tekur, C., Mu, J., Zhu, M., Krakovna, V., Lee, H., Maschinot, A., Cevey, S., Choe, H., Bai, A., Srinivasan, H., Gasaway, D., Young, N., Siegler, P., Holtmann-Rice, D., Piratla, V., Baumli, K., Yogev, R., Hofer, A., van Hasselt, H., Grant, S., Chervonyi, Y., Silver, D., Hogue, A., Agarwal, A., Wang, K., Singh, P., Flynn, F., Lipschultz, J., David, R., Bellot, L., Yang, Y.-Y., Le, L., Graziano, F., Olszewska, K., Hui, K., Maurya, A., Parotsidis, N., Chen, W., Oguntebi, T., Kelley, J., Baddepudi, A., Mauerer, J., Shaw, G., Siegman, A., Yang, L., Shetty, S., Roy, S., Song, Y., Stokowiec, W., Burnell, R., Savant, O., Busa-Fekete, R., Miao, J., Ghosh, S., MacDermed, L., Lippe, P., Dektiarev, M., Behrman, Z., Mentzer, F., Nguyen, K., Wei, M., Verma, S., Knutsen, C., Dasari, S., Yan, Z., Mitrichev, P., Wang, X., Shejwalkar, V., Austin, J., Sunkara, S., Potti, N., Virin, Y., Wright, C., Liu, G., Riva, O., Pot, E., Kochanski, G., Le, Q., Balasubramaniam, G., Dhar, A., Liao, Y., Bloniarz, A., Shukla, D., Cole, E., Lee, J., Zhang, S., Kafle, S., Vashishtha, S., Mahmoudieh, P., Chen, G., Hoffmann, R., Srinivasan, P., Lago, A. D., Shalom, Y. B., Wang, Z., Elabd, M., Sharma, A., Oh, J., Kothawade, S., Le, M., Monteiro, M., Yang, S., Alarakyia, K., Geirhos, R., Mincu, D., Garnes, H., Kobayashi, H., Mariooryad, S., Krasowiak, K., Zhixin, Lai, Mourad, S., Wang, M., Bu, F., Aharoni, O., Chen, G., Goyal, A., Zubov, V., Bapna, A., Dabir, E., Kothari, N., Lamerigts, K., Cao, N. D., Shar, J., Yew, C., Kulkarni, N., Mahaarachchi, D., Joshi, M., Zhu, Z., Lichtarge, J., Zhou, Y., Muckenhirn, H., Selo, V., Vinyals, O., Chen, P., Brohan, A., Mehta, V., Cogan,

S., Wang, R., Geri, T., Ko, W.-J., Chen, W., Viola, F., Shivam, K., Wang, L., Elish, M. C., Popa, R. A., Pereira, S., Liu, J., Koster, R., Kim, D., Zhang, G., Ebrahimi, S., Talukdar, P., Zheng, Y., Poklukar, P., Mikhalap, A., Johnson, D., Vijayakumar, A., Omernick, M., Dibb, M., Dubey, A., Hu, Q., Suman, A., Aggarwal, V., Kornakov, I., Xia, F., Lowe, W., Kolganov, A., Xiao, T., Nikolaev, V., Hemingray, S., Li, B., Iljazi, J., Rybiński, M., Sandhu, B., Lu, P., Luong, T., Jenatton, R., Govindaraj, V., Hui, Li, Dulac-Arnold, G., Park, W., Wang, H., Modi, A., Pouget-Abadie, J., Greller, K., Gupta, R., Berry, R., Ramachandran, P., Xie, J., McCafferty, L., Wang, J., Gupta, K., Lim, H., Bratanič, B., Brock, A., Akolzin, I., Sproch, J., Karliner, D., Kim, D., Goedeckemeyer, A., Shazeer, N., Schmid, C., Calandriello, D., Bhatia, P., Choromanski, K., Montgomery, C., Dua, D., Ramalho, A., King, H., Gao, Y., Nguyen, L., Lindner, D., Pitta, D., Johnson, O., Salama, K., Ardila, D., Han, M., Farnese, E., Odoom, S., Wang, Z., Ding, X., Rink, N., Smith, R., Lehri, H. T., Cohen, E., Vats, N., He, T., Gopavarapu, P., Paszke, A., Patel, M., Gansbeke, W. V., Loher, L., Castro, L., Voitovich, M., von Glehn, T., George, N., Niklaus, S., Eaton-Rosen, Z., Rakićević, N., Jue, E., Perel, S., Zhang, C., Bahat, Y., Pouget, A., Xing, Z., Huot, F., Shenoy, A., Bos, T., Coriou, V., Richter, B., Noy, N., Wang, Y., Ontanon, S., Qin, S., Makarchuk, G., Hassabis, D., Li, Z., Sharma, M., Venkatesan, K., Kemaev, I., Daniel, R., Huang, S., Shah, S., Ponce, O., Warren, Chen, Faruqui, M., Wu, J., Andačić, S., Payrits, S., McDuff, D., Hume, T., Cao, Y., Tessler, M., Wang, Q., Wang, Y., Rendulic, I., Agustsson, E., Johnson, M., Lando, T., Howard, A., Padmanabhan, S. G. S., Daswani, M., Banino, A., Kilgore, M., Heek, J., Ji, Z., Caceres, A., Li, C., Kassner, N., Vlaskin, A., Liu, Z., Grills, A., Hou, Y., Sukkerd, R., Cheon, G., Shetty, N., Markeeva, L., Stanczyk, P., Iyer, T., Gong, Y., Gao, S., Gopalakrishnan, K., Blyth, T., Reynolds, M., Bhoopchand, A., Bilenko, M., Gharibian, D., Zayats, V., Faust, A., Singh, A., Ma, M., Jiao, H., Vijayanarasimhan, S., Aroyo, L., Yadav, V., Chakera, S., Kakarla, A., Meshram, V., Gregor, K., Botea, G., Senter, E., Jia, D., Kovacs, G., Sharma, N., Baur, S., Kang, K., He, Y., Zhuo, L., Kostelac, M., Laish, I., Peng, S., O'Bryan, L., Kasenberg, D., Rao, G. R., Leurent, E., Zhang, B., Stevens, S., Salazar, A., Zhang, Y., Lobov, I., Walker, J., Porter, A., Redshaw, M., Ke, H., Rao, A., Lee, A., Lam, H., Moffitt, M., Kim, J., Qiao, S., Koo, T., Dadashi, R., Song, X., Sundararajan, M., Xu, P., Kawamoto, C., Zhong, Y., Barbu, C., Reddy, A., Verzetti, M., Li, L., Papamakarios, G., Klimczak-Plucińska, H., Cassin, M., Kavukcuoglu, K., Swavely, R., Vaucher, A., Zhao, J., Hemsley, R., Tschannen, M., Ge, H., Menghani, G., Yu, Y., Ha, N., He, W., Wu, X., Song, M., Sterneck, R., Zinke, S., Calian, D. A.,

Marsden, A., Ruiz, A. C., Hessel, M., Gueta, A., Lee, B., Farris, B., Gupta, M., Li, Y., Saleh, M., Misra, V., Xiao, K., Mendolicchio, P., Buttimore, G., Krayvanova, V., Nayakanti, N., Wiethoff, M., Pande, Y., Mirhoseini, A., Lao, N., Liu, J., Hua, Y., Chen, A., Malkov, Y., Kalashnikov, D., Gupta, S., Audhkhasi, K., Zhai, Y., Kopalle, S., Jain, P., Ofek, E., Meyer, C., Baatarsukh, K., Strejček, H., Qian, J., Freedman, J., Figueira, R., Sokolik, M., Bachem, O., Lin, R., Kharrat, D., Hidey, C., Xu, P., Duan, D., Li, Y., Ersoy, M., Everett, R., Cen, K., Santamaria-Fernandez, R., Taubenfeld, A., Mackinnon, I., Deng, L., Zablotskaia, P., Viswanadha, S., Goel, S., Yates, D., Deng, Y., Choy, P., Chen, M., Sinha, A., Mossin, A., Wang, Y., Szlam, A., Hao, S., Rubenstein, P. K., Toksoz-Exley, M., Aperghis, M., Zhong, Y., Ahn, J., Isard, M., Lacombe, O., Luisier, F., Anastasiou, C., Kalley, Y., Prabhu, U., Dunleavy, E., Bijwadia, S., Mao-Jones, J., Chen, K., Pasumarthi, R., Wood, E., Dostmohamed, A., Hurley, N., Simsa, J., Parrish, A., Pajarskas, M., Harvey, M., Skopek, O., Kochinski, Y., Rey, J., Rieser, V., Zhou, D., Lee, S. J., Acharya, T., Li, G., Jiang, J., Zhang, X., Gipson, B., Mahintorabi, E., Gelmi, M., Khajehnouri, N., Yeh, A., Lee, K., Matthey, L., Baker, L., Pham, T., Fu, H., Pak, A., Gupta, P., Vasconcelos, C., Sadovsky, A., Walker, B., Hsiao, S., Zochbauer, P., Marzoca, A., Velan, N., Zeng, J., Baechler, G., Driess, D., Jain, D., Huang, Y., Tao, L., Maggs, J., Levine, N., Schneider, J., Gemzer, E., Petit, S., Han, S., Fisher, Z., Zelle, D., Biles, C., Ie, E., Fadeeva, A., Liu, C., Franco, J. V., Collister, A., Zhang, H., Wang, R., Zhao, R., Kieliger, L., Shuster, K., Zhu, R., Gong, B., Chan, L., Sun, R., Basu, S., Zimmermann, R., Hayes, J., Bapna, A., Snoek, J., Yang, W., Datta, P., Abdallah, J. A., Kilgour, K., Li, L., Mah, S., Jun, Y., Rivière, M., Karmarkar, A., Spalink, T., Huang, T., Gonzalez, L., Tran, D.-H., Nowak, A., Palowitch, J., Chadwick, M., Talius, E., Mehta, H., Sellam, T., Fränken, P., Nicosia, M., He, K., Kini, A., Amos, D., Basu, S., Jobe, H., Shaw, E., Xu, Q., Evans, C., Ikeda, D., Yan, C., Jin, L., Wang, L., Yadav, S., Labzovsky, I., Sampath, R., Ma, A., Schumann, C., Siddhant, A., Shah, R., Youssef, J., Agarwal, R., Dabney, N., Tonioni, A., Ambar, M., Li, J., Guyon, I., Li, B., Soergel, D., Fang, B., Karadzhov, G., Udrescu, C., Trinh, T., Raunak, V., Noury, S., Guo, D., Gupta, S., Finkelstein, M., Petek, D., Liang, L., Billock, G., Sun, P., Wood, D., Song, Y., Yu, X., Matejovicova, T., Cohen, R., Andra, K., D'Ambrosio, D., Deng, Z., Nallatamby, V., Songhori, E., Dangovski, R., Lampinen, A., Botadra, P., Hillier, A., Cao, J., Baddi, N., Kuncoro, A., Yoshino, T., Bhagatwala, A., Ranzato, M., Schaeffer, R., Liu, T., Ye, S., Sarvana, O., Nham, J., Kuang, C., Gao, I., Baek, J., Mittal, S., Wahid, A., Gergely, A., Ni, B., Feldman, J., Muir, C., Lamblin, P., Macherey, W.,

Dyer, E., Kilpatrick, L., Campos, V., Bhutani, M., Fort, S., Ahmad, Y., Severyn, A., Chatziprimou, K., Ferludin, O., Dimarco, M., Kusupati, A., Heyward, J., Bahir, D., Villela, K., Millican, K., Marcus, D., Bahargam, S., Unlu, C., Roth, N., Wei, Z., Gopal, S., Ghoshal, D., Lee, E., Lin, S., Lees, J., Lee, D., Hosseini, A., Fan, C., Neel, S., Wu, M., Altun, Y., Cai, H., Piqueras, E., Woodward, J., Bissacco, A., Haykal, S., Bordbar, M., Sundaram, P., Hodkinson, S., Toyama, D., Polovets, G., Myers, A., Sinha, A., Levinboim, T., Krishnakumar, K., Chhaparia, R., Sholokhova, T., Gundavarapu, N. B., Jawahar, G., Qureshi, H., Hu, J., Momchev, N., Rahtz, M., Wu, R., S, A. P., Dhamdhere, K., Guo, M., Gupta, U., Eslami, A., Schain, M., Blokzijl, M., Welling, D., Orr, D., Bolelli, L., Perez-Nieves, N., Sirotenko, M., Prasad, A., Kar, A., Pigem, B. D. B., Terzi, T., Weisz, G., Ghosh, D., Mavalankar, A., Madeka, D., Daugaard, K., Adam, H., Shah, V., Berman, D., Tran, M., Baker, S., Andrejczuk, E., Chole, G., Raboshchuk, G., Mirzazadeh, M., Kagohara, T., Wu, S., Schallhart, C., Orlando, B., Wang, C., Rrustemi, A., Xiong, H., Liu, H., Vezer, A., Ramsden, N., yiin Chang, S., Mudgal, S., Li, Y., Vieillard, N., Hoshen, Y., Ahmad, F., Slone, A., Hua, A., Potikha, N., Rossini, M., Stritar, J., Prakash, S., Wang, Z., Dong, X., Nazari, A., Nehoran, E., Tekelioglu, K., Li, Y., Badola, K., Funkhouser, T., Li, Y., Yerram, V., Ganeshan, R., Formoso, D., Langner, K., Shi, T., Li, H., Yamamori, Y., Panda, A., Saade, A., Scarpati, A. S., Breaux, C., Carey, C., Zhou, Z., Hsieh, C.-J., Bridgers, S., Butryna, A., Gupta, N., Tulsyan, V., Woo, S., Eltyshev, E., Grathwohl, W., Parks, C., Benjamin, S., Panigrahy, R., Dodhia, S., Freitas, D. D., Sauer, C., Song, W., Alet, F., Tolins, J., Paduraru, C., Zhou, X., Albert, B., Zhang, Z., Shu, L., Bansal, M., Nguyen, S., Globerson, A., Xiao, O., Manyika, J., Hennigan, T., Rong, R., Matak, J., Bakalov, A., Sharma, A., Sinopalnikov, D., Pierson, A., Roller, S., Brown, G., Gao, M., Fukuzawa, T., Ghafouri, A., Vassigh, K., Barr, I., Wang, Z., Korsun, A., Jayaram, R., Ren, L., Zaman, T., Khan, S., Lunts, Y., Deutsch, D., Uthus, D., Katz, N., Samsikova, M., Khalifa, A., Sethi, N., Sun, J., Tang, L., Alon, U., Luo, X., Yu, D., Nayyar, A., Petrini, B., Truong, W., Hellendoorn, V., Chinaev, N., Alberti, C., Wang, W., Hu, J., Mirrokni, V., Balashankar, A., Aharon, A., Mehta, A., Iscen, A., Kready, J., Manning, L., Mohananey, A., Chen, Y., Tripathi, A., Wu, A., Petrovski, I., Hwang, D., Baeuml, M., Chandrakaladharan, S., Liu, Y., Coaguila, R., Chen, M., Ma, S., Tafti, P., Tatineni, S., Spitz, T., Ye, J., Vicol, P., Rosca, M., Puigdomènech, A., Yahav, Z., Ghemawat, S., Lin, H., Kirk, P., Nabulsi, Z., Brin, S., Bohnet, B., Caluwaerts, K., Veerubhotla, A. S., Zheng, D., Dai, Z., Petrov, P., Xu, Y., Mehran, R., Xu, Z., Zintgraf, L., Choi, J., Hombaiah, S. A., Thoppilan, R., Reddi, S., Lew, L., Li, L.,

Webster, K., Sawhney, K., Lamprou, L., Shakeri, S., Lunayach, M., Chen, J., Bagri, S., Salcianu, A., Chen, Y., Donchev, Y., Magister, C., Nørly, S., Rodrigues, V., Izo, T., Noga, H., Zou, J., Köppe, T., Zhou, W., Lee, K., Long, X., Eisenbud, D., Chen, A., Schenck, C., To, C. M., Zhong, P., Taropa, E., Truong, M., Levy, O., Martins, D., Zhang, Z., Semturs, C., Zhang, K., Yakubovich, A., Moreno, P., McConnaughey, L., Lu, D., Redmond, S., Weerts, L., Bitton, Y., Refice, T., Lacasse, N., Conmy, A., Tallec, C., Odell, J., Forbes-Pollard, H., Socala, A., Hoech, J., Kohli, P., Walton, A., Wang, R., Sazanovich, M., Zhu, K., Kapishnikov, A., Galt, R., Denton, M., Murdoch, B., Sikora, C., Mohamed, K., Wei, W., First, U., McConnell, T., Cobo, L. C., Qin, J., Avrahami, T., Balle, D., Watanabe, Y., Louis, A., Kraft, A., Ariafar, S., Gu, Y., Rives, E., Yoon, C., Rusu, A., Cobon-Kerr, J., Hahn, C., Luo, J., Yuvein, Zhu, Ahuja, N., Benenson, R., Kaufman, R. L., Yu, H., Hightower, L., Zhang, J., Ni, D., Hendricks, L. A., Wang, G., Yona, G., Jain, L., Barrio, P., Bhupatiraju, S., Velusamy, S., Dafoe, A., Riedel, S., Thomas, T., Yuan, Z., Bellaiche, M., Panthaplackel, S., Kloboves, K., Jauhari, S., Akbulut, C., Davchev, T., Gladchenko, E., Madras, D., Chuklin, A., Hill, T., Yuan, Q., Madhavan, M., Leonhard, L., Scandinaro, D., Chen, Q., Niu, N., Douillard, A., Damoc, B., Onoe, Y., Pedregosa, F., Bertsch, F., Leichner, C., Pagadora, J., Malmaud, J., Ponda, S., Twigg, A., Duzhyi, O., Shen, J., Wang, M., Garg, R., Chen, J., Evci, U., Lee, J., Liu, L., Kojima, K., Yamaguchi, M., Rajendran, A., Piergiovanni, A., Rajendran, V. K., Fornoni, M., Ibagon, G., Ragan, H., Khan, S. M., Blitzer, J., Bunner, A., Sun, G., Kosakai, T., Lundberg, S., Elue, N., Guu, K., Park, S., Park, J., Narayanaswamy, A., Wu, C., Mudigonda, J., Cohn, T., Mu, H., Kumar, R., Graesser, L., Zhang, Y., Killam, R., Zhuang, V., Giménez, M., Jishi, W. A., Ley-Wild, R., Zhai, A., Osawa, K., Cedillo, D., Liu, J., Upadhyay, M., Sieniek, M., Sharma, R., Paine, T., Angelova, A., Addepalli, S., Parada, C., Majumder, K., Lamp, A., Kumar, S., Deng, X., Myaskovsky, A., Sabolić, T., Dudek, J., York, S., de Chaumont Quitry, F., Nie, J., Cattle, D., Gunjan, A., Piot, B., Khawaja, W., Bang, S., Wang, S., Khodadadeh, S., R, R., Rawlani, P., Powell, R., Lee, K., Griesser, J., Oh, G., Magalhaes, C., Li, Y., Tokumine, S., Vogel, H. N., Hsu, D., BC, A., Jindal, D., Cohen, M., Yang, Z., Yuan, J., de Cesare, D., Bruguier, T., Xu, J., Roy, M., Jacovi, A., Belov, D., Arya, R., Meadowlark, P., Cohen-Ganor, S., Ye, W., Morris-Suzuki, P., Banzal, P., Song, G., Ponnuramu, P., Zhang, F., Scrivener, G., Zaiem, S., Rochman, A. R., Han, K., Ghazi, B., Lee, K., Drath, S., Suo, D., Girgis, A., Shenoy, P., Nguyen, D., Eck, D., Gupta, S., Yan, L., Carreira, J., Gulati, A., Sang, R., Mirylenka, D., Cooney, E., Chou, E., Ling, M., Fan, C., Coleman, B., Tubone, G., Kumar, R.,

Baldridge, J., Hernandez-Campos, F., Lazaridou, A., Besley, J., Yona, I., Bulut, N., Wellens, Q., Pierigiovanni, A., George, J., Green, R., Han, P., Tao, C., Clark, G., You, C., Abdolmaleki, A., Fu, J., Chen, T., Chaugule, A., Chandorkar, A., Rahman, A., Thompson, W., Koanantakool, P., Bernico, M., Ren, J., Vlasov, A., Vassilvitskii, S., Kula, M., Liang, Y., Kim, D., Huang, Y., Ye, C., Lepikhin, D., and Helmholz, W. (2025). Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., and Salakhutdinov, R. (2019). Transformer-XL: Attentive language models beyond a fixed-length context. In Korhonen, A., Traum, D., and Màrquez, L., editors, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Daniel, N., Radev, D., and Allison, T. (2003). Sub-event based multi-document summarization. In Proceedings of the HLT-NAACL 03 Text Summarization Workshop, pages 9–16.

Demszky, D., Guu, K., and Liang, P. (2018). Transforming question answering datasets into natural language inference datasets.

Deutsch, D., Bedrax-Weiss, T., and Roth, D. (2021). Towards question-answering as an automatic metric for evaluating the content quality of a summary. Transactions of the Association for Computational Linguistics, 9:774–789.

Deutsch, D. and Roth, D. (2021). Understanding the extent to which content quality metrics measure the information quality of summaries. In Bisazza, A. and Abend, O., editors, Proceedings of the 25th Conference on Computational Natural Language Learning, pages 300–309, Online. Association for Computational Linguistics.

Deutsch, D. and Roth, D. (2023). Incorporating question answering-based signals into abstractive summarization via salient span selection. In Vlachos, A. and Augenstein, I., editors, Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 575–588, Dubrovnik, Croatia. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, 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.

Ding, Y., Zhang, L. L., Zhang, C., Xu, Y., Shang, N., Xu, J., Yang, F., and Yang, M. (2024). Longrope: extending llm context window beyond 2 million tokens. In Proceedings of the 41st International Conference on Machine Learning, ICML'24. JMLR.org.

Dobrovolskii, V. (2021). Word-level coreference resolution. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 7670–7675, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dror, R., Baumer, G., Shlomov, S., and Reichart, R. (2018). The hitchhiker's guide to testing statistical significance in natural language processing. In Gurevych, I. and Miyao, Y., editors, Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Dyer, C., Ballesteros, M., Ling, W., Matthews, A., and Smith, N. A. (2015). Transition-based dependency parsing with stack long short-term memory. In Zong, C. and Strube, M., editors, 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 334–343, Beijing, China. Association for Computational Linguistics.

Efron, B. and Tibshirani, R. J. (1994). An Introduction to the Bootstrap. Chapman and Hall/CRC.

Eschenbach, C., Habel, C., Herweg, M., and Rehkamper, K. (1989). Remarks on plural anaphora. In Somers, H. and McGee Wood, M., editors, Fourth Conference of the European Chapter of the Association for Computational Linguistics, Manchester, England. Association for Computational Linguistics.

Fabbri, A., Li, I., She, T., Li, S., and Radev, D. (2019). Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In Korhonen, A., Traum, D., and Màrquez, L., editors, Proceedings of the 57th Annual Meeting

of the Association for Computational Linguistics, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Fabbri, A., Wu, C.-S., Liu, W., and Xiong, C. (2022). QAFactEval: Improved QA-based factual consistency evaluation for summarization. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V., editors, Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2587–2601, Seattle, United States. Association for Computational Linguistics.

Fierro, C., Amplayo, R. K., Huot, F., De Cao, N., Maynez, J., Narayan, S., and Lapata, M. (2024). Learning to plan and generate text with citations. In Ku, L.-W., Martins, A., and Srikumar, V., editors, Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 11397–11417, Bangkok, Thailand. Association for Computational Linguistics.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., Presser, S., and Leahy, C. (2020). The pile: An 800gb dataset of diverse text for language modeling.

Ginzburg, J. (1994). An update semantics for dialogue. In Proceedings of the 1st Tilburg International Workshop on Computational Semantics.

Glikson, E. and Woolley, A. W. (2020). Human trust in artificial intelligence: Review of empirical research. Academy of Management Annals, 14(2):627–660.

Godbole, A., Monath, N., Kim, S., Rawat, A. S., McCallum, A., and Zaheer, M. (2024). Analysis of plan-based retrieval for grounded text generation. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 13101–13119, Miami, Florida, USA. Association for Computational Linguistics.

Godfrey, J. J., Holliman, E. C., and McDaniel, J. (1992). Switchboard: Telephone speech corpus for research and development. In Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing - Volume 1, ICASSP'92, page 517–520, USA. IEEE Computer Society.

Grenander, M., Cohen, S. B., and Steedman, M. (2022). Sentence-incremental neural coreference resolution. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors,

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 427–443, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Grenander, M., Varia, S., Czarnowska, P., Vyas, Y., Halder, K., and Min, B. (2025). Exploration of plan-guided summarization for narrative texts: the case of small language models. In 7th Workshop on Narrative Understanding, Albuquerque, New Mexico and Online. Association for Computational Linguistics.

Grosz, B. J., Joshi, A. K., and Weinstein, S. (1995). Centering: A framework for modeling the local coherence of discourse. Computational Linguistics, 21(2):203–225.

Grosz, B. J. and Sidner, C. L. (1986). Attention, intentions, and the structure of discourse. Computational Linguistics, 12(3):175–204.

Gunjal, A. and Durrett, G. (2024). Molecular facts: Desiderata for decontextualization in LLM fact verification. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, Findings of the Association for Computational Linguistics: EMNLP 2024, pages 3751–3768, Miami, Florida, USA. Association for Computational Linguistics.

Guo, M., Ainslie, J., Uthus, D., Ontanon, S., Ni, J., Sung, Y.-H., and Yang, Y. (2022). LongT5: Efficient text-to-text transformer for long sequences. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V., editors, Findings of the Association for Computational Linguistics: NAACL 2022, pages 724–736, Seattle, United States. Association for Computational Linguistics.

Haddock, N. (1989). Incremental Semantics and Interactive Syntactic Processing. PhD thesis, University of Edinburgh.

Halliday, M. A. K. (1967). Notes on transitivity and theme in english: Part 2. Journal of Linguistics, 3(2):199–244.

Heim, I. (1982). The Semantics of Definite and Indefinite Noun Phrases. PhD thesis, University of Massachusetts Amherst, Amherst, Massachusetts, USA. Available at `https://semanticsarchive.net/Archive/jA2YTJmN/Heim%20Dissertation%20with%20Hyperlinks.pdf`.

Heim, I. (1983). File change semantics and the familiarity theory of definiteness. In Bäuerle, R., Schwarze, C., and von Stechow, A., editors, Meaning, Use, and Interpretation of Language, pages 164–189. De Gruyter, Berlin, Boston.

Hermann, K. M., Kociský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. In NIPS, pages 1693–1701.

Hirschman, L. and Chinchor, N. (1998). Appendix F: MUC-7 coreference task definition (version 3.0). In Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998.

Hirst, G. (1981). Discourse-oriented anaphora resolution in natural language understanding: a review. American Journal of Computational Linguistics, 7(2):85–98.

Hirst, G. (1987). Semantic Interpretation and the Resolution of Ambiguity. Cambridge University Press, Cambridge.

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

Honnibal, M., Montani, I., Van Landeghem, S., and Boyd, A. (2020). spaCy: Industrial-strength Natural Language Processing in Python.

Honovich, O., Aharoni, R., Herzig, J., Taitelbaum, H., Kukliansy, D., Cohen, V., Scialom, T., Szpektor, I., Hassidim, A., and Matias, Y. (2022). TRUE: Re-evaluating factual consistency evaluation. In Feng, S., Wan, H., Yuan, C., and Yu, H., editors, Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering, pages 161–175, Dublin, Ireland. Association for Computational Linguistics.

Hou, Y. (2021). End-to-end neural information status classification. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, Findings of the Association for Computational Linguistics: EMNLP 2021, pages 1377–1388, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2006). OntoNotes: The 90% solution. In Moore, R. C., Bilmes, J., Chu-Carroll, J., and Sanderson, M., editors, Proceedings of the Human Language Technology Conference of the

NAACL, Companion Volume: Short Papers, pages 57–60, New York City, USA. Association for Computational Linguistics.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2022). LoRA: Low-rank adaptation of large language models. In International Conference on Learning Representations.

Hua, Y., Deng, Z., and McKeown, K. (2023). Improving long dialogue summarization with semantic graph representation. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, Findings of the Association for Computational Linguistics: ACL 2023, pages 13851–13883, Toronto, Canada. Association for Computational Linguistics.

Huot, F., Maynez, J., Narayan, S., Amplayo, R. K., Ganchev, K., Louis, A. P., Sandholm, A., Das, D., and Lapata, M. (2023). Text-blueprint: An interactive platform for plan-based conditional generation. In Croce, D. and Soldaini, L., editors, Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, pages 105–116, Dubrovnik, Croatia. Association for Computational Linguistics.

Hwang, E., Zhou, Y., Gunel, B., Wendt, J. B., and Tata, S. (2025). SUMIE: A synthetic benchmark for incremental entity summarization. In Rambow, O., Wanner, L., Apidianaki, M., Al-Khalifa, H., Eugenio, B. D., and Schockaert, S., editors, Proceedings of the 31st International Conference on Computational Linguistics, pages 10839–10864, Abu Dhabi, UAE. Association for Computational Linguistics.

Hwang, E., Zhou, Y., Wendt, J. B., Gunel, B., Vo, N., Xie, J., and Tata, S. (2024). Enhancing incremental summarization with structured representations. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, Findings of the Association for Computational Linguistics: EMNLP 2024, pages 3830–3842, Miami, Florida, USA. Association for Computational Linguistics.

Ingria, R. J. P. and Stallard, D. (1989). A computational mechanism for pronominal reference. In 27th Annual Meeting of the Association for Computational Linguistics, pages 262–271, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. (2023). Survey of hallucination in natural language generation. ACM Comput. Surv., 55(12).

Jin, X., Li, M., and Ji, H. (2022). Event schema induction with double graph autoencoders. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V., editors, Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 2013–2025, Seattle, United States. Association for Computational Linguistics.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. (2020). SpanBERT: Improving pre-training by representing and predicting spans. Transactions of the Association for Computational Linguistics, 8:64–77.

Joshi, M., Levy, O., Zettlemoyer, L., and Weld, D. (2019). BERT for coreference resolution: Baselines and analysis. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, 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 5803–5808, Hong Kong, China. Association for Computational Linguistics.

Kamp, H. (1981). A theory of truth and semantic representation. In Groenendijk, J. A. G., Janssen, T. M. V., and Stokhof, M. J. B., editors, Formal methods in the study of language, pages 277–322. Mathematisch Centrum, Amsterdam.

Kamp, H. and Reyle, U. (1993). From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory. Kluwer Academic Publishers, Dordrecht.

Karttunen, L. (1969). Discourse referents. In Proceedings of the 1969 conference on Computational Linguistics, pages 1–38. reprinted as Karttunen 1976.

Karttunen, L. (1976). Discourse referents. In McCawley, J., editor, Syntax and Semantics, volume 7, pages 363–385. Academic Press. reprinted in Javier Gutieérrez-Rexach (ed.), Semantics: Critical Concepts in Linguistics Vol. III: Noun Phrase Classes, 20-39, Routledge.

Khosla, S., Yu, J., Manuvinakurike, R., Ng, V., Poesio, M., Strube, M., and Rosé, C. (2021). The CODI-CRAC 2021 shared task on anaphora, bridging, and discourse deixis in dialogue. In Khosla, S., Manuvinakurike, R., Ng, V., Poesio, M., Strube, M., and Rosé, C., editors, Proceedings of the CODI-CRAC 2021 Shared Task on Anaphora, Bridging, and Discourse Deixis in Dialogue, pages 1–15, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kim, Y., Chang, Y., Karpinska, M., Garimella, A., Manjunatha, V., Lo, K., Goyal, T., and Iyyer, M. (2024). FABLES: Evaluating faithfulness and content selection in book-length summarization. In First Conference on Language Modeling.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Kirstain, Y., Ram, O., and Levy, O. (2021). Coreference resolution without span representations. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 14–19, Online. Association for Computational Linguistics.

Klenner, M. and Tuggener, D. (2011). An incremental entity-mention model for coreference resolution with restrictive antecedent accessibility. In Mitkov, R. and Angelova, G., editors, Proceedings of the International Conference Recent Advances in Natural Language Processing 2011, pages 178–185, Hissar, Bulgaria. Association for Computational Linguistics.

Kočiský, T., Schwarz, J., Blunsom, P., Dyer, C., Hermann, K. M., Melis, G., and Grefenstette, E. (2018). The NarrativeQA reading comprehension challenge. Transactions of the Association for Computational Linguistics, 6:317–328.

Kryscinski, W., Keskar, N. S., McCann, B., Xiong, C., and Socher, R. (2019). Neural text summarization: A critical evaluation. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, 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 540–551, Hong Kong, China. Association for Computational Linguistics.

Kryscinski, W., Rajani, N., Agarwal, D., Xiong, C., and Radev, D. (2022). BOOKSUM: A collection of datasets for long-form narrative summarization. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, Findings of the Association for Computational Linguistics: EMNLP 2022, pages 6536–6558, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kudo, T. and Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Blanco, E. and Lu, W., editors, Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Laban, P., Schnabel, T., Bennett, P. N., and Hearst, M. A. (2022). SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. Transactions of the Association for Computational Linguistics, 10:163–177.

Le, N. T. and Ritter, A. (2024). Are language models robust coreference resolvers? In First Conference on Language Modeling.

Lee, K., He, L., Lewis, M., and Zettlemoyer, L. (2017). End-to-end neural coreference resolution. In Palmer, M., Hwa, R., and Riedel, S., editors, Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Lee, K., He, L., and Zettlemoyer, L. (2018). Higher-order coreference resolution with coarse-to-fine inference. In Walker, M., Ji, H., and Stent, A., editors, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 687–692, New Orleans, Louisiana. Association for Computational Linguistics.

Lei, Y. and Huang, R. (2025). Multi-document summarization through multi-document event relation graph reasoning in LLMs: a case study in framing bias mitigation. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T., editors, Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 26603–26619, Vienna, Austria. Association for Computational Linguistics.

Levy, M., Jacoby, A., and Goldberg, Y. (2024). Same task, more tokens: the impact of input length on the reasoning performance of large language models. In Ku, L.-W., Martins, A., and Srikumar, V., editors, Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15339–15353, Bangkok, Thailand. Association for Computational Linguistics.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7871–7880, Online. Association for Computational Linguistics.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Lin, S., Hilton, J., and Evans, O. (2022). TruthfulQA: Measuring how models mimic human falsehoods. In Muresan, S., Nakov, P., and Villavicencio, A., editors, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Liu, B., Schlegel, V., Batista-navarro, R., and Ananiadou, S. (2023a). Entity coreference and co-occurrence aware argument mining from biomedical literature. In Strube, M., Braud, C., Hardmeier, C., Li, J. J., Loaiciga, S., and Zeldes, A., editors, Proceedings of the 4th Workshop on Computational Approaches to Discourse (CODI 2023), pages 54–60, Toronto, Canada. Association for Computational Linguistics.

Liu, F., Zettlemoyer, L., and Eisenstein, J. (2019a). The referential reader: A recurrent entity network for anaphora resolution. In Korhonen, A., Traum, D., and Màrquez, L., editors, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5918–5925, Florence, Italy. Association for Computational Linguistics.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. (2024). Lost in the middle: How language models use long contexts. Transactions of the Association for Computational Linguistics, 12:157–173.

Liu, S., Zhang, H., Wang, H., Song, K., Roth, D., and Yu, D. (2023b). Open-domain event graph induction for mitigating framing bias.

Liu, T., Jiang, Y. E., Monath, N., Cotterell, R., and Sachan, M. (2022). Autoregressive structured prediction with language models. In Goldberg, Y., Kozareva, Z.,

and Zhang, Y., editors, Findings of the Association for Computational Linguistics: EMNLP 2022, pages 993–1005, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., and Zhu, C. (2023c). G-eval: NLG evaluation using gpt-4 with better human alignment. In Bouamor, H., Pino, J., and Bali, K., editors, Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 2511–2522, Singapore. Association for Computational Linguistics.

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.

Logan IV, R. L., McCallum, A., Singh, S., and Bikel, D. (2021). Benchmarking scalable methods for streaming cross document entity coreference. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, 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 4717–4731, Online. Association for Computational Linguistics.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In International Conference on Learning Representations.

Lu, J. and Ng, V. (2020). Conundrums in entity coreference resolution: Making sense of the state of the art. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6620–6631, Online. Association for Computational Linguistics.

Luo, X. (2005). On coreference resolution performance metrics. In Mooney, R., Brew, C., Chien, L.-F., and Kirchhoff, K., editors, Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, pages 25–32, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Mahon, L. and Lapata, M. (2024). A modular approach for multimodal summarization of TV shows. In Ku, L.-W., Martins, A., and Srikumar, V., editors, Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1:

Long Papers), pages 8272–8291, Bangkok, Thailand. Association for Computational Linguistics.

Markert, K., Hou, Y., and Strube, M. (2012). Collective classification for fine-grained information status. In Li, H., Lin, C.-Y., Osborne, M., Lee, G. G., and Park, J. C., editors, Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 795–804, Jeju Island, Korea. Association for Computational Linguistics.

Martin, S., Poddar, S., and Upasani, K. (2020). MuDoCo: Corpus for multidomain coreference resolution and referring expression generation. In Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, Proceedings of the Twelfth Language Resources and Evaluation Conference, pages 104–111, Marseille, France. European Language Resources Association.

Martinelli, G., Barba, E., and Navigli, R. (2024). Maverick: Efficient and accurate coreference resolution defying recent trends. In Ku, L.-W., Martins, A., and Srikumar, V., editors, Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 13380–13394, Bangkok, Thailand. Association for Computational Linguistics.

Maynez, J., Narayan, S., Bohnet, B., and McDonald, R. (2020). On faithfulness and factuality in abstractive summarization. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1906–1919, Online. Association for Computational Linguistics.

McCoy, K. E. and Strube, M. (1999). Generating anaphoric expressions: Pronoun or definite description? In The Relation of Discourse/Dialogue Structure and Reference.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings.

Mitkov, R. (2002). Anaphora Resolution. Routledge.

Murakhovs'ka, L., Wu, C.-S., Laban, P., Niu, T., Liu, W., and Xiong, C. (2022). MixQG: Neural question generation with mixed answer types. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V., editors, Findings of the Association for Computational Linguistics: NAACL 2022, pages 1486–1497, Seattle, United States. Association for Computational Linguistics.

Narayan, S., Cohen, S. B., and Lapata, M. (2018). Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J., editors, Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Narayan, S., Maynez, J., Amplayo, R. K., Ganchev, K., Louis, A., Huot, F., Sandholm, A., Das, D., and Lapata, M. (2023). Conditional generation with a question-answering blueprint. Transactions of the Association for Computational Linguistics, 11:974–996.

Narayan, S., Zhao, Y., Maynez, J., Simões, G., Nikolaev, V., and McDonald, R. (2021). Planning with learned entity prompts for abstractive summarization. Transactions of the Association for Computational Linguistics, 9:1475–1492.

Nawrot, P., Łańcucki, A., Chochowski, M., Tarjan, D., and Ponti, E. (2024). Dynamic memory compression: Retrofitting LLMs for accelerated inference. In Forty-first International Conference on Machine Learning.

Nissim, M., Dingare, S., Carletta, J., and Steedman, M. (2004). An annotation scheme for information status in dialogue. In Lino, M. T., Xavier, M. F., Ferreira, F., Costa, R., and Silva, R., editors, Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04), Lisbon, Portugal. European Language Resources Association (ELRA).

Noreen, E. W. (1989). Computer-Intensive Methods for Testing Hypotheses: An Introduction. Wiley, New York.

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T.,

Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Łukasz Kaiser, Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, J. H., Kiros, J., Knight, M., Kokotajlo, D., Łukasz Kondraciuk, Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H. P., Michael, Pokorny, Pokrass, M., Pong, V. H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M. B., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S.,

Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. (2024). Gpt-4 technical report.

Otmazgin, S., Cattan, A., and Goldberg, Y. (2023). LingMess: Linguistically informed multi expert scorers for coreference resolution. In Vlachos, A. and Augenstein, I., editors, Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 2752–2760, Dubrovnik, Croatia. Association for Computational Linguistics.

Pagnoni, A., Balachandran, V., and Tsvetkov, Y. (2021). Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4812–4829, Online. Association for Computational Linguistics.

Pagnoni, A., Fabbri, A., Kryscinski, W., and Wu, C.-S. (2023). Socratic pretraining: Question-driven pretraining for controllable summarization. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 12737–12755, Toronto, Canada. Association for Computational Linguistics.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, Advances in Neural Information Processing Systems 32, pages 8024–8035. Curran Associates, Inc.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In Moschitti, A., Pang, B., and Daelemans, W., editors, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In Walker, M., Ji, H., and Stent,

A., editors, 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.

Poesio, M., Grishina, Y., Kolhatkar, V., Moosavi, N., Roesiger, I., Roussel, A., Simonjetz, F., Uma, A., Uryupina, O., Yu, J., and Zinsmeister, H. (2018). Anaphora resolution with the ARRAU corpus. In Poesio, M., Ng, V., and Ogrodniczuk, M., editors, Proceedings of the First Workshop on Computational Models of Reference, Anaphora and Coreference, pages 11–22, New Orleans, Louisiana. Association for Computational Linguistics.

Poesio, M., Mehta, R., Maroudas, A., and Hitzeman, J. (2004a). Learning to resolve bridging references. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04), pages 143–150, Barcelona, Spain.

Poesio, M., Stevenson, R., Di Eugenio, B., and Hitzeman, J. (2004b). Centering: A parametric theory and its instantiations. Computational Linguistics, 30(3):309–363.

Polanyi, L. (1988). A formal model of the structure of discourse. Journal of Pragmatics, 12(5):601–638.

Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., and Zhang, Y. (2012). CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In Pradhan, S., Moschitti, A., and Xue, N., editors, Joint Conference on EMNLP and CoNLL - Shared Task, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Prince, E. F. (1981). Toward a taxonomy of given-new information. In Cole, P., editor, Syntax and semantics: Vol. 14. Radical Pragmatics, pages 223–255. Academic Press, New York.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. OpenAI blog.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. OpenAI blog.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res., 21(1).

Rajpurkar, P., Jia, R., and Liang, P. (2018). Know what you don't know: Unanswerable questions for SQuAD. In Gurevych, I. and Miyao, Y., editors, Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In Su, J., Duh, K., and Carreras, X., editors, Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Richardson, M., Burges, C. J., and Renshaw, E. (2013). MCTest: A challenge dataset for the open-domain machine comprehension of text. In Yarowsky, D., Baldwin, T., Korhonen, A., Livescu, K., and Bethard, S., editors, Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 193–203, Seattle, Washington, USA. Association for Computational Linguistics.

Roberts, C. (2012). Information structure in discourse: Towards an integrated formal theory of pragmatics. Semantics and Pragmatics, 5(6):1–69.

Sanh, V., Webson, A., Raffel, C., Bach, S., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Raja, A., Dey, M., Bari, M. S., Xu, C., Thakker, U., Sharma, S. S., Szczechla, E., Kim, T., Chhablani, G., Nayak, N., Datta, D., Chang, J., Jiang, M. T.-J., Wang, H., Manica, M., Shen, S., Yong, Z. X., Pandey, H., Bawden, R., Wang, T., Neeraj, T., Rozen, J., Sharma, A., Santilli, A., Fevry, T., Fries, J. A., Teehan, R., Scao, T. L., Biderman, S., Gao, L., Wolf, T., and Rush, A. M. (2022a). Multitask prompted training enables zero-shot task generalization. In International Conference on Learning Representations.

Sanh, V., Webson, A., Raffel, C., Bach, S., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Raja, A., Dey, M., Bari, M. S., Xu, C., Thakker, U., Sharma, S. S., Szczechla, E., Kim, T., Chhablani, G., Nayak, N., Datta, D., Chang, J., Jiang, M. T.-J., Wang, H., Manica, M., Shen, S., Yong, Z. X., Pandey, H., Bawden, R., Wang,

T., Neeraj, T., Rozen, J., Sharma, A., Santilli, A., Fevry, T., Fries, J. A., Teehan, R., Scao, T. L., Biderman, S., Gao, L., Wolf, T., and Rush, A. M. (2022b). Multitask prompted training enables zero-shot task generalization. In International Conference on Learning Representations.

Scialom, T., Dray, P.-A., Lamprier, S., Piwowarski, B., Staiano, J., Wang, A., and Gallinari, P. (2021). QuestEval: Summarization asks for fact-based evaluation. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 6594–6604, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shaham, U., Segal, E., Ivgi, M., Efrat, A., Yoran, O., Haviv, A., Gupta, A., Xiong, W., Geva, M., Berant, J., and Levy, O. (2022). SCROLLS: Standardized CompaRison over long language sequences. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 12007–12021, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shao, Y., Jiang, Y., Kanell, T., Xu, P., Khattab, O., and Lam, M. (2024). Assisting in writing Wikipedia-like articles from scratch with large language models. In Duh, K., Gomez, H., and Bethard, S., editors, Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 6252–6278, Mexico City, Mexico. Association for Computational Linguistics.

Song, H., Su, H., Shalyminov, I., Cai, J., and Mansour, S. (2024). FineSurE: Fine-grained summarization evaluation using LLMs. In Ku, L.-W., Martins, A., and Srikumar, V., editors, Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 906–922, Bangkok, Thailand. Association for Computational Linguistics.

Sotudeh, S. and Goharian, N. (2024). Learning to rank salient content for query-focused summarization. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 15038–15048, Miami, Florida, USA. Association for Computational Linguistics.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, Advances in Neural Information Processing Systems, volume 27. Curran Associates, Inc.

Thirukovalluru, R., Monath, N., Shridhar, K., Zaheer, M., Sachan, M., and McCallum, A. (2021). Scaling within document coreference to long texts. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 3921–3931, Online. Association for Computational Linguistics.

Toshniwal, S., Ettinger, A., Gimpel, K., and Livescu, K. (2020a). PeTra: A Sparsely Supervised Memory Model for People Tracking. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 5415–5428, Online. Association for Computational Linguistics.

Toshniwal, S., Wiseman, S., Ettinger, A., Livescu, K., and Gimpel, K. (2020b). Learning to Ignore: Long Document Coreference with Bounded Memory Neural Networks. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 8519–8526, Online. Association for Computational Linguistics.

Toshniwal, S., Xia, P., Wiseman, S., Livescu, K., and Gimpel, K. (2021). On generalization in coreference resolution. In Ogrodniczuk, M., Pradhan, S., Poesio, M., Grishina, Y., and Ng, V., editors, Proceedings of the Fourth Workshop on Computational Models of Reference, Anaphora and Coreference, pages 111–120, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Urbanek, J., Fan, A., Karamcheti, S., Jain, S., Humeau, S., Dinan, E., Rocktäschel, T., Kiela, D., Szlam, A., and Weston, J. (2019). Learning to speak and act in a fantasy text adventure game. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, 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 673–683, Hong Kong, China. Association for Computational Linguistics.

Uryupina, O., Artstein, R., Bristot, A., Cavicchio, F., Rodriguez, K., and Poesio, M. (2016). ARRAU: Linguistically-motivated annotation of anaphoric descriptions. In

Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pages 2058–2062, Portorož, Slovenia. European Language Resources Association (ELRA).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc.

Vilain, M., Burger, J., Aberdeen, J., Connolly, D., and Hirschman, L. (1995). A model-theoretic coreference scoring scheme. In Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995.

Wang, A., Pang, R. Y., Chen, A., Phang, J., and Bowman, S. R. (2022). SQuALITY: Building a long-document summarization dataset the hard way. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 1139–1156, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wang, B., Lu, W., Wang, Y., and Jin, H. (2018). A neural transition-based model for nested mention recognition. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J., editors, Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1011–1017, Brussels, Belgium. Association for Computational Linguistics.

Wang, C., Liu, X., Yue, Y., Tang, X., Zhang, T., Jiayang, C., Yao, Y., Gao, W., Hu, X., Qi, Z., Wang, Y., Yang, L., Wang, J., Xie, X., Zhang, Z., and Zhang, Y. (2023). Survey on factuality in large language models: Knowledge, retrieval and domain-specificity.

Wang, X., Shi, W., Kim, R., Oh, Y., Yang, S., Zhang, J., and Yu, Z. (2019). Persuasion for good: Towards a personalized persuasive dialogue system for social good. In Korhonen, A., Traum, D., and Màrquez, L., editors, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5635–5649, Florence, Italy. Association for Computational Linguistics.

Webber, B. (1978). A Formal Approach to Discourse Anaphora. PhD thesis, Harvard University. Published by Garland, New York, 1979.

Webster, K. and Curran, J. R. (2014). Limited memory incremental coreference resolution. In Tsujii, J. and Hajic, J., editors, Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 2129–2139, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Webster, K., Recasens, M., Axelrod, V., and Baldridge, J. (2018). Mind the GAP: A balanced corpus of gendered ambiguous pronouns. Transactions of the Association for Computational Linguistics, 6:605–617.

Weischedel, R., Marcus, M., Palmer, M., Hovy, E., Belvin, R., Pradhan, S., and Ramshaw, L. (2010). Ontonotes: A large training corpus for enhanced processing. In Handbook of Natural Language Processing and Machine Translation. Springer, New York, NY.

Weischedel, R., Palmer, M., Marcus, M., Hovy, E., Pradhan, S., Ramshaw, L., Xue, N., Taylor, A., Kaufman, J., Franchini, M., El-Bachouti, M., Belvin, R., and Houston, A. (2013). Ontonotes release 5.0. LDC2013T19, Philadelphia, Penn.: Linguistic Data Consortium.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2019). Huggingface's transformers: State-of-the-art natural language processing.

Wu, W., Wang, F., Yuan, A., Wu, F., and Li, J. (2020). CorefQA: Coreference resolution as query-based span prediction. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6953–6963, Online. Association for Computational Linguistics.

Xia, P., Sedoc, J., and Van Durme, B. (2020). Incremental neural coreference resolution in constant memory. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 8617–8624, Online. Association for Computational Linguistics.

Xiong, W., Gupta, A., Toshniwal, S., Mehdad, Y., and Yih, S. (2023). Adapting pre-trained text-to-text models for long text sequences. In Bouamor, H., Pino, J., and Bali, K., editors, Findings of the Association for Computational Linguistics: EMNLP 2023, pages 5566–5578, Singapore. Association for Computational Linguistics.

Xu, L. and Choi, J. D. (2020). Revealing the myth of higher-order inference in coreference resolution. In Webber, B., Cohn, T., He, Y., and Liu, Y., editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 8527–8533, Online. Association for Computational Linguistics.

Xu, L. and Choi, J. D. (2022). Online coreference resolution for dialogue processing: Improving mention-linking on real-time conversations. In Nastase, V., Pavlick, E., Pilehvar, M. T., Camacho-Collados, J., and Raganato, A., editors, Proceedings of the 11th Joint Conference on Lexical and Computational Semantics, pages 341–347, Seattle, Washington. Association for Computational Linguistics.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (2021). mT5: A massively multilingual pre-trained text-to-text transformer. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., and Zhou, Y., editors, Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 483–498, Online. Association for Computational Linguistics.

Yan, K., Liu, W., Xie, S., and Peng, Y. (2024). Graph-based event schema induction in open-domain corpus. PeerJ Computer Science, 10:e2155.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc.

Yu, J., Khosla, S., Manuvinakurike, R., Levin, L., Ng, V., Poesio, M., Strube, M., and Rosé, C. (2022a). The CODI-CRAC 2022 shared task on anaphora, bridging, and discourse deixis in dialogue. In Yu, J., Khosla, S., Manuvinakurike, R., Levin, L., Ng, V., Poesio, M., Strube, M., and Rose, C., editors, Proceedings of the CODI-CRAC

2022 Shared Task on Anaphora, Bridging, and Discourse Deixis in Dialogue, pages 1–14, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Yu, J., Khosla, S., Moosavi, N. S., Paun, S., Pradhan, S., and Poesio, M. (2022b). The universal anaphora scorer. In Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Odijk, J., and Piperidis, S., editors, Proceedings of the Thirteenth Language Resources and Evaluation Conference, pages 4873–4883, Marseille, France. European Language Resources Association.

Yu, J., Uma, A., and Poesio, M. (2020). A cluster ranking model for full anaphora resolution. In Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, Proceedings of the Twelfth Language Resources and Evaluation Conference, pages 11–20, Marseille, France. European Language Resources Association.

Zha, Y., Yang, Y., Li, R., and Hu, Z. (2023). AlignScore: Evaluating factual consistency with a unified alignment function. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Zhang*, T., Kishore*, V., Wu*, F., Weinberger, K. Q., and Artzi, Y. (2020). Bertscore: Evaluating text generation with bert. In International Conference on Learning Representations.

Zhang, W., Wiseman, S., and Stratos, K. (2023). Seq2seq is all you need for coreference resolution. In Bouamor, H., Pino, J., and Bali, K., editors, Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 11493–11504, Singapore. Association for Computational Linguistics.

Appendix A

Supplementary Material for Chapter 2

A.1 FineSurE Faithfulness Prompt

The prompt used in the FineSurE metrics for faithfulness evaluation in shown in Figure A.1.

You will receive a transcript followed by a corresponding summary. Your task is to assess the factuality of each summary sentence across nine categories:
* no error: the statement aligns explicitly with the content of the transcript and is factually consistent with it.
* out-of-context error: the statement contains information not present in the transcript.
* entity error: the primary arguments (or their attributes) of the predicate are wrong.
* predicate error: the predicate in the summary statement is inconsistent with the transcript.
* circumstantial error: the additional information (like location or time) specifying the circumstance around a predicate is wrong.
* grammatical error: the grammar of the sentence is so wrong that it becomes meaningless.
* coreference error: a pronoun or reference with wrong or non- existing antecedent.
* linking error: error in how multiple statements are linked together in the discourse (for example temporal ordering or causal link).
* other error: the statement contains any factuality error which is not defined here.

Instruction:
First, compare each summary sentence with the transcript.
Second, provide a single sentence explaining which factuality error the sentence has.
Third, answer the classified error category for each sentence in the summary.
Provide your answer in JSON format. The answer should be a list of dictionaries whose keys are "sentence", "reason", and "category":
["sentence": "first sentence", "reason": "your reason", "category": "no error", "sentence": "second sentence", "reason": "your reason", "category": "out-of-context error", "sentence": "third sentence", "reason": "your reason", "category": "entity error",]

Transcript:
{input text}

Summary with N sentences:
{summary sentence 1}
{summary sentence 2}
...
{summary sentence N}

Figure A.1: The FineSurE prompt for faithfulness evaluation in summarization.

Appendix B

Supplementary Material for Chapter 3

B.1 Full OntoNotes Results

The full precision and recall scores on OntoNotes can be found in Table B.1.

B.2 Hyperparameters and Other Model Details

The main hyperparameters are listed in Table B.2.

The bottom four rows refer to the maximum number of learned embeddings we use for each feature. Additionally:

- The top performing Part-Inc model uses 20 sentences as active input, with the remainder as memory (up to 512 tokens total).
- During training, the Sent-Inc model accumulates gradients after every 32 sentences to ensure that the memory used does not exceed capacity.

Our implementation is based off of Xu and Choi (2020)'s codebase. We find their model hyperparameters are already extremely well-tuned, and so we do not explore further hyperparameter tuning for these cases. Regarding new hyperparameters introduced in this work, we follow previous work in choosing sensible values. For example, the StackLSTM and Action History LSTM hidden sizes follow Dyer et al. (2015)'s recommendations.

We train all models using NVIDIA Tesla V100 16 GB cards on an HPC cluster. Training convergence takes approximately 24 hours. Both Sent-Inc and Part-Inc models contain around 140 million parameters.

Enc. Size	Model	*MUC* R	*MUC* P	*MUC* F1	B^3 R	B^3 P	B^3 F1	$CEAF_{\phi_4}$ R	$CEAF_{\phi_4}$ P	$CEAF_{\phi_4}$ F1	Avg. F1
Large	SpanBERT	84.8	85.8	85.3	77.9	78.3	78.1	74.2	76.4	75.3	79.6
	CorefQA+SP	87.4	88.6	88.0	82.0	82.4	82.2	78.3	79.9	79.1	83.1
	s2f+Longformer	85.1	86.5	85.8	77.9	80.3	79.1	75.4	76.8	76.1	80.3
	s2e + se_ct	85.3	87.2	86.3	78.6	80.7	79.6	75.2	78.2	76.7	80.9
Base	SpanBERT	83.1	84.3	83.7	75.3	76.2	75.8	71.2	74.6	72.9	77.4
	CorefQA+SP	87.4	85.2	86.3	76.5	78.7	77.6	75.6	76.0	75.8	79.9
Base	longdoc	83.3	83.0	83.2	75.5	74.1	74.8	70.1	72.8	71.4	76.4
	ICoref	83.1	84.2	83.6	74.3	75.8	75.0	71.7	73.3	72.5	77.0
	Part-Inc (Ours)	83.7	82.1	82.9	75.9	73.0	74.4	68.8	74.5	71.6	76.3
Base	ICoref-*inc*	74.0	79.7	76.7	58.6	70.6	64.0	63.7	63.1	63.4	68.0
	Sent-Inc (Ours)	78.1	79.4	78.8	68.9	68.3	68.6	55.8	71.2	62.5	70.0

Table B.1: Main results on the OntoNotes 5.0 test set with the CoNLL 2012 Shared Task metrics and the average F1 (the CoNLL F1 score). The top four systems are not directly comparable to ours, since they train with a 'Large' encoder (either SpanBERT or Longformer (Beltagy et al., 2020)). Note that scores for Xia et al. (2020) and Toshniwal et al. (2020b) differ from their reported results because we re-train them with SpanBERT-*base* instead of *large*. Results for Joshi et al. (2020)-base are taken from Lu and Ng (2020), which report the SpanBERT-*base* results.

Hyperparameter	Value
Encoder Learning Rate	2e-5
Task Learning Rate	1e-4
Adam Eps	1e-6
Adam Weight Decay	1e-2
Gradient Clipping Norm	1
Dropout Rate	0.3
StackLSTM Hidden Size	200
Action History Hidden Size	30
f_M Hidden Size	1000
f_C Hidden Size	3000
New Cell Threshold (α)	0.0
ϕ_C Max Entity Count	10
ϕ_C Max Mention Distance	10
ϕ_M Max Span Width	30
Max Speaker Number	20

Table B.2: Hyperparameters used during training.

Appendix C

Supplementary Material for Chapter 4

C.1 Full OntoNotes Results

The full precision and recall scores on OntoNotes can be found in Table C.1.

C.2 NER-Augmented Inference Additional Information

In our experiments with NER-augmented inference, we explored using several different sets of NER categories. We found that certain NER types in OntoNotes, such as ordinal and cardinal numbers, rarely or never intersect with mentions in the coreference annotations. After analyzing the degree of overlap between each NER category and mentions in the coreference annotations, we found using the GPE, PERSON and ORG tags in the first oracle experiment (with Forced Mention Start), and a set of ten categories in the second experiment, resulted in the best scores. The ten categories are: GPE, ORG, PERSON, LAW, FAC, LANGUAGE, EVENT, PRODUCT, LOC, DATE and WORK_OF_ART.

The full precision and recall scores for the NER-Augmented Inference experiments are shown in Table C.2.

	MUC			B^3			$CEAF_{\phi_4}$			Avg.
Model	P	R	F1	P	R	F1	P	R	F1	F1
GPT-4	73.9	73.5	73.7	60.8	64.7	62.7	49.3	55.7	52.3	62.9
SpanBERT	85.8	84.8	85.3	78.3	77.9	78.1	76.4	74.2	75.3	79.6
CorefQA	88.6	87.4	88.0	82.4	82.0	82.2	79.9	78.3	79.1	83.1
s2e	86.5	85.1	85.8	80.3	77.9	79.1	76.8	75.4	76.1	80.3
wl-coref	84.9	87.9	86.3	77.4	82.6	79.9	76.1	77.1	76.6	81.0
ASP	86.1	88.4	87.2	80.2	83.2	81.7	78.9	78.3	78.6	82.5
Link-Append	87.4	88.3	87.8	81.8	83.4	82.6	79.1	79.9	79.5	83.3
Copy+$T0_{pp}$	86.1	89.2	87.6	80.6	84.3	82.4	78.9	80.1	79.5	83.2
Token Action + $T0_{3B}$	85.9	88.6	87.2	79.6	83.5	81.5	78.9	78.0	78.5	82.4
Token Action, Non-Inc.	86.1	87.9	87.0	79.8	82.2	81.0	79.1	77.3	78.2	82.0
Full-Prefix Incremental	86.7	84.3	85.5	80.5	77.5	79.0	78.9	70.1	74.3	79.6
Model-based, C=0	86.7	82.7	84.6	79.8	74.8	77.3	78.7	66.9	72.3	78.1
Model-based, C=50	86.5	83.3	84.8	79.8	76.2	78.0	80.0	67.6	73.3	78.7
Model-based, C=100	87.0	83.4	85.1	80.7	75.8	78.2	79.6	68.1	73.4	78.9
Model-based, C=200	86.8	83.3	85.0	80.4	76.2	78.2	80.0	67.9	73.5	78.9

Table C.1: Results on the OntoNotes test set. The bottom section shows our proposed methods.

	MUC			B^3			$CEAF_{\phi_4}$			Avg.
Model	P	R	F1	P	R	F1	P	R	F1	F1
Non-Incremental	86.5	87.1	86.8	81.2	82.2	81.7	79.5	77.9	78.7	82.4
Incremental	86.6	83.3	84.9	80.6	77.1	78.8	78.9	70.0	74.2	79.3
+ NER Force Men. Start	85.4	84.7	85.0	79.2	78.9	79.0	77.1	72.3	74.6	79.6
+ NER Exact Str. Match	86.0	84.3	85.2	80.0	78.3	79.1	78.2	71.7	74.8	79.7
+ Pseudosingleton, 30K	86.5	84.0	85.2	80.9	77.9	79.4	77.6	72.8	75.2	79.9
+ Pseudosingleton, 60K	85.7	85.3	85.5	79.8	79.9	79.9	77.4	74.8	76.1	80.5

Table C.2: Results on the OntoNotes validation set with different methods for augmenting the dataset with singleton mentions.

Appendix D

Supplementary Material for Chapter 5

D.1 Examples

An example of a synthetic plan from Sonnet 3.5, alongside the associated gold summary is shown in Figure D.1. An example of a coarse plan with citations, used for generating QA pairs, is shown in Figure D.2.

D.2 Coarse Planning Prompt

The prompt used by Sonnet 3.5 to generate plans is shown in Figure D.3.

D.3 Phi-3.5-mini Prompts

The Phi-3.5-mini planning and summarization prompts (Baseline, E2E, Multi-Task) are shown in Figure D.5.

D.4 Claude Baseline Prompt

The best performing summarization prompt of the 16 summarization prompts we tried is listed in Figure D.4.

D.5 Human Evaluation Details

Our evaluation rubric can be found in Figure D.6, and we provide an example of the type of claims extracted in Figure D.7.

Captain Linden and his lieutenant "Split" Campbell make up the first manned expedition from Earth to this particular planet, aiming to investigate a large silver river on its surface. The seemingly-endless silvery strip that traveled the planet's surface was unidentifiable as of yet. They see the river-like thing early on, but Campbell spots a humanoid through his telescope–this being is much like a human man, including the fact that he wore clothing. Captain Linden decides it's time for introductions, as if he senses he can trust this being, but they watch as a female and then many other people join the first man on the surface, seemingly coming out of an underground city. Linden and Campbell think their ship is out of sight, and watch a ritual that the man is performing to the setting sun. The crowd of people continues to increase, and Linden notices that the landscape is moving: trees are shifting in the ground. He and Campbell stay in the ship and observe the various types of clothing and the ritual itself, as well as the moving trees which seemed to be moving to attack the people. They are indeed warriors starting an attack, and started swinging weapons. Linden tells Campbell to start the siren on their ship to scare away the attackers, and the first man they'd seen, presumably the leader, starts towards the ship. Once they are close enough, it is obvious that the humanoids don't have eyebrows or eye lashes. Captain Linden hands the leader a medallion that plays a song, as a token of friendship. Tomboldo, the leader, starts a round of introductions through a lot of gesturing. Linden hopes to learn about the Serpent River through the people to understand its cultural significance, and these people start to ask about the siren noises. The warriors attack again and panic ensues, pushing the humans to use weapons this time. Gravgak, the guard who had been escorting the humans, is knocked down. As Linden tries to tend to him, Gravgak knocks him out with his club. Linden is unconscious for a few weeks, and Vauna, Tomboldo's daughter, spends a lot of time by the Captian's side. Linden reminds Campbell that they weren't allowed to marry anyone from this planet, but mostly in an effort to warn himself to be careful around Vauna. He learns that these people are called the Benzendellas. Tomboldo is baffled by the technology that the humans have, but Linden is not able to communicate his questions about the Serpent River. He sees Gravgak, who apologizes for the accidental injury, but from Vauna's reaction Linden is not sure if he is telling the truth. Gravgak insists on talking to Vauna in private, but Vauna's father calls them back. It is Tomboldo's thanks to the humans that gives a glimpse into the meaning of the Serpent River: he says the humans will ride with them on the rope of life, which they call Kao-Wagwattl.

1. Captain Linden and Campbell lead first manned expedition to planet.
2. They observe a silver river-like feature on the planet's surface.
3. Campbell spots humanoid beings through his telescope.
4. They witness people emerging from an underground city.
5. Linden and Campbell observe a ritual and moving trees.
6. Tree-like warriors attack the humanoid people.
7. Linden uses ship's siren to scare away attackers.
8. Humans make first contact with the planet's inhabitants.
9. Warriors attack again, prompting humans to use weapons.
10. Gravgak knocks Linden unconscious.
11. Linden recovers, forms connection with Vauna.
    Q: How long is Linden unconscious? A: a few weeks
12. Humans learn about the Benzendellas and their technology.
    Q: What does Linden not know about? A: the Serpent River
13. Gravgak apologizes for injuring Linden.
14. Tomboldo reveals the Serpent River is called Kao-Wagwattl.

Figure D.1: An example reference summary from SQuALITY (top), with the Sonnet 3.5 synthetic plan and QA pairs (bottom).

Captain Linden and his lieutenant “Split” Campbell make up the first manned expedition from Earth to this particular planet, aiming to investigate a large silver river on its surface. [1] The seemingly-endless silvery strip that traveled the planet’s surface was unidentifiable as of yet. [2] They see the river-like thing early on, but Campbell spots a humanoid through his telescope–this being is much like a human man, including the fact that he wore clothing. [3] Captain Linden decides it’s time for introductions, as if he senses he can trust this being, but they watch as a female and then many other people join the first man on the surface, seemingly coming out of an underground city. [4] Linden and Campbell think their ship is out of sight, and watch a ritual that the man is performing to the setting sun. [5] The crowd of people continues to increase, and Linden notices that the landscape is moving: trees are shifting in the ground. [6] He and Campbell stay in the ship and observe the various types of clothing and the ritual itself, as well as the moving trees which seemed to be moving to attack the people. [7] They are indeed warriors starting an attack, and started swinging weapons. [8] Linden tells Campbell to start the siren on their ship to scare away the attackers, and the first man they’d seen, presumably the leader, starts towards the ship. [9] Once they are close enough, it is obvious that the humanoids don’t have eyebrows or eye lashes. [10] Captain Linden hands the leader a medallion that plays a song, as a token of friendship. [11] Tomboldo, the leader, starts a round of introductions through a lot of gesturing. [12] Linden hopes to learn about the Serpent River through the people to understand its cultural significance, and these people start to ask about the siren noises. [13] The warriors attack again and panic ensues, pushing the humans to use weapons this time. [14] Gravgak, the guard who had been escorting the humans, is knocked down. [15] As Linden tries to tend to him, Gravgak knocks him out with his club. [16] Linden is unconscious for a few weeks, and Vauna, Tomboldo’s daughter, spends a lot of time by the Captian’s side. [17] Linden reminds Campbell that they weren’t allowed to marry anyone from this planet, but mostly in an effort to warn himself to be careful around Vauna. [18] He learns that these people are called the Benzendellas. [19] Tomboldo is baffled by the technology that the humans have, but Linden is not able to communicate his questions about the Serpent River. [20] He sees Gravgak, who apologizes for the accidental injury, but from Vauna’s reaction Linden is not sure if he is telling the truth. [21] Gravgak insists on talking to Vauna in private, but Vauna’s father calls them back. [22] It is Tomboldo’s thanks to the humans that gives a glimpse into the meaning of the Serpent River: he says the humans will ride with them on the rope of life, which they call Kao-Wagwattl. [23]

1. Captain Linden and Campbell lead first manned expedition to planet. [1]
2. They observe a silver river-like feature on the planet’s surface. [1, 2]
3. Campbell spots humanoid beings through his telescope. [3]
4. They witness people emerging from an underground city. [4]
5. Linden and Campbell observe a ritual and moving trees. [5, 6, 7]
6. Tree-like warriors attack the humanoid people. [8]
7. Linden uses ship’s siren to scare away attackers. [9]
8. Humans make first contact with the planet’s inhabitants. [10, 11, 12]
9. Warriors attack again, prompting humans to use weapons. [14]
10. Gravgak knocks Linden unconscious. [15, 16]
11. Linden recovers, forms connection with Vauna. [17, 18]
12. Humans learn about the Benzendellas and their technology. [19, 20]
13. Gravgak apologizes for injuring Linden. [21]
14. Tomboldo reveals the Serpent River is called Kao-Wagwattl. [23]

Figure D.2: An example reference summary from SQuALITY with sentence markers in brackets (top), with the Sonnet 3.5 synthetic plan with citations in brackets (bottom). This version of the coarse plans is passed to the QA pipeline described in order to extract QA pairs.

Human: Given a text, with enumerated sentences, devise a plan for a summary based on major events in the text. Each plan point should be a simple, short sentence (3-10 words), followed by references to all relevant sentences that can be used to validate the information it contains.
Here are some examples:
<example>
Text:
Christina Aguilera had a good Valentine's Day: She announced yesterday that she's engaged to Matt Rutler, her boyfriend of three years, Radar reports. [1] Aguilera, 33, posted a photo to Facebook of her and Rutler holding hands on the beach—and her hand is sporting a huge ring. [2] "He asked and I said..." reads the caption. [3] Aguilera was previously married to Jordan Bratman, with whom she has a 6-year-old son. [4] (As they say, the couple who gets arrested together stays together.) [5]
Plan:
1. Christina Aguilera had a good Valentine's Day. [1] 2. She announced her engagement to Matt Rutler. [1] 3. She posted a photo of herself and Rutler holding hands. [2] 4. She was previously married to Jordan Bratman. [4] 5. She has a son with Bratman. [4] 6. The couple was arrested together previously. [5]
</example>
<example>
Text:
The US stands by the "one-China" policy, but that doesn't mean it can't sell weapons directly to Taiwan, citing the Taiwan Relations Act to ensure Taiwan can adequately defend itself—and China isn't happy about it. [1] The Obama administration announced a $1.8 billion arms package sale to Congress on Wednesday, Reuters reports, including guided-missile frigates, anti-tank missiles, Amphibious Assault Vehicles, and $416 million worth of guns, ammo, and other supplies. [2] The announcement came amid reports that the US had stalled the sale to avoid hearing about it from China, which still claims Taiwan as a territory, per the Wall Street Journal. [3] Reuters notes the sale comes as US-China relations simmer over the latter's man-made islands in the South China Sea and US patrols in those waters. [4] China notes it's going to sanction the companies involved in the sale (including Lockheed Martin and Raytheon), with a foreign ministry official telling Xinhua that the sale flouts international rules and "severely" damages China's sovereignty. [5] "China's government and companies will not carry out cooperation and commercial dealings with these types of companies," a ministry spokesman says. [6] A Pentagon spokesman gave the equivalent of an eyeroll Wednesday, per the New York Times, noting, "The Chinese can react to this as they see fit. [7] It's a clear-eyed, sober view of an assessment of Taiwan's defense needs. [8] There's no need for it to have any derogatory effect on our relationship with China." [9] Meanwhile, the AP notes that China has issued similar threats before, with "no evidence they've had any meaningful effect." (All this despite a lengthy handshake last month.) [10]
Plan:
1. US announced an arms package sale to Taiwan. [2] 2. China is not happy about it. [1] 3. China threatens to sanction companies involved in the sale. [5] 4. US shrugs off the threat. [9] 5. China has issued similar threats before without any meaningful effect. [10]
</example>
Assistant: Ok. How many plan points do you want me to include?
Human: This will depend on the length of the text. If the text is long you can include many plan points. Make sure each significant event or occurrence is represented in the plan.
Assistant: What are the numbers in the bracket such as [1], [2] etc at the end of each plan point?
Human: Good question. For each plan point, you are required to cite the relevant sentence number which can be used to validate the information contained in the plan point. If multiple sentences need to be cited, then separate the sentence numbers with comma such as [1, 2, 3] or [8, 10].
Assistant: Ok, so the numbers at the end of the plan point correspond to the relevant sentence numbers based on which the plan point was generated.
Human: Yes, that is correct. Please be very careful with the citation. It is very important that you get the citation correct for all of the plan points. Please note again that each sentence in the Text ends with the sentence number such as [1], [2] etc.
Assistant: Ok, I will do my best.
Human: Now it's your turn to write plans. I'll give you the text and you give me the plan with citations for each plan point. Provide your response after "Plan:".
Text:
{}
Assistant:

Figure D.3: Prompt supplied to Sonnet 3.5 to produce plans from summaries. For brevity, we only include two ICL examples here. Our actual prompt contains three ICL examples.

Human: Please summarize the following text (included within <text>and </text>tags) in up to 512 words.
Return the summary within <summary>and </summary>tags.
<text>
{}
</text>
Assistant:

Figure D.4: The prompt used for summarizing documents with Sonnet 3.5.

Baseline:

Generate a summary for the following text. Enclose the summary within <summary> and </summary> tags.
Text:
{}

E2E:

Generate a plan followed by a summary for the following text. Enclose the plan within <plan> and </plan> tags and enclose the summary within <summary> and </summary> tags.
Text:
{}

Multi-Task (for plans):

Generate a plan for the following text. Enclose the plan within <plan> and </plan> tags.
Text:
{}

Multi-Task (for summaries):

Generate a summary for the following text. Enclose the summary within <summary> and </summary> tags.
Text:
{}

Figure D.5: The prompts we used for summarizing documents with Phi-3.

Instructions

1. After reading the story, write a list of "key facts" from the source document. A key fact should be a major narrative point in the story. Feel free to consult the associated reference summaries to refine the list.

2. Write a list of atomic facts from the predicted summary, using the definition from Kim et al. (2024).

3. Then compute each metric as follows:

Coverage: For each key fact in the document, does the key fact appear in the predicted summary? Compute coverage as:

$$\frac{\text{\# of key facts in the source document that appear in the predicted summary}}{\text{\# key facts in the source document}}$$

Faithfulness: For each atomic fact in the predicted summary, is it supported in the source document? Compute faithfulness as:

$$\frac{\text{\# of supported atomic facts in the predicted summary}}{\text{\# of atomic facts in the predicted summary}}$$

Conciseness: For each atomic fact in the predicted summary, is it highly relevant to the story? More concretely, does the atomic fact appear in one of the reference summaries? Compute conciseness as:

$$\frac{\text{\# of atomic facts in the predicted summary that appear in a reference summary}}{\text{\# of atomic facts in the predicted summary}}$$

Grounding: For each plan point in the predicted plan, does the predicted summary contain / refer to it? Compute grounding as:

$$\frac{\text{\# of plan points appearing in the predicted summary}}{\text{\# of plan points}}$$

Figure D.6: Evaluation rubric used for the manual evaluation task.

1. Captain Linden and his lieutenant ""Split"" Campbell are exploring a planet, in particular a large, silver river.
2. They observe a group of human-like beings emerge from underground and prepare to meet them.
3. They observe moving trees, which turn out to be warriors in disguise preparing to attack the former group.
4. Linden and Campbell hit a siren on the ship, startling the attackers into retreating.
5. Linden meets the leader of the aliens, Tomboldo, and presents him a song-playing medallion.
6. Soon, they are attacked again. Linden and Campbell use their capsule bombs to dispell the warriors.
7. Their guard, Gravgak, is injured. After being awoken, Gravgak, possibly accidentally, knocks out Linden with a club.
8. Linden spends several weeks recuperating, tended by Campbell and Tomboldo's beautiful daughter Vauna.
9. While recovering, Linden reminds Campbell that marrying native inhabitants is against their mission's code of conduct.
10. Tomboldo announces to the Benzendella and the humans they will travel on the river serpent, called Kao-Wagwattl.

---

1. Captain Linden and Splitland their ship on a strange planet.
2. They want to learn more about it.
3. They see a long silvery serpent-like object crawling on the surface.
4. Linden and Campbell are the first humans to land on the planet.
5. Linden and Campbell have been sent by EGGWE.
6. They know it is inhabited by humanoid creatures.
7. They know there is a long, cylindrical rope crawling the surface.
8. Linden orders Campbell to take a closer look at the rope.
9. Campbell reports the object is a living creature.
10. The creature is upright and wearing clothes.
11. Linden is excited to see a human-like creature on the planet.
12. Linden orders Campbell to get ready to meet the creature.
13. Linden and Campbell observe a group of natives watching the sunset.
14. The leader is wearing a red sash and headress.
15. Other members of the group are all handsome and musclar.
16. Members of the group are all wearing white fur ornaments for protection.
17. Linden and Campbell notice the trees around the natives are moving.
18. They realize the natives are not aware they are being watched.
19. A group of savage warriors suddently appear and attack the natives.
20. The natives try to defend themselves but are no match.
21. The warriors are armed with crude clubs and whips.
22. The natives rally around their leader.
23. Linden and Campbell decide to intervene to save them.
24. Linden and Campbell use a siren to scare off the warriors.
25. Linden and Campbell descend from the ship to join the natives.
26. The natives invite Linden and Campbell to their city.
27. Gravgak is a guard responsible for keeping an eye on the trees.
28. Campbell is injured when Gravgak accidentally hits him with a club.
29. Linden uses a capsule bomb to stop the attackers.
30. Gravgak is killed.
31. The natives nurse Campbell back to health.

Figure D.7: (Top) Key facts extracted from a document in SQuALITY. (Bottom) Atomic facts from the same document.