Diving Deep into Event Semantics

Zhengzhong Liu

CMU-LTI-24-012
July 29, 2024

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
www.lti.cs.cmu.edu

Thesis Committee:
Teruko Mitamura (Chair), Carnegie Mellon University
Eduard Hovy, Carnegie Mellon University
Taylor Berg-Kirkpatrick, University of California San Diego
Vicent Ng, The University of Texas at Dallas

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
In Language and Information Technologies

Keywords: event,event mention,script,event schema,coreference,implicit arguments,semantics,indirect supervision,quasi-identity,large language models, mechanical interpretation

Abstract

Events are crucial discourse elements in natural language, playing a vital role in semantic understanding due to their complex structures that interconnect various parts of discourse. Events interact with other discourse elements to form diverse structures. Extensive research has been conducted on analyzing them, primarily focusing on frame structures (examining semantic roles such as participants, time, and location) and various forms of anaphora involving multiple events and entities, such as event coreference, event schema (event sequence, script), and ellipsis.

The rich interactive nature of events presents both challenges and opportunities. On one hand, predicting and analyzing event structures can be complex. For example, conducting a standard document level event slot filling, can involve multiple structure prediction tasks (e.g., event mention, coreference, arguments, and schemas). On the other hand, the interactions among these structures can be leveraged to enhance model predictions, or provide lens to study the mechanisms of models. This thesis explores the complexities and benefits of such interactions across different data availability scenarios, developing prediction methods including direct supervised training, crowdsourced event datasets, and studying automatically formed mechanisms related to them.

In the first part, we present empirical results analyzing event semantics with expert-annotated task-specific annotated datasets. We start by introducing methods studying isolated structures, such as event mention prediction, pair-wise event coreference, and event sequencing. We then present approaches to solve problems involving multiple structures, using multi-step or joint learning methods, such as joint coreference and sequencing, slot filling and verb phrase ellipsis.

Recognizing the high cost of scaling expert-annotated datasets, the second part of this thesis explores methods to increase data availability through crowdsourcing and indirect supervision signals. An intriguing outcome of these approaches is their ability to reveal previously unspecified interactions between different textual structures. A key contribution in this area is our LLM360 language model project, which shares intermediate checkpoints throughout a model's training process. We demonstrate the project's utility for interpretability analysis, using the complex anaphora task of Winograd schemas as a case study.

This thesis demonstrates methods to address and utilize the complex nature of events. We find that scaling up data size leads to more iterations of such structures "automatically" appearing during analysis. Looking ahead, this work opens avenues for developing more sophisticated models that better capture relationships between event structures and for exploring large language models' potential in understanding complex event semantics. Additionally, our interpretability analysis, particularly through LLM360, paves the way for investigating how these models process event semantics internally. This research could lead to more transparent and explainable AI systems, advancing our understanding of complex language processing.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Equipped with the strong power of neural network models, modern NLP systems excel particularly in recognizing and extracting information content from text documents, as demonstrated by the prosperity and development in the field of Information Extraction. Yet there still exist many challenges for machines to understand the semantics of human language. In this thesis, we will delve into the semantics from the angle of discourse analysis, focusing on phenomena around anaphora and key discourse elements, particularly events.

Events are crucial discourse elements in natural language, playing a vital role in semantic understanding due to their complex structures that interconnect various parts of discourse. They interact with other discourse elements to form diverse structures, making them essential building blocks of documents and key to the process of document understanding. Textual realization of events and entities serves as the main medium connecting to the underlying world. We believe analyzing how models handle or parse them can deepen our understanding of computational methods for language understanding.

Broadly, event semantics are closely related to many problems in the field of Natural Language Processing, which involves creating structures to connect the information pieces in discourse (e.g., Semantic Role Labeling, Discourse Parsing) and inferring implicit information from the surface text (e.g., ellipsis of verbs, implicit arguments). Events play interesting bridging roles in these structures, providing inference and connection mechanisms among discourse elements.

1.1 Motivation

Understanding event structures in natural language is a fundamental challenge in the field of Natural Language Processing (NLP). The central role of events makes them crucial for improving the performance of NLP systems and serves as a lens through which many interesting linguistic phenomena can be studied. This is still true in the era of LLM, as to be demonstrated in §8.

The motivation for studying event semantics is twofold: firstly, event semantics are crucial for better text understanding; secondly, the central role of events makes them an excellent candidate for interpreting how powerful models, such as large language models (LLMs), function. More specifically, we summarize the reasons as follows:

1. **Semantic Understanding**: Events encapsulate essential semantic information, including

participants, time, and location. By analyzing event structures, we can capture nuanced meanings and relationships that are crucial for comprehensive semantic understanding.

2. **Complex Interactions**: Events interact with other discourse elements in intricate ways. These interactions form diverse structures that are pivotal for tasks like anaphora resolution, coreference, and event sequencing. Understanding these interactions enhances our ability to predict and analyze event structures accurately. This allows us to develop advanced models that leverage the rich interactions between events and other discourse elements. Methods like joint prediction and indirect supervision can exploit these interactions to improve model performance.
3. **Understanding Data**: Data is always a key theme in NLP and AI in general. Scaling expert-annotated datasets for event analysis is costly and time-consuming. Studying event structures motivates the development of scalable data annotation methods, such as crowdsourcing and indirect supervision, to increase data availability. This scalability is crucial for training robust and capable models, or for evaluating models and understanding their behaviors.
4. **Interpretability and Transparency**: Understanding how models process event semantics internally is vital for interpretability and transparency in AI systems. By studying event structures, we can provide insights into model training processes and help uncover the mechanisms behind event semantics processing. This transparency is essential for building trustworthy and explainable AI systems.

In summary, studying event structures is motivated by their central role in semantic understanding, and the intricate interactions they form with other discourse elements. The complexity of these structures presents challenges at predicting multiple interacting structures together, but also offers opportunities to leverage these interactions and deepen our understanding of language phenomena and potential limitation of models.

1.2 Structures and Problems Related to Event Semantics

Building upon the motivation to study event semantics, we now delve into the specific structures and problems associated with event semantics in more detail.

In an underlying world (real or imaginary), we consider the static configuration of entities and their properties as *states*. Events are processes that involve a change of states. For example, a starting state may be a vase placed on top of a table. If the vase falls from the table, its physical location changes. This change of state can be called the ”fall” event.

In text documents, events are typically realized as or referred to by spans of text, known as **Event Mentions** or **Event Nuggets**. The structures around textual event mentions are very rich. They may be connected to a location, a time interval, and several participants. Event mentions may also interact with one another. Another crucial aspect of event semantics involves the concept of **event arguments**. Event arguments are the participants, properties, and other contextual elements associated with an event.

Events are important discourse units that form the backbone of our communication. They play various roles in documents, with some being more central to the discourse by connecting other entities and events or providing key information of a story, while others are less relevant

and not easily identifiable by NLP systems. Hence, it is crucial to quantify the "importance" of events. For example, in a news excerpt describing a debate around a jurisdiction process, the event "trial" is central as the main discussion topic, while "war" is not. Researchers recognize the need to identify central events in applications such as detecting salient relations [222] and identifying the climax in storylines [207]. The salience of discourse units is important for language understanding tasks, including document analysis [15], information retrieval [220], and semantic role labeling [41]. Thus, proper models for finding important events are desired. This task of **event salience detection** aims to find events that are most relevant to the main content of documents.

Humans often omit information from utterances to avoid redundancy when the context is clear enough to resolve the missing parts. Two example problems related to event semantics are **Verb Phrase Ellipsis (VPE)** and **Implicit Argument Detection (IAD)**. Both can be viewed as forms of cross-sentence ellipsis resolution or cross-sentence anaphora problems, where we need to find a phrase to be anaphoric to the elliptical slot. The key difference is that the missing information in VPE is part of the predicate/event mention, while the missing information in IAD is the event argument.

Due to their connection with arguments, event semantics play a crucial role in phenomena related to other discourse elements. In the **Winograd Schema Challenge** [114], the resolution of pronouns often depends on changing one verb (one event predicate). Understanding how the events described in the sentences influence the referential properties of the pronouns is a very interesting problem. We will provide a pilot study on this topic in §8.

Various types of relations can be established between events. Events may collectively form larger structures. Analogous to the entity coreference problem, the same event can be represented by multiple text spans, necessitating the resolution of **coreference** to connect these mentions, which is potentially helpful for completing the information of an event. Another formulation is Schank's script theory [185], which suggests that information is centered around event sequences, enabling language understanding and inference. Many event structures are related to Schank's script. For instance, the "subevent" relation proposed in [102] organizes events into script clusters. The TempEval tasks [202] consider the temporal ordering of events. Following these directions, we proposed a new task named **Event Sequencing** for TAC KBP 2017, which aims at ordering events from text documents that belong to the same script. Both clustering events into the same script and event ordering are required in the task.

This thesis presents our study on these various structures related to event semantics. Below, we provide a brief formulation of the tasks about these structures:

Detection of Event Mention, Event Nuggets, Predicates. A first task in event semantics is the detection and classification of event mentions, also known as event nuggets. This task involves identifying spans of text that refer to events and then classifying these mentions into predefined event types. We follow the task definition used in the TAC-KBP event track evaluations [143, 144, 145], where each event mention is also assigned a realis attribute indicating its nature: ACTUAL (events that actually occurred), GENERIC (mentions of the event in general), or OTHER (other variations). This task is crucial because accurately identifying and classifying event mentions lays the foundation for further event-centric analyses, such as understanding event arguments and event salience. An example of event nugget annotation is shown in Figure 1.1.

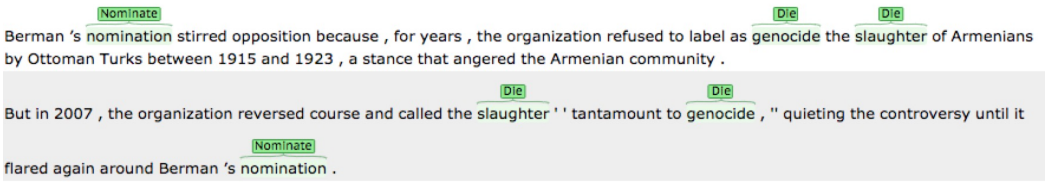

Figure 1.1: An example event nugget annotation

Verb Phrase Ellipsis. VPE is an anaphoric process where a verbal constituent is partially or totally unexpressed but can be resolved through an antecedent in the context. This phenomenon poses significant challenges for natural language processing because the missing verbal elements need to be inferred from surrounding text. For example, in the sentence ”His wife also [works for the paper], as **did** his father,” the light verb **did** represents the verb phrase ”works for the paper.” Identifying and resolving these elliptical structures is crucial for accurate semantic interpretation and text understanding. VPE requires models that can recognize and interpret context to reconstruct the missing verbal information, and locate the phrase spans accurately.

Argument Extraction and Implicit Argument Detection (IAD) Argument Extraction involves identifying the participants, properties, and other discourse elements associated with an event. Traditional argument extraction focuses on identifying these elements within a single sentence. IAD is a sub-problem of event argument extraction that focuses on finding arguments mentioned across sentence boundaries. This task is akin to implicit semantic role labeling (SRL), where the goal is to identify argument spans that fill the roles of frames. Event arguments often extend beyond sentence boundaries, introducing non-local or implicit arguments at the document level. For instance, in the sentence ”The new computer cost 3000 dollars, while the old one cost 1000 dollars. Nevertheless, he still **bought** the more expensive one,” the *money* argument for the *purchase* event, triggered by ”bought,” appears in the previous sentence. Detecting these implicit arguments is essential for comprehensive event understanding and accurate event argument modeling, requiring systems to effectively utilize broader context beyond individual sentences.

Event Salience Detection. The task of event salience detection aims to identify events that are most relevant to the main content of documents and the events that author would like to emphasize. These events are often central to the discourse, connecting other entities and events or providing key information for a story. For example, in a news excerpt describing a debate around a jurisdiction process, the event ”trial” is central as the main discussion topic, while ”war” is not. Recognizing central events helps improve the performance of various applications, such as detecting salient relations [222], identifying climaxes in storylines [207], semantic role labeling [41], or information retrieval [220].

Event Coreference and Quasi Coreference. Coreference resolution, the task of linking surface mentions to their underlying discourse entities, is an important task in natural language processing.

Most of the early work focused on coreference of entity mentions. The definition of event coreference is naturally similar to that of entity coreference. However, we would like to highlight the Quasi-Identity problem. That is, some of event relations exhibit subtle deviation from the perfect identity of events [102]. Quasi-identity occurs when event mentions are related but do not fully match in all aspects, making the task of coreference resolution more complex. Studies has discovered that distinguishing subevent and coreference relations is crucial for anaphora resolution[102, 216]. There are several types of quasi-identity that to be considered, we propose the new *Spatiotemporal Continuity* category during our crowdsourincg experiments in §7:

1. **Exact Coreference Cluster and Hopper**: In typical coreference literature, multiple references to the same underlying subject matter are easily considered coreferential, sharing an exact identity. However, this becomes tricky since stricter approaches to event coreference would require all event features to be identical. This approach cannot handle certain cases. For instance, two reports about the same terrorist incident may differ in the number of perpetrators, especially in the immediate aftermath of the event when facts are still being uncovered. **Event hoppers** [189] relax this requirement by allowing coreference of two events that are intuitively the same, even though certain features may differ. While this flexibility addresses the aforementioned annotation problem, it also makes the annotation task more challenging, necessitating the expertise of trained annotators.
2. **Subevent Relation**: This occurs when one event is a component or part of a larger event. For example, ”the trial” might include subevents such as ”the hearing” or ”the verdict,” which are part of the overall trial event but represent distinct phases or components.
3. **Membership Relation**: This type of quasi-identity involves events that belong to a common category or group but are not identical. For instance, multiple incidents of ”protests” in different cities may be related as members of a broader category of events but are not the same event.
4. **Spatiotemporal Continuity**: In this thesis, our crowdsourcing workflow (§7) allowed us to empirically identify a new type of quasi-identity, which we term spatiotemporal continuity after [212]. This occurs when an event gradually evolves over space and time, leading to cases of partial coreference. For example, the ”Haitian cholera outbreak” can be seen as a single extended event that unfolds across different regions and periods, resulting in mentions that share a spatiotemporal connection but are not identical.

Winograd Schema. The Winograd Schema Challenge [114] is a benchmark task designed to evaluate a system's ability to understand context and resolve ambiguous pronouns. Each schema consists of a pair of sentences that differ by only one or two words and contain a pronoun whose reference is altered by this small change. For example, in the sentence ”The city councilmen refused the demonstrators a permit because they [feared/advocated] violence,” the pronoun ”they” could refer to either the city councilmen or the demonstrators, depending on whether the verb is ”feared” or ”advocated.” Resolving such ambiguities often requires understanding the key event and its commonsense implications. The fact that a subtle change can cause a significant difference in prediction makes the Winograd Schemas particularly suitable for analyzing how well a model handles semantics.

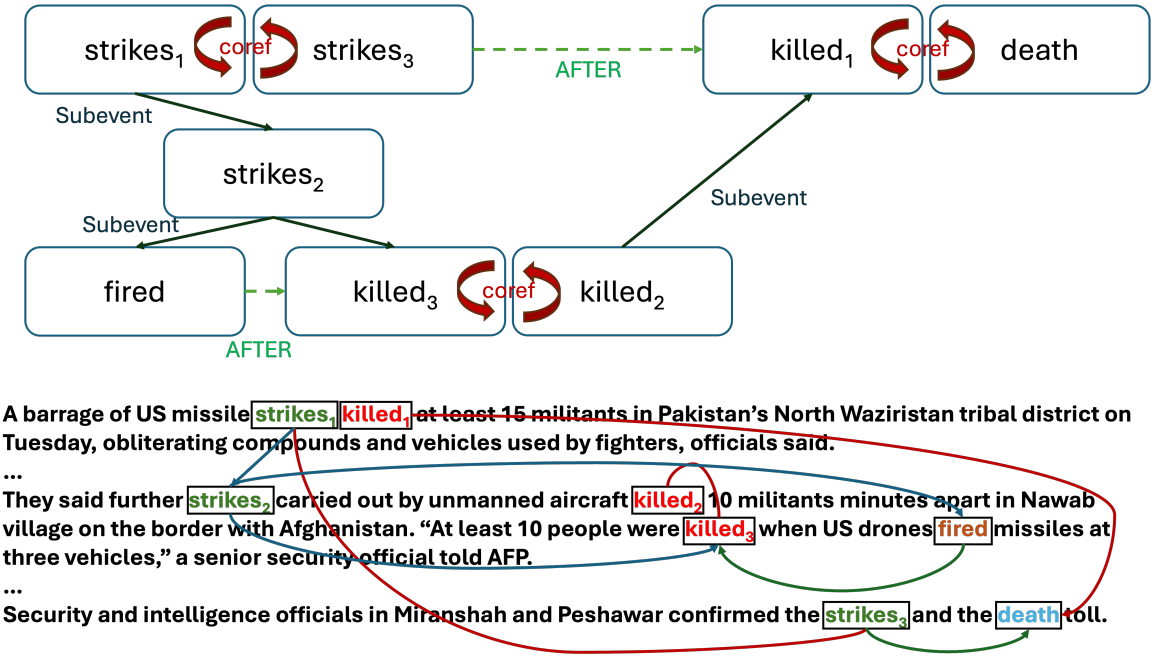

Figure 1.2: There are many relations between event mentions. Above: The event relations shown as a Directed Graph; Below: The actual event mentions in text annotated by the relations. Red lines are coreference links; solid blue arrows represent Subevent relations; dotted green arrows represent After relations.

Event Sequencing. As the name suggests, Event Sequencing puts events in logical orders. The task of event sequencing is related to early linguistic theory such as Schank's *scripts* [185], which suggest that humans organize information through procedural data structures that reassemble sequences of events. For example, the sequence of verbs *order, eat, pay, leave* may trigger the restaurant script. Humans use common sense to reason about the typical ordering of these events (e.g., *order* should be the first event, *leave* should be the last event). Event Sequencing is highly related to Event Quasi-Identity/Coreference problems. It connects with many concepts with the Quasi-Identity framework. The key relation, *AFTER* relation, is defined between event hoppers. As show in Figure 1.2, the event *fired* and the hopper *(killed$_2$/killed$_3$)* forms an AFTER relation, and both are subevent of *strikes$_2$*[1]. Further, it is worth noting that an event sequence often exhibit as subevents of some other events in text articles, such as the chain *fired*, *killed$_2$/killed$_3$* are subevents of *strikes$_2$*. However, this is not always guaranteed, as various other graph structures can be formed. The **Event Sequencing** task requires one to identify events within the same script chain and classify their logical orders. The key relation type in a event sequence is "AFTER". The AFTER relation connect events that follow the script order. For example, the event *order* should be followed by *eating*, reflecting a natural sequence of activities in a restaurant script. While the Event Sequencing task might seem similar to a temporal ordering task [170], it focuses on the

[1]Note that *fired* and *strikes$_2$* should be considered as their singleton hoppers, details are omitted in the figure.

logical and script order rather than time span issues (e.g., whether two events overlap in time or if one event's end time is earlier than another's).

1.3 Approach and Thesis Outline

At a high level, this thesis tackles the problems through two primary directions. In the first part, we present algorithms for decoding various event structures, primarily using supervised settings with expert-annotated data. However, the amount of training data available for each task is limited, and the datasets typically cover small, focused domains because these tasks are difficult to annotate, even for humans. Moreover, many semantic phenomena, such as different types of coreference or relations between events and entities, are not explicitly articulated in regular human utterances and thus require explicit annotation.

On the other hand, a large amount of training data is often necessary for learning-based systems to successfully capture this knowledge. Limited training data results in systems that struggle to generalize. Supervised methods often face challenges with basic structures and intricate details, which are far from decoding semantics. To address this, in the second part of the thesis, we explore three different approaches to scale the data. These approaches reveal intriguing insights and show sparks of semantics (I cannot resist to overload this term from [26]).

Part I: Supervised Structure Prediction for Event Semantics. We first present our earlier work on several event structure prediction problems using supervised approaches, including **Event Detection (§ 2)**, **Event Coreference (§ 3)**, **Event Sequencing (§ 4), Ellipsis (§ 5)**. In this part, we explore the rich linguistic phenomena observed regarding event mentions, such as the properties (e.g., Realis) attached to mentions and the relations (e.g., coreference and subevents) between them. Previous corpus studies on these problems have resulted in several annotated datasets. Hence, we focus on developing supervised learning approaches that leverage the unique structures of these problems.

Part II: Scaling Up Data for Event Semantics. While we have made progress in predicting structures, like any supervised approaches, these model tend to only focus on predicting the structures being taught. In this part, we shift our focus to scaling up the data, exploring possibilities from three angles. These approaches surface new, interesting semantic phenomena. Here is an overview of these three angles:

1. **Crowdsourcing.** Crowdsourcing is one typical way to scale up annotated dataset. However, as we previously discussed, many of these annotation tasks are often very complex to annotate even for expert annotators (also reported by [177]), which may be too challenging for crowdworkers. Therefore, we propose a new annotation workflow that breaks down the annotation tasks into simpler steps. We apply this annotation approach on a **cross-document coreference** setting. In addition to asking the crowdworkers on the coreference decisions, we ask follow-up questions to collect evidence for event mention, time, location, and participant(s) overlap between corefering mentions. This approach allows us to find a novel type of partial identity termed as *spatiotemporal continuity*. We detail this approach in **§7**.

2. **Indirect Supervision.** Another approach to scaling data is to find large amounts of indirect supervision signals without direct human annotation. We apply this method to study the problem of predicting **event salience** in **§6**. Inspired by previous work on Entity Salience [63], we use summarization as a proxy task to create an event salience dataset using articles from the Annotated New York Times corpus [183]. Besides improving model performance, we observe that the model learns to place script-related event mentions and correlated event arguments into the learned embedding kernels.
3. **Language Model on Coreference.** Large Language Models (LLMs) are currently among the best approaches for solving NLP problems. It is surprising to see these models excel at complex questions, even effectively addressing many problems mentioned in this thesis. However, some problems remain unsolved, at least for medium-sized language models. The recent LLM performance on Winogrande [182] is an interesting example. Winogrande is a more challenging dataset than the original Winograd Schema, created by systematically debiasing machine-detectable embedding associations. Llama3 8B [3] scores an impressive 76.1% on Winogrande but still falls short of the human level performance of 94.0%. While it is intriguing that language models trained on next-token prediction achieve significant scores on such tasks, the performance gap leaves room for further study. In this chapter, we analyze the model's performance and the ability developmental trajectory of an LLM, using model checkpoints from the LLM360 [125] project. Specifically, we adopt circuit analysis methods [49, 77, 78, 94, 210] to observe what circuits are formed to solve the Winograd Schemas. We find that while an LLM develops a few seemly robust algorithms for some Winograd Schemas, it often solves many of them without using the proper context, particularly in models around the FLOP level of $4e22$ (a 7B model trained on 1T tokens).

Part I

Analyzing Event Structures with Expert Annotated Data

In human language, the expression of events are rich discourse element in terms of various properties and their relations with other discourse elements in the document. Events mentions are used to refer to possible happenings or state changes in the world. Bach classify the notions of events or eventualities into three different types of components [9]: *states* (durative and changeless), *processes* (durative without goal) and *actions* (durative with explicit goals), where the latter two are *non-states* notions. Under this definition, prior computational and corpus study on events either focus on the *non-states* [143] notions, or all the 3 components [4, 170]. In this thesis, we generally focuses on the *non-state* event mentions, and will use the term *states* to represent the state eventuality explicitly.

In addition to the basic characteristics described in Bach [9], many attributes can be used to decorate events, reflecting their semantic richness. One of the most notable and practically useful property for event mentions are their **types**. Under a certain ontology for a particular domain, we can assign some notable and salient event types to these event mentions. For instance, "price drop" and "investment" in a business domain; "bombing" and "injury" in a military domain.

The **realis status** introduced in TAC KBP series of annotation, or the **epistemic status** in the IC domain dataset, are used to classify whether the event expressions are corresponding to an actual event. For instance, an event can refer to a real happening, but can also refer to a hypothetical state, or event refer to something that might happen in the future.

Apparently these are only some examples of a large number of possible attributes, some other attributes include event duration [159, 160, 213, 214], past or future [103] events and many more. The study of these properties are out of the scope of this thesis. In **§2**, we will introduce our early attempts on predicting event mention spans, and the event attributes including realis and epistemic.

The richness of event mentions also lie in the complex interaction and structures among them. For example, two event mentions may be coreferential, follow a sequential/temporal relation, or follow other anaphora relations. Furthermore, the relations between the event mentions may reflect relations between their arguments, hence can promote inference on the entity mentions. In this part, we will present our work on learning event coreference in **§3**, event sequencing in **§4**, and another type of anaphora, verb phrase ellipsis, in **§5**.

In this part, we will focus on presenting supervised approaches on learning the event semantics. Indirect supervision methods will be presented in Part II.

Chapter 2

Event Detection

2.1 Introduction

Detecting the event mention instances is the first and most fundamental step for event semantic modeling. There are different ways to mark an event mention instance in text. One can mark the event predicate span, or the whole context of the event mention, including the arguments. In this chapter, we describe our approach in detecting the predicate span, a task which is often referred to as Event Nugget Detection or Event Trigger Detection. In this chapter, we present our system on event detection on two languages: English and Chinese. Our featured based models are ranked as one the top systems in the TAC-KBP evaluations [143, 144].

2.1.1 Event Mention Detection Task

An event nugget detection task normally requires a system to identify spans of text the refers to an event mention, and then classify them as one of the predefined event types. We follow the task definition in TAC-KBP event track evaluations [143, 144, 145], where each mention is additionally assigned with a realis attribute of ACTUAL (events that actually occurred), GENERIC (mentions of the event in general) or OTHER (other variations). An example annotation is shown in Figure 2.1.

Limited by resources, event detection datasets are normally limited to a small scope. For example, in all the RichERE annotations, there are only 38 type-subtypes, as listed in Table 2.1.

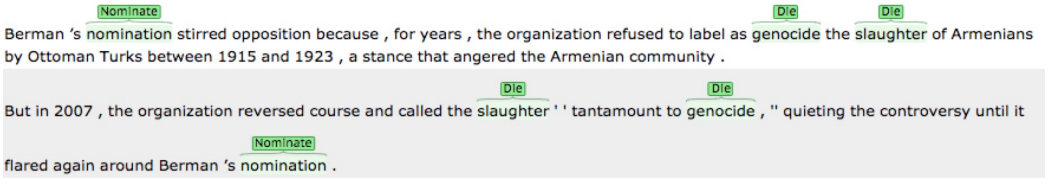

Figure 2.1: An example event nugget annotation

Type	Subtype	Type	Subtype	Type	Subtype
Business	Start-Org	Life	Divorce	Justice	Release-Parole
Business	End-Org	Life	Injure	Justice	Trial-Hearing
Business	Declare-Bankruptcy	Life	Die	Justice	Sentence
Business	Merge-Org	Transaction	Transfer Ownership	Justice	Fine
Conflict	Attack	Transaction	Transfer Money	Justice	Charge-Indict
Conflict	Demonstrate	Transaction	Transaction	Justice	Sue
Contact	Meet	Movement	Transport-Person	Justice	Extradite
Contact	Correspondence	Movement	Transport-Artifact	Justice	Acquit
Contact	Broadcast	Personnel	Start-Position	Justice	Convict
Contact	Contact	Personnel	End-Position	Justice	Appeal
Manufacture	Artifact	Personnel	Nominate	Justice	Execute
Life	Be-Born	Personnel	Elect	Justice	Pardon
Life	Marry	Justice	Arrest-Jail		

Table 2.1: List of Event Types and Subtypes in the RichERE annotations

Other event related triggers in a text documents are ignored. The task of event detection is essentially searching for domain-specific triggers.

2.2 Model for Type Classification

We consider event nugget detection as a sequence labeling task and deploy a Conditional Random Field (CRF) model to detect mention span and event type. The CRF model is trained with the structured perceptron [47], which is outlined in Algorithm 1. The decoding step is done using standard Viterbi algorithm. After training, we obtain the model using the average weight variation as described in Collins [47].

A number of "multi-tagged" mentions are annotated in the corpus, in which a mention might have one or more event types. For instance, an event nugget "KILL" is often associated with "Life-Die" and "Conflict-attack". To deal with them, We simply combine multiple labels for each mention into a single label[1].

[1]Though this can be treated as a multi-label classification problem, however, simple concatenation only result in a label set of 56 types, which can be easily handled

Algorithm 1 Structured perceptron.

Input: training examples $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$
Input: number of iterations T
Output: weight vector $\mathbf{w}$

$\mathbf{w} \leftarrow \mathbf{0}$ ▷ Initialization.
for $t \leftarrow 1..T$ **do**
  for $i \leftarrow 1..N$ **do**
    $\hat{y}^{(i)} = \arg\max_{y \in \mathcal{Y}(x^{(i)})} \mathbf{w} \cdot \Phi(x^{(i)}, y)$
    if $\hat{y}^{(i)} \neq y^{(i)}$ **then**
      $\mathbf{w} \leftarrow \mathbf{w} + \Phi(x^{(i)}, y^{(i)}) - \Phi(x^{(i)}, \hat{y}^{(i)})$
return $\mathbf{w}$

An event mention is normally composed by its mention trigger and the arguments. To get a list of arguments for the event mention. We run two Semantic Role Labeling system, the PropBank style Fanse Parser [56] and the FrameNet style Semafor Parser [198]. In addition, to reduce sparsity, we further incorporate a few external data resources, including WordNet [73] and a set of Brown Clustering labels trained on newswire data [192]. Lemma, part-of-speech, NER and parsing information are all obtained through the Stanford CoreNLP system [132].

Using these resources, we employ regular linguistic features for mention type detection, which are summarized as followed:

1. Part-of-Speech, lemma, named entity tag of words in the 2-word window of the trigger (both side), the trigger word itself and the direct dependent words of the trigger.
2. Brown clusters, WordNet Synonym and derivative forms of the trigger.
3. Whether the words in the 5 word window match some selected WordNet senses, including "Leader", "Worker", "Body Part", "Monetary System", "Possession", "Government", "Crime" and "Pathological State".
4. Closest named entity type.
5. Dependency features, including lemma, dependency type and part-of-speech of the child dependencies and head dependencies.
6. Semantic role related features includes the frame name and the argument role, named entity tag, argument head word lemma and WordNet sense (selected from the above list as well) of the arguments.

The WordNet related features are selected following the intuition that certain category of words are likely to imply the existence of certain events. For example, "Leader" are normally associated with "Personnel" type. The model generalize better by selecting appropriate levels of word sense.

2.3 Realis

We train a separate Realis detection model using Supper Vector Machine in LIBLINEAR[2]. We reuse many features from the mention detection to capture the context of these mentions. However, we exclude most of the lexicalized features because they tend to be overfitting in our prior experiments. We design a specific feature to capture whether the phrase containing the event mention is quoted (if the whole sentence is quoted, we do not fire this feature).

2.4 Adapting to a Chinese Event Detection system

To extend our system to handle Chinese documents, we develop similar features for Chinese. Most of the features can be reused without changes in the Chinese system, which includes: window based features[3], syntactic based features, entity features, head word features and SRL features. We also use the Brown clustering features with clusters induced form Chinese Gigaword 3[4].

Due to the nature of Chinese language, the Chinese tokens normally contain more internal structure and each single character in the token may convey useful semantic information. The way how the individual characters combine will affect the semantic of the event word. This is previously discussed in Li et al. [115] as verb structures. In other words, the position of a character in the verb matters. For example, the character 解 means "unbind" in the word 解雇 (fire), but means "console" in the word 劝解 (console). Following these intuition, we further add the following character related features:

1. Whether the token contains a character.
2. The contained character and its character level Part-of-Speech.
3. The first character of the token.
4. The last character of the token.
5. Base verb structure feature as described in [115]: we use a feature to represent one of the base verb structure. In addition to the 6 main structures proposed by Li et al. [115], we added 3 structures for completeness: 1. No verb character found 2. The verb character is found after 2 characters and 3. Other: any cases that are not defined above.

2.4.1 Improving Recall on Chinese

During the system development, we observe that our Chinese system suffers from serious low recall despite all the features we added in. By following the training procedures, we hypothesize that the annotated Chinese data is not complete (see § 2.6.2 for more discussion). As a result, our learning algorithm will be biased by the missed events and learn incorrect negative signals. The final model thus will be very conservative in making predictions, leading to a low recall.

We mitigate the problem by ignoring all training sentences that do not contain an event mention, which reduce the probability of missed annotations. On our development experiments,

[2]https://www.csie.ntu.edu.tw/ cjlin/liblinear/

[3]However since the discussion forum training data are quite noisy, we restrict the POS window to 1 instead.

[4]`http://www.cs.brandeis.edu/~clp/conll15st/dataset.html`

we found that this simple trick can directly raise our nugget detection performance by about 3%. The performance improvement also support our hypothesis that the Chinese dataset is indeed not fully annotated.

2.5 Experiments

We have participated in TAC-KBP Event Nugget track 2015 and 2016 [122, 124]. We thus follow the official evaluation setup. The TAC-KBP 2015 track provides a training corpus of 158 documents and an evaluation set of 202 documents. There are no new training data provided in TAC-KBP 2016, thus we train our system using both the training and testing data from TAC-KBP 2015. The evaluation set of TAC-KBP 2016 contains 169 documents. The datasets are all coming from two different genre: newswire or discussion forum. One major change in the TAC-KBP 2016 evaluation is that the types required for evaluation is reduced to 18 types, which is a subset of the previous 38 types.

2.5.1 Evaluation Results

Here we directly report our official system performance in the evaluation workshop [143, 144]. In TAC-KBP 2015, our official event nugget performance is summarized in Table 2.4a. Our system ranks the third place in the Span and Realis sub-score, and second in the All attribute sub-score. In TAC-KBP 2016, our official english nugget score in Table 2.4b. Our system ranks 2nd among all the participants in terms of the final event type F1 score (Table 2.2, our system name is LTI-CMU1). The other sub-scores are also competitive, which ranks at the top few systems.

	Prec.	Recall	F1
UTD1	47.66	46.35	46.99
LTI-CMU1	61.69	34.94	44.61
wip1	51.76	38.98	44.47
NYU3	41.88	47.21	44.38
SoochowNLP3	49.92	38.81	43.67
RPI-BLENDER2	44.51	39.87	42.07
SYDNEY1	46.48	30.33	36.70
Washington1	42.15	29.41	34.65
aipheshd-t161	36.83	29.28	32.62
UMBC2	37.36	27.33	31.57
HITS3	41.79	25.30	31.52
CMUML3	60.44	15.58	24.77
UI-CCG2	25.81	18.53	21.57
IHMC20161	0.69	0.52	0.59

Table 2.2: English Nugget Type System Ranking of TAC-KBP 2016

We are a little surprise to see our English nugget detection performance drops about 13% (span and type) comparing to from KBP 2015 to KBP 2016. However, our relative ranking is almost unchanged. Our analysis [144] has shown that this is actually because the change in the type set: systems on both years perform equally well (or even better in 2016) on the selected types. In fact, the average score on these types are actually lower than the full set.

Our Chinese event detection systems also produce competitive results. The Chinese system is the first place on all the sub-score in TAC-KBP 2016 (Table 2.3).

		Prec.	Recall	F1
Span	LTI-CMU1	56.46	39.55	46.52
	UTD1	47.23	43.16	45.1
	LTI-CMU3	56.19	35.35	43.4
	UI-CCG1	28.34	39.61	33.04
	RPI-BLENDER1	62.46	18.48	28.52
Type	LTI-CMU1	50.72	35.53	41.79
	UTD1	41.9	38.29	40.01
	LTI-CMU3	49.7	31.26	38.38
	UI-CCG1	24.01	33.55	27.99
	RPI-BLENDER2	59.87	17.5	27.08
Realis	LTI-CMU1	42.7	29.92	35.18
	UTD1	35.27	32.23	33.68
	LTI-CMU3	43.11	27.12	33.29
	RPI-BLENDER2	48.46	14.16	21.92
	UI-CCG1	9.65	13.49	11.25
All	LTI-CMU1	38.91	27.26	32.06
	UTD1	31.76	29.02	30.33
	LTI-CMU3	38.54	24.25	29.77
	RPI-BLENDER2	46.69	13.65	21.12
	UI-CCG1	8.31	11.62	9.69

Table 2.3: Official Chinese Nugget Performance Ranking at TAC-KBP 2016.

In order to test the effect of our realis model, we evaluate its performance given gold standard mention span and types in the training data. We report the 5-fold validation result in Table 2.5. Note that the precision and recall are the same because the gold spans are given.

2.6 Discussions

2.6.1 Multi-Type Events

There is a small number of event types in the evaluations. Further, the possible types of double tagging are limited by in the dataset. This is because the current annotation scheme only considers

	Precision	Recall	F1
Span	82.46	50.30	62.49
Type	73.68	44.94	55.83
Realis	62.09	37.87	47.05
All	55.12	33.62	41.77

(a) Official English Event Nugget Performance at TAC-KBP 2015.

	Prec.	Recall	F1
Span	69.82	39.54	50.49
Type	61.69	34.94	44.61
Realis	45.78	25.93	33.11
All	40.19	22.76	29.06

(b) Official English Event Nugget Performance at TAC-KBP 2016.

Fold	Precision	Recall	F1
1	71.68	71.63	71.66
2	64.06	64.06	64.06
3	62.07	62.07	62.07
4	72.66	72.66	72.66
5	62.21	62.21	62.21

Table 2.5: 5-fold validation for Realis detection on training data with gold span and types

a limited pool of event types. Our current solution is simply treating the double-tagged types as a new class in classification. However, we realize there are rich phenomenon behind this. In fact, we find the event arguments to be closely related to these different event types. For example, a predicate "kill" may have types "Conflict.Attack" and "Life.Death": the former one is more related to the state of the "Attacker" while the latter one relates to the state of the "Victim".

2.6.2 Chinese Data Annotation

We hypothesize that the Chinese datasets are not fully annotated. We take a closer look in the data and found a number of missed event nuggets. Here we list a couple examples:

(1) 支持香港同胞争取[Personnel.Elect 选举]与被[Personnel.Elect 选举]权！

(2) 司务长都是骑着二八去[TransferOwnership 买]菜去。

(3) 海豹行动是绝密，塔利班竟然可以预先得知？用个火箭就可以[Conflict.Attack打]下来，这个难度也实在是太高了吧。

In the above examples, we show several event nuggets. However, mentions annotated in red are not actually annotated in the Rich ERE datasets. Especially, in example 1, the first 选举 is annotated but the second one is not. Such inconsistencies happen a lot across the dataset. When training with such data, the classifier will likely to be quite conservative on making event nugget predictions. We conduct a very simple quantitative analysis by comparing the ACE 2005 Chinese annotation against the Rich ERE Chinese annotation. Table 2.6a and Table 2.6b summarize the top 5 double-character tokens annotated in ACE and RichERE. For the most popular event mentions,

Rich ERE annotated only a smaller percentage comparing to ACE.

In addition, we find that the most popular event nuggets are mostly single character in the Rich ERE datasets, such as 打(170), 说(148), 死(131), 杀(118) . In fact, in top 20 most popular event nuggets of Rich ERE, there are 17 single-character nuggets, this number is only 6 in ACE. These single character tokens are more ambiguous comparing to a double character mention (for example, 打 can represent the action of "calling someone" or "attacking someone", which corresponds to very different event type. This is because language in discuss forum posts are normally not formal. This actually challenges our event nugget systems to deal with deeper semantic problems.

Token	Annotated	Total	%
冲突	100	119	84.03%
访问	64	90	71.11%
受伤	53	59	89.83%
死亡	46	50	92.00%
前往	44	52	84.62%

(a) Top 5 double character mentions in ACE 2005 Dataset

Token	Annotated	Total	%
战争	96	223	43.05%
死亡	24	33	72.73%
暗杀	22	40	55.00%
入侵	18	22	81.82%
自杀	17	33	51.52%

(b) Top 5 double character mentions in TAC-KBP 2016 Dataset

Chapter 3

Pairwise Event Coreference and Sequencing

3.1 Introduction

Coreference resolution, the task of linking surface mentions to their underlying discourse entities, is an important task in natural language processing. Most of the early work focused on coreference of entity mentions. Recently, event coreference has attracted attention on both theoretical and computational aspects. However, most event coreference work is preliminary and applied in quite different circumstances, making comparisons difficult or impossible.

However, one aspect that makes the problem challenging is that events can form a complex structure and relate to each other in various ways [16, 106]. In particular, some of event relations exhibit subtle deviation from the perfect identity of events[102]. In event anaphora, subevent relation is one of the important relations. Studies has also discovered that distinguishing subevent and coreference relations is crucial for anaphora resolution[102, 216].

In this chapter, we first provide an overview of many prior work around event coreference and highlights the differences in settings. The comparisons to related work in prior papers are not really appropriate due to these differences. We then present a supervised approach to event coreference, and describe a method for propagating information between events and their arguments that can improve results. In our method, different argument types support different methods of propagation. For these experiments, we annotate and use a corpus of 65 documents in the Intelligence Community (IC) domain that contains a rich set of within-document coreference links [102].

3.2 Related Work

Table 3.1 summarizes recent work on event coreference resolution. For the reasons below, only one supervised system [2] and two unsupervised [17, 51] on within-document event coreference are suitable as a basis for ongoing comparison.

	Gold standard used	Cross/within Document		Compatible definition	Corpus
		Within	Cross		
Lee et al. [112]			✓	✓	ECB
Sangeetha and Arock [184]	ACE mention, arguments and attributes	✓		✓	ACE
Cybulska and Vossen [51]	Mention head word	✓		✓	IC
McConky et al. [135]	ACE mention, arguments and attributes	✓		✓	ACE
Li et al. [116]	Human entity and event mention detection	✓		✓	Unavailable
Bejan and Harabagiu [17]		✓(ACE, ECB)	✓(ECB)	✓	ECB, ACE
Chen and Ji [39]	ACE mention, arguments and attributes	✓		✓	ACE
Elkhlifi and Faiz [70]		✓			Unavailable
Naughton [148]		✓			IBC, ACE
Pradhan et al. [169]		✓			OntoNotes
Ahn [2]	ACE mention, arguments and attributes	✓		✓	ACE
Bagga and Baldwin [11]			✓		Unavailable

Table 3.1: A range of event coreference resolution work with different settings.

3.2.1 Problem Definition

Different approaches use different definitions of the problem (see Compatible Definition column). However, as discussed in recent linguistic studies [102, 175], the existence of different types and degrees of coreference makes it necessary to agree on the definition of coreference before performance can be compared. The lack of clarity about what coreference should encompass rules out several systems for comparison. OntoNotes created restricted event coreference [169], linking only some nominalized events and some verbs, without reporting event-specific results. Both Naughton [148] and Elkhlifi and Faiz [70] worked on sentence-level coreference, which is closer to the definition of Danlos [54]. However it is unclear when one sentence contains multiple event mentions, and hence these are not comparable to systems that process more specific coreference units.

3.2.2 Dataset and Settings

Early work by Bagga and Baldwin [11] conduct experiments only on cross-document coreference. Recent advanced work on event coreference is by Bejan and Harabagiu [17] and Lee et al. [112] use the ECB corpus[1] (or a refined version[2]) to evaluate performance, which is annotated mainly for cross-document coreference. In this corpus, within-document coreference is only very partially annotated; most difficult coreference instances are not marked.

(1) 1. Indian naval forces `came to the rescue (E1)` of a merchant vessel under `attack (E3)` by pirates in the Gulf of Ade on Saturday, `capturing (E2)` 23 of the raiders, India `said (E4)`.

2. Indian commandos boarded the larger pirate boat, **seizing** 12 Somali and 11 Yemeni nationals as well as arms and equipment, the statement `said`.

(2) 1. The Indian navy `captured (E2)` 23 piracy suspects who `tried (E5)` to `take over (E3)` a merchant vessel in the Gulf of Aden, between the Horn of Africa and the Arabian Peninsula, Indian officials `said (E4)`.

2. In addition to the 12 Somali and 11 Yemeni suspects, the Indian navy `seized` two small boats and "a substantial cache of arms and equipment", the military `said` in a statement.

The examples sentences are extracted from two documents from the ECB. In both documents, event mentions appear in the first sentence are annotated once, but not in the subsequent sentences. In example 1, we find in one of the subsequent sentences the event mention "`seizing`" which should actually marked as coreferent with "`capturing (E2)`". In example 2, we find a more tricky case: the mention "`seized`", which has semantics similar to "`captured`" but this pair is not marked as coreference due to different patients. In cross-document settings, we also find discrepancies between the definitions. In ECB, "`attack (E3)`" in example 1 is annotated as coreferent with "`take over (E3)`" in example 1, which we believe is wrong: at best, the attack is only a part of the attempt to take over the merchant vessel.

[1] http://adi.bejan.ro/data/ECB1.0.tar.gz
[2] http://nlp.stanford.edu/pubs/jcoref-corpus.zip

Goyal et al. [88] use a distributional semantic approach on event coreference. However, they adopt a unconventional evaluation setting. They draw from the IC corpus an equal number of positive and negative testing examples, which is different from the natural data distribution.

3.2.3 Gold Standard Annotations

Recent work using the ACE 2005 corpus[3] Ahn [2], Chen and Ji [39], Chen et al. [40], McConky et al. [135], Sangeetha and Arock [184] agrees with our definition of coreference. However, the ACE corpus annotations, in addition to event mentions, also include argument structures, entity ids, and time-stamps. Most coreference systems on the ACE corpus make use of this additional information. This makes them impossible to compare to systems that do not make this simplifying assumption. It also makes results achieved on ACE hard to compare to results on corpora without this additional information. Among these work, only Ahn [2] reported some results using system generating arguments, we compare our system against it.

There are also other problems that make the comparison difficult. Li et al. [116] use a hand-annotated web corpus, which is not publicly available for comparison. In summary, anyone wanting to work on within-document event coreference has to obtain a corpus that is fully annotated, that does not include additional facilitating information, whose definition of coreference respects the theoretical considerations of partial coreference, and that has other systems freely available for comparison. Meeting these criteria is not easy. The closest work we find is by Cybulska and Vossen [51] and Bejan and Harabagiu [17], both adopt unsupervised methods for event coreference. Ahn [2] also reported results on ACE by swapping gold standard annotations with system results.We compare our system to their results on their corresponding corpus.

3.3 Corpus

Our system is trained and evaluated on the IC domain corpus, which annotates several different event relations. Table 3.2 summarizes the corpus level statistics and the average over documents. In this work, we focus on full coreference relations. The inter-annotator agreement among 3 annotators for full coreference is 0.614 in terms of Fleiss' kappa [82]. For detailed definition for the corpus, we refer readers to Hovy et al. [102]. To facilitate future research, We also report our system results on the ACE 2005 training dataset, which contains 599 documents.

3.4 System description

Our system is almost end-to-end, except that we start with a minimal gold standard head word annotations in order to focus on the core coreference problem. This approach is the same as Cybulska and Vossen [51] and Bejan and Harabagiu [17][4].

[3] http://www.itl.nist.gov/iad/mig/tests/ace/2005/doc/

[4] Although Bejan and Harabagiu [17] use automatic mention detection to extend the mention set for training, they only use true mentions of the ACE dataset at evaluation time.

	Total	Avg.
Event Mention	2678	41.2
Non-elliptical Domain Event Mention	1998	30.7
Reporting Event Mention	669	10.29
Full coreference relations	1253	21.6
Subevent relations (parent-child)	455	8
Membership relations (parent-child)	161	2.9

Table 3.2: Corpus Statistics

3.4.1 Procedure

Similar to Chen et al. [40], we approach the problem first with a conventional pairwise model:

1. Supervised classification that determines the probability whether two mentions co-refer. The classifier used in the experiment is Random Forest [23], implemented in Weka [91].
2. Clustering that processes all the pairwise scores to output the final clusters of pairs.
3. In addition, we added a third step after clustering, information is propagated between event mentions to enrich the original feature set.

The last step tries to enrich the event representations during clustering. Typically, the information carried from one event to its coreferent mention is about the participants (agent, patient, etc.). When an event has been enriched by receiving information from another, it may in turn now be linkable to a third event. The system repeats this process until no more information can be propagated. Currently, the propagation includes two parts: 1. if one mention has missing arguments, they will be copied over from the co-referred counterpart; 2. if both arguments are present, information not presented in one will be copied from another.

Similarly, Lee et al. [112] show that jointly modeling references to events and entities can boost the performance on both. We hold a similar assumption. But by focusing on events and their arguments, we can perform propagation specific to each type of argument, for instance, geographical reasoning as described below.

3.4.2 Features

In addition to typical lexical and discourse features, we also model an event mention with its surface form and its arguments, including agent, patient[5], and location. We use a rich set of 105 semantic features, described in table 3.3.

[5]Specifically, these are defined as ARG0, ARG1 in PropBank. They could be more-specific variants roles such as *experiencer*, but we prefer a smaller set for simplicity.

Agent, patient extraction and propagation

We use the semantic parser *Fanse* [198] to annotate the predicate arguments defined in PropBank. For nominal events, we extract agent and patient using heuristics such as finding the token attached to the event mention with specific words (such as "by") and modifiers as agent (e.g., HAMAS in HAMAS's attack). During the propagation step, information not present in one entity can be copied from another.

Location extraction and propagation

In contrast to agent and patient, the propagation of location information employs external information to gain additional power. We use the *Stanford Entity Recognition* [80] engine to identify location mentions. *DBpedia Spotlight* [137] is run to disambiguate location entities. DBpedia [8] information, such as cities, country, and alternative names, are then injected. When the location is not found in DBpedia, we search the mention string in *GeoNames*[6] and use the first result with highest Dice coefficient with the surface string. This world knowledge enriches annotation. For example, we can now match the mention "Istanbul" with the country name "Turkey".

3.4.3 Clustering

We conduct experiments with two simple clustering methods. The first is a pure transitive closure that links all pairs mentions that the classification engine judges as positive. The second is the Best-Link algorithm of Ng and Cardie [149], which links each mention to its antecedent with the highest likelihood when the classifier judges as positive.

3.5 Evaluation

3.5.1 Evaluation Metrics

Coreference evaluation metrics have been discussed by the community for years. To enable comparison, we report most metrics used by the CoNLL 2012 shared task [168], including MUC [43], B-Cubed [10], entity-based CEAF [130], and BLANC [173]. Pairwise scores are used to provide a direct view on performance.

3.5.2 Experiments and Results

We split the documents in IC corpus randomly into 40 documents for training and development, and 25 for testing. Parameters such as the probability threshold to determine coreference are tuned on the 40 documents using five-fold cross validation. Optimization is not done separately for each metric. We simply use a universal classifier threshold optimized for pairwise case. During experiment, the propagation step is actually performed for only one iteration, since no further

[6] http://www.geonames.org/

information is propagated. On the ACE corpus, we simply apply the best model configuration from IC corpus and train on 90% of the documents (539) for training and 10% for testing (60).

Table 3.3 summarizes the overall average results obtained by BestLink on both ACE and IC corpus (BestLink consistently outperforms naively full transitive closure). We also attach three other reported results at the end. Note that these results are not directly comparable: Cybulska and Vossen [51] and Bejan and Harabagiu [17] use unsupervised methods, thus their reported results are evaluated on the whole corpus; Ahn [2] also use a 9:1 train-test split, but the split might be different with ours. A simple comparison shows that our results outperform these systems in all metrics, which is notable because all these metrics are designed to capture the performance from different aspects.

To interpret the results, it should also be noted that because of the existence of large number of singleton clusters, some measures such as B^3 seem to be high even using the most naive feature set. By looking at the pairwise performance, however, we see that current best F-score is only about 50%. There are still many challenges in event coreference.

	Pairwise			MUC			B^3			CEAF-e			BLANC		
IC corpus	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
Discourse + Lexical	32.69	25.11	28.40	41.7	33.58	37.2	79.46	74.06	76.67	66.89	73.95	70.24	59.77	61.2	60.43
+ Syntactic	47.12	35.15	40.26	52.6	47.63	50.0	82.24	81.46	81.85	76.91	80.21	78.53	64.76	68.59	66.42
+ Semantic (no arguments)	51.15	42.22	46.26	54.5	49.1	51.68	82.12	82.08	82.1	74.93	78.31	76.58	65.41	69.98	67.35
+ Arguments	55.96	47.86	51.60	56.87	**55.81**	56.33	83.38	**85.58**	**84.46**	**88.13**	80.73	**80.43**	68.77	**75.21**	71.46
+ Propagation	**59.04**	**48.27**	**53.11**	**68.72**	55.5	**61.44**	**89.28**	79.89	84.33	75.14	**82.9**	78.83	**82.28**	70.77	**75.06**
Cybulska and Vossen [51]	—	—	—	—	—	—	81.0	71.0	76.0	—	—	—	—	—	—
ACE corpus															
This work	55.86	40.52	**46.97**	53.42	48.75	50.98	89.9	88.86	**89.38**	85.54	87.42	**86.47**	70.88	70.01	70.43
Bejan and Harabagiu [17]	43.3	47.1	45.1	—	—	—	83.4	84.2	83.8	76.9	76.5	76.7	—	—	—
Ahn [2]	—	—	43.3	—	—	—	—	—	—	—	—	—	—	—	—

Table 3.3: Evaluation results and comparisons

3.6 Discussion

The evaluation results show that almost all types of features help to improve the performance over all metrics rather consistently. However, preliminary error analysis shows that some events are still clustered incorrectly even when arguments match. We argue that limitations in argument extraction and entity coreference prevent these features from contributing directly to correct coreference decisions. On the other hand, the results of propagation show that new information helps to find more links but inevitably comes with a drop in precision. We consider that modeling event and arguments holistically like Lee et al. [112] would help guide the propagation. By inspecting the data, we hypothesize that the main benefits brought by the propagation scheme is to match arguments of two coreferent events. If the arguments are nominal events, they will be then marked as coreferent due to the feature "Event as Entity" (See Semantic features in table 3.4). In the following example, if the two event mentions "planning" are marked as coreference, then the corresponding argument "attack" will be also marked as coreference.

(3) A member of the Islamic militant movement HAMAS suspected of **planning** a suicide **attack** against Israel surrendered to Palestinian police here after a six-hour shootout on Friday. HAMAS's military wing, was on the run from both Palestinian and Israeli police for **planning** anti-Israeli **attacks**.

This hypothesis is also in line with our observation that propagation can only be performed for one round, because the nominal event themselves are unlikely to have other nominal events as arguments. Such interactions between event mentions also remind us that conference can be possibly improved by other types of event relations, such as subevent relations.

Furthermore, the system tends to merge clusters where the event mention head words are the same because the head word feature receives a high weight in the model, even when this is not appropriate. More work should be performed on disambiguating such difficult cases.

We show that rich linguistic features, especially event arguments, can improve event coreference performance. Argument specific information propagation further help finding new relations. However, our proposed model is based on a simple pairwise event coreference model, which we haven't incorporate the structural information of the document. In the next chapter, we will propose a structure aware model for event coreference.

Type (counts)	Feature Name	Description
Discourse (5)	Sentence Distance	Number of sentences between two events mentions.
	Event Distance	Number of event mentions between two event mentions.
	Position	One event is in the title, or the first sentence.
Lexical (12)	Surface Similarity	Several string similarity measures computed between event mention headwords: Dice coefficient, edit distance, Jaro coefficient, lemma match and exact phrase match.
	Modifier Similarity	Dice coefficient similarity of the modifiers of the event mentions.
Syntactic (38)	Part Of Speech	Binary features for plurality, tense, noun or verb for the event mention head words.
	Dependency	The dependency label connecting the two event mentions.
	Negation	Whether two mention head words are both negated.
	Determiner	Whether the event mentions are modified by determiners
Semantic (16)	Coreference	Whether the predicates are in the same entity coreference cluster (only for nominal events).
	WordNet Similarity	Wu-Palmer similarity [161] of the headword pair.
	Senna Embeddings	Cosine similarity of event mention head word embeddings (Senna embeddings [48]).
	Distributional	Distributional similarity between predicates in Goyal et al. [88].
	Verb Ocean	Predicate word relations in "Verb Ocean" [44].
	Semantic Frame	Whether two predicates trigger the same semantic frame (extracted by Semafor [55]).
	Mention Type	Predicate word type (generated by IBM Sire [83]) match.
Arguments (34)	Surface	Dice coefficient and Wu-Palmer similarity between argument pairs;
	Coreference	Entity coreference between argument pairs; Numeric word match between the argument pairs.
	Existence	Whether each argument slot is instantiated.
	Location	Containment and alternatives name match between the location arguments based on geographical resources such as DBpedia [8] and GeoNames.

Table 3.4: List of features (with counts) in the pairwise model. Entity coreference are from the Stanford Entity Coreference Engine [111].

Chapter 4

Graph Based Event Coreference and Sequencing

4.1 Introduction

The rich relations among the textual event mentions help connect them. The mentions then collectively convey the meaning of the narrative. In the previous chapter, we introduce a pairwise approach for event coreference. However, the simple pairwise model does not consider the structural constraints. In this chapter, we propose a graph based approach, and apply them to two different types of relation: **Event Hopper Coreference (EH)** and **Event Sequencing (ES)**.

Event Hopper Coreference: In this chapter, we use the dataset of the TAC-KBP dataset, that defines a new **Event Hopper** Coreference task [145]: Two event mentions are considered coreferent if they refer to the conceptually same underlying event, even if their arguments are not strictly identical. For example, mentions that share similar temporal and location scope, though not necessarily the same expression, are considered to be coreferent (*Attack in Baghdad on Thursday* vs. *Bombing in the Green Zone last week*). This means that the event arguments of coreferential events mentions can be non-coreferential (18 killed vs. dozens killed), as long as they refer to the same event, judging from the available evidence.

Event Sequencing: The coreference relations build up events from scattered mentions. On the basis of events, various other types of relations can then be established between them. The Event Sequencing task studies one such relation. The task is motivated by Schank's *scripts* [185], which suggests that human organize information through procedural data structures, reassembling sequences of events. For example, the list of verbs *order, eat, pay, leave* may trigger the restaurant script. A human can conduct reasoning with a typical ordering of these events based on common sense (e.g., *order* should be the first event, *leave* should be the last event).

The ES task studies how to group and order events from text documents belonging to the same script. Figure 4.1 shows some annotation examples. Conceptually, event sequencing relations hold between the events, while coreference relations hold between textual event mentions. Given a document, the ES task requires systems to identify events within the same script and classify their inter-relations. These relations can be represented as labeled Directed Acyclic Graphs (DAGs).

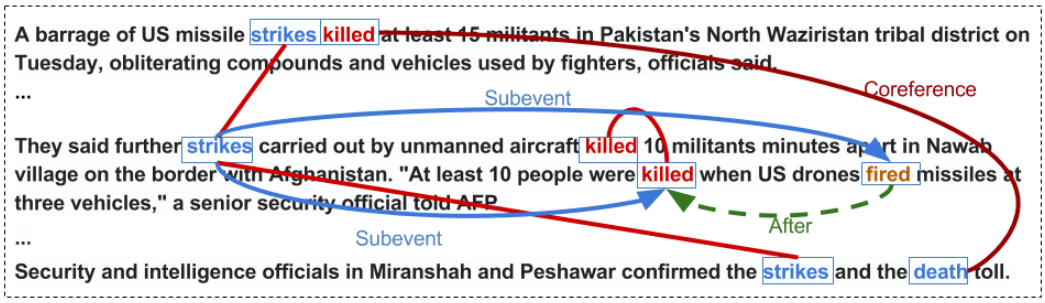

Figure 4.1: Example of Event Coreference and Sequence relations. Red lines are coreference links; solid blue arrows represent Subevent relations; dotted green arrows represent After relations. This figure is taken from the dataset annotated by annotators, the links are different from Fig. 1.2 annotated by the authors.

There are two types of relations[1]: **After** relations connect events following script orders (e.g. *order* followed by *eating*); **Subevent** relations connect events to a larger event that contains them. In this paper, we focus only on the **After** relations.

Since script-based understanding is built in the ES task, it has some unique properties comparing to pure temporal ordering: 1. event sequences from different scripts provide separate logical divisions of text, while temporal ordering considers all events to lie on a single timeline; 2. temporal relations for events occurring at similar time points may be complicated. Script-based relations may alleviate the problem. For example, if a `bombing` `kills` some people, the temporal relation of the `bombing` and `kill` may be "inclusion" or "after". This is considered an **After** relation in ES because `bombing` causes the `kill`ing.

For structure prediction, decoding — recovering the complex structure from local decisions — is one of the core problems. The most successful decoding algorithm for coreference nowadays is mention ranking based [21, 65, 113]. These models rank the antecedents (mentions that appear earlier in discourse) and recover the full coreference clusters from local decisions. However, unlike coreference relations, sequencing relations are directed. Coreference decoding algorithms cannot be directly applied to such relations (§4.3.1). To solve this problem, we propose a unified graph-based framework that tackles both event coreference and event sequencing. Our method achieves state-of-the-art results on the event coreference task (§4.4.4) and beats an informed baseline on the event sequencing task (§4.4.5). Finally, we analyze the results and discuss the difficult challenges for both tasks (§4.5). Detailed definitions of these tasks can be found in the corresponding task documents[2].

[1]Detailed definition of relations can be found in http://cairo.lti.cs.cmu.edu/kbp/2016/after/
[2]`http://cairo.lti.cs.cmu.edu/kbp/2017/event/documents`

4.2 Related Work

Many researchers have worked on event coreference tasks since Humphreys et al. [105]. In the previous chapter, we have summarized a variety of work under different settings on event coreference. Recent advances in event coreference have been promoted by the standardized annotation corpora. A lot of work is conducted on the popular ACE corpus [36, 37, 39, 40, 184]. Unlike the TAC KBP setting, the definition of event coreference in the ACE corpus requires strict argument matching.

In this chapter, we mainly follow the TAC-KBP event nugget tasks [143]. There is a small but growing amount of work on conducting event coreference on the TAC-KBP datasets [128, 129, 164]. The TAC dataset uses a relaxed coreference definition comparing to other corpora, requiring two event mentions to intuitively refer to the same real-world event despite differences of their participants.

For event sequencing, there are few supervised methods on script-like relation classification due to the lack of data. To the best of our knowledge, the only work in this direction is by Araki et al. [5]. This work focuses on the other type of relations in the event sequencing task: **Subevent** relations. There is also a rich literature on unsupervised script induction [31, 42, 79, 166, 178] that extracts scripts as a type of common-sense knowledge from raw documents. The focus of this work is to make use of massive collections of text documents to mine event co-occurrence patterns. In contrast, our work focuses on parsing the detailed relations between event mentions in each document.

Another line of work closely related to event sequencing is to detect other temporal relations between events. Recent computational approaches for temporal detection are mainly conducted on the TimeBank corpus [170]. There have been several studies on building automatic temporal reasoning systems [30, 59, 201]. In comparison, the Event Sequencing task is motivated by the Script theory, which places more emphasis on common-sense knowledge about event chronology.

4.3 Model

4.3.1 Graph-Based Decoding Model

In the Latent Antecedent Tree (LAT) model popularly used for entity coreference decoding [21, 74], each node represents an event mention and each arc a coreference relation, and new mentions are connected to some past mention considered most similar. Thus the LAT model represents the decoding structure as a tree. This can represent any coreference cluster, because coreference relations are by definition equivalence relations[3].

In contrast, tree structures cannot always fully cover an Event Sequence relation graph, because 1. the After links are directed, not symmetric, and 2. multiple event nodes can link to one node, resulting in multiple parents.

To solve this problem, we extend the LAT model and propose its graph version, namely the Latent Antecedent Graph (LAG) model. Figure 4.2 contrast LAT and LAG with decoding examples. The left box shows two example decoded trees in LAT, where each node has one single

[3]An equivalence relation is reflexive, symmetric and transitive.

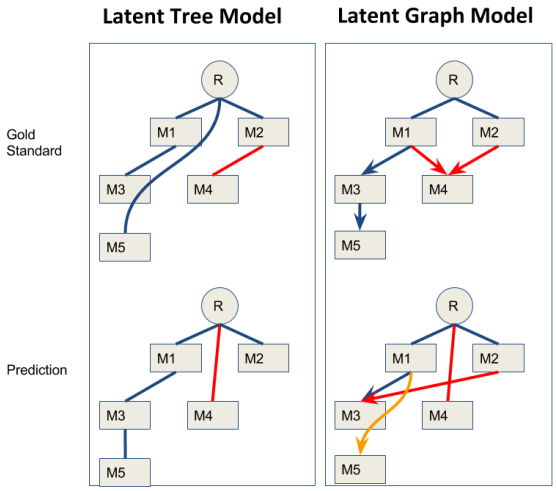

Figure 4.2: Latent Tree Model (left): tree structure formed by undirected links. Latent Graph Model (right): a DAG form by directed links. Dashed red links highlight the discrepancy between prediction and gold standard. The dotted yellow link (bottom right) can be inferred from other links.

parent. The right box shows two example decoded trees in LAG, where each node can be linked to multiple parents.

Formally, we define the series of (pre-extracted) event mentions of the document as $M = \{m_0, m_1, \ldots, m_n\}$, following their discourse order. m_0 is an artificial root node preceding all mentions. For each mention m_j, let $A_j = \{m_0, m_1, \ldots, m_{j-1}\}$ be the set of its potential antecedents: Let $\mathcal{A}$ denotes the set of antecedents for all the mentions in the sequence $\{A_0, A_1, \ldots, A_n\}$. The two tasks in question can be considered as finding the appropriate antecedent(s) from $\mathcal{A}$. Similarly, we define the gold antecedent set $\tilde{\mathcal{A}} = \{\tilde{A}_0, \tilde{A}_1, \ldots, \tilde{A}_n\}$, where $\tilde{A}_i$ represent the set of antecedents of m_i allowed by the gold standard. In the coreference task, $\tilde{A}_i$ contains all antecedents that are coreferent with m_i. In the sequencing task, $\tilde{A}_i$ contains all antecedents that have an $After$ relation to m_i.

We can now describe the decoding process. We represent each arc as $\langle m_i, m_j, r\rangle (i < j)$, where r is the relation name. The relation direction can be specified in the relation name r (e.g. r can be *after.forward* or *after.backward*). Further, an arc from the root node m_0 to node m_j represents that m_j does not have any antecedent. The score of the arc is the dot product between the weight parameter $\vec{w}$ and a feature vector $\Phi(\langle m_i, m_j, r\rangle)$, where Φ is an arc-wise feature function. The decoded graph z can be determined by a set of binary variables $\vec{z}$, where $\vec{z}_{ijr} = 1$ if there is an arc $\langle m_i, m_j, r\rangle$ or 0 otherwise. The final score of z is the sum of scores of all arcs:

$$score(z) = \sum_{i,j,r} \vec{z}_{ijr} \vec{w} \cdot \Phi(\langle m_i, m_j, r\rangle) \tag{4.1}$$

The decoding step is to find the output $\hat{z}$ that maximizes the scoring function:

$$\hat{z} = \arg \max_{z \in \mathcal{Z}(\mathcal{A})} score(z) \tag{4.2}$$

where $\mathcal{Z}(\mathcal{A})$ denotes all possible decoding structures given the antecedent sets $\mathcal{A}$. It is useful to note that the decoding step can be applied in the same way to the gold antecedent set $\tilde{\mathcal{A}}$.

Algorithm 2 shows the Passive-Aggressive training algorithm [50] used in our decoding framework. Line 7 decodes the maximum scored structure from all possible gold standard structures using the current parameters $\vec{w}$. Intuitively, this step tries to find **the "easiest" correct graph** — the correct graph with the highest score — for the current model. Several important components remain unspecified in algorithm 2: (1) the decoding step (line 5, 7); (2) the match criteria: whether to consider the system decoding structure as correct (line 6); (3) feature delta: computation of feature difference (line 8); (4) loss computation (line 9). We detail the actual implementation of these steps in §4.3.1.

Algorithm 2 PA algorithm for training

Input: Training data D, number of iterations T

1:

Output: Weight vector $\vec{w}$

2:

3: $\vec{w} = \vec{0}$;

4: $\langle \mathcal{A}, \tilde{\mathcal{A}} \rangle \in D$;

5: **for do** $t \leftarrow 1..T$ $\hat{z} = \arg\max_{\mathcal{Z}(\mathcal{A})} score(z)$

6: **if** $\neg Match(\hat{z}, \tilde{\mathcal{A}})$ **then**

7: $\tilde{z} = \arg\max_{\mathcal{Z}(\tilde{\mathcal{A}})} score(z)$

8: $\Delta = FeatureDelta(\tilde{z}, \hat{z})$

9: $\tau = \frac{loss(\tilde{z}, \hat{z})}{||\Delta||^2}$

10: $w = w + \tau\Delta$

return w

Minimum Decoding Structure

Similar to the LAT model, there may be many decoding structures representing the same configuration. In LAT, since there is exactly one link per node, the number of links in different decoding structures is the same, hence comparable. In LAG, however, one node is allowed to link to multiple antecedents, creating a potential problem for decoding. For example, consider the sequence $m_1 \xrightarrow{\text{after}} m_2 \xrightarrow{\text{after}} m_3$, both of the following structures are correct:

1. $\langle m_1, m_2, after \rangle$, $\langle m_2, m_3, after \rangle$
2. $\langle m_1, m_2, after \rangle$, $\langle m_2, m_3, after \rangle$, $\langle m_1, m_3, after \rangle$

However, the last relation in the second decoding structure can actually be inferred via transitivity. We do not intend to spend the modeling power on such cases. We empirically avoid such redundant cases by using the **transitive reduction graph** for each structure. For a directed acyclic graph, a transitive reduction graph contains the fewest possible edges that have the same reachability relation as the original graph. In the example above, structure 1 is a transitive reduction graph for structure 2. We call the decoding structures that corresponding to the reduction graphs as *minimum decoding structures*. For LAG, we further restrict $\mathcal{Z}(\mathcal{A})$ to contain only minimum decoding structures.

Training Details in Latent Antecedent Graph

In this section, we describe the decoding details for LAG. Note that if we enforce a single antecedent for each node (as in our coreference model), it falls back to the LAT model [21].

Decoding: We use a greedy **best-first decoder** [149], which makes a left-to-right pass over the mentions. The decoding step is the same for line 5 and 7. The only difference is that we will use gold antecedent set ($\tilde{\mathcal{A}}$) at line 7. For each node m_j, we keep all links that score higher than the root link $\langle 0, m_j, r \rangle$.

Cycle and Structure Check: Incremental decoding a DAG may introduce cycles to the graph, or violate the minimum decoding structure criterion. To solve this, we maintain a set $R(m_i)$ that is reachable from m_i during the decoding process. We reject a new link ($\langle m_j, m_i \rangle$ if $m_j \in R(m_i)$) to avoid cycles. We also reject a redundant link ($\langle m_i, m_j \rangle$ if $m_j \in R(m_i)$) to keep a minimum decoding structure. Our current implementation is greedy, we leave investigations of search or global inference based algorithms to future work.

Selecting the Latent Event Mention Graph: Note that sequence relations are on the event level. Given a unique event graph, it may still correspond to multiple mention graphs. In our implementation, we use a minimum set of event mentions to represent the full event graph by taking one single mention from each event. Following the "easiest" intuition, we select the single mention that will result in the highest score given the current feature weight w.

Match Criteria: We consider two graphs to match when their inferred graphs are the same. The inferred graph is defined by taking the transitive closure of the graph and propagate the links through the coreference relations. For example, in Figure 4.1, the mention `fired` will be linked to two `killed` mentions after propagation.

Feature Delta: In structural perceptron training [47], the weights are updated directly by the feature delta. For all the features $\tilde{f}$ of the gold standard graph $\tilde{z}$ and features $\hat{f}$ of a decoded graph $\hat{z}$, the feature delta is simply: $\Delta = \tilde{f} - \hat{f}$. However, a decoded graph may contain links that are not directly presented but inferable from the gold standard graph. For example, in Figure 4.2, the prediction graph has a link from $M5$ to $M1$ (the orange arc), which is absent but inferable from the gold standard tree. If we keep these links when computing Δ, the model does not converge well. We thus remove the features on the inferable links from $\hat{f}$ when computing Δ.

Loss: We define the loss to be the number of different edges in two graphs. Following Björkelund and Kuhn [21], we further penalize erroneous root attachment: an incorrect link to the root m_0 adds the loss by 2. For example, in Figure 4.2 the prediction graph (bottom right) incorrectly links m_4 to Root and misses a link to m_3, which cause a total loss of 3. In addition, to be consistent with the feature delta computation, we do not compute loss for predicted links that are inferable from the gold standard.

4.3.2 Features

Event Coreference Features

For event coreference, we design a simple feature set to capture syntactic and semantic similarity of arcs. The main features are summarized in Table 4.1. In the TAC KBP 2015 coreference task setting, the event mentions are annotated with two attributes. There are 38 event types and subtype

Head	Headword token and lemma pair, and whether they are the same.
Type	The pair of event types, and whether they are the same.
Realis	The pair of realis types and whether they are the same.
POS	POS pair of the two mentions and whether they are the same.
Exact Match	Whether the 5-word windows of the two mentions match exactly.
Distance	Sentence distance between the two mentions.
Frame	Frame name pair of the two mentions and whether they are the same.
Syntactic	Whether a mention is the syntactic ancestor of another.

Table 4.1: Coreference Features. Parsing is done using Stanford CoreNLP [132]; frame names are produced by Semafor [56].

pairs (e.g., *Business.Merge-Org, Conflict.Attack*). There also 3 realis type: events that actually occurred are marked as *Actual*; events that are not specific are marked as *Generic*; other events such as future events are marked as *Other*. For these two attributes, we use the gold annotations in our feature sets.

Event Sequencing Features

An event sequencing system needs to determine whether the events are in the same script and order them. We design separate feature sets to capture these aspects: the Script Compatibility set considers whether mentions should belong to the same script; the Event Ordering set determines the relative ordering of the mentions. Our final features are the cross products of features from the following 3 sets.

1. **Surface-Based Script Compatibility**: these features capture whether two mentions are script compatible based on the surface information, including:
    - Mention headword pair.
    - Event type pair.
    - Whether two event mentions appear in the same cluster in Chambers's event schema database [32].
    - Whether the two event mentions share arguments, and the semantic frame name of the shared argument (produced by the Semafor parser [56]).
2. **Discourse-Based Script Compatibility**: these features capture whether two event mentions are related given the discourse context.
    - Dependency path between the two mentions.
    - Function words (words other than Noun, Verb, Adjective and Adverb) in between the

two mentions.

- The types of other event mentions between the two mentions.
- The sentence distance of two event mentions.
- Whether there are temporal expressions in the sentences of the two mentions, extracted from the AGM-TMP slot using a PropBank parser [198]

3. **Event Ordering**: this feature set tries to capture the ordering of events. We use the discourse ordering of two mentions (forward: the antecedent is the parent; backward: the antecedent is the child), and temporal ordering produced by CAEVO [30].

Taking the *after* arc from `fired` to `killed` in Figure 4.1 as an example, a feature after the cross product is: Event type pair is *Conflict.Attack* and *Life.Die*, discourse ordering is *backward*, and sentence distance is 0.

4.4 Experiments

4.4.1 Dataset

We conduct experiments on the dataset released in Text Analysis Coreference (TAC-KBP) 2017 Event Sequencing task (released by LDC under the catalog name LDC2016E130). This dataset contains rich event relation annotations, with event mentions and coreference annotated in TAC-KBP 2015, and additional annotations on Event Sequencing[4]. There are 158 documents in the training set and 202 in the test set, selected from general news articles and forum discussion threads. The event mentions are annotated with 38 type-subtype and 3 realis status (Actual, Generic, Other). Event Hopper, After, and Subevent links are annotated between event mentions. For all experiments, we develop our system and conduct ablation studies using 5-fold cross-validation on the training set, and report performance on the test set.

4.4.2 Baselines and Benchmarks

Coreference: we compare our event coreference system against the top performing systems from TAC-KBP 2015 (LCC, UI-CCG, and LTI). In addition, we also compare the results against two official baselines [143]: the Singleton baseline that put each event mention in its own cluster and the Match baseline that creates clusters based on mention type and realis status match.
Sequencing: This work is an initial attempt to this problem, so there is currently no comparable prior work on the same task. We instead compare with a baseline using event temporal ordering systems. We use a state-of-the-art temporal system named Caevo [30]. To make a fair comparison, we feed the gold standard event mentions to the system along with mentions predicted by Caevo[5]. However, since the script-style After links are only connected between mentions in the same script, directly using the output of Caevo produces very low precision. Instead, we run a stronger baseline: we take the gold standard script clusters and then only ask Caevo to predict links within these clusters (Oracle Cluster + Temporal).

[4]http://cairo.lti.cs.cmu.edu/kbp/2016/after/
[5]We keep the mentions predicted by Caevo because its inference may be affected by these mentions.

4.4.3 Evaluation Metrics

Evaluating Event Coreference: We evaluate our results using the official scorer provided by TAC-KBP, which uses 4 coreference metrics: *BLANC* [173], *MUC* [43], B^3 [10] and *CEAF-E* [130]. Following the TAC KBP task, systems are ranked using the average of these 4 metrics.
Evaluating Event Sequencing: The TAC KBP scorer evaluates event sequencing using the metric of the TempEval task [200, 202]. The TempEval metric calculates special precision and recall values based on the closure and reduction graphs:

$$Precision = \frac{|Response^- \cap Reference^+|}{|Response^-|} \quad Recall = \frac{|Reference^- \cap Response^+|}{|Reference^-|}$$

where $Response$ represents the After link graph from the system response and $Reference$ represents the After link graph from the gold standard. G^+ represents the graph closure for graph G and G^- represents the graph reduction for graph G. As preprocessing, relations are automatically propagated through coreference clusters (currently using gold standard clusters). The final score is the standard F-score: geometric mean of the precision and recall values.

4.4.4 Evaluation Results for Event Coreference

The test performance on Event Coreference is summarized in Table 4.2. Comparing to the top 3 coreference systems in TAC-KBP 2015, we outperform the best system by about 2 points absolute F-score on average. Our system is also competitive on individual metrics. Our model performs the best based on B^3 and CEAF-E, and is comparable to the top performing systems on MUC and BLANC.

Note that while the `Matching` baseline only links event mentions based on event type and realis status, it is very competitive and performs close to the top systems. This is not surprising since these two attributes are based on the gold standard. To take a closer look, we conduct an ablation study by removing the simple match features one by one. The results are summarized in Table 4.3. We observe that some features produce mixed results on different metrics: they provide improvements on some metrics but not all. This is partially caused by the different characteristics of different metrics. On the other hand, these features (parsing and frames) are automatically predicted, which make them less stable. Furthermore, the Frame features contain duplicate information to event types, which makes it less useful in this setting.

Besides the presented features, we have also designed features using event argument. However, we do not report the results since the argument features decrease the performance on all metrics.

4.4.5 Evaluation Results for Event Sequencing

The evaluation results on Event Sequencing is summarized in Table 4.4. Because the baseline system has access to the oracle script clusters, it produces high precision. However, the low recall value shows that it fails to produce enough After links. Our analysis shows that a lot of After relations are not indicated by clear temporal clues, but can only be solved with script knowledge. In Example 3, the baseline system is able to identify "fled" is after "ousted" from explicit marker

	B^3	CEAF-E	MUC	BLANC	AVG.
`Singleton`	78.10	68.98	0.00	48.88	52.01
`Matching`	78.40	65.82	**69.83**	76.29	71.94
LCC	82.85	74.66	68.50	**77.61**	75.69
UI-CCG	83.75	75.81	63.78	73.99	74.28
LTI	82.27	75.15	60.93	71.57	72.60
This work	**85.59**	**79.65**	67.81	77.37	**77.61**

Table 4.2: Test Results for Event Coreference with the `Singleton` and `Matching` baselines.

	B^3	CEAF-E	MUC	BLANC	AVG.
ALL	81.97	74.80	76.33	76.07	77.29
-Distance	81.92	74.48	76.02	77.55	77.50
-Frame	82.14	75.01	76.28	77.74	77.79
-Syntactic	81.87	74.89	75.79	76.22	77.19

Table 4.3: Ablation study for Event Coreference.

"after". However, it fails to identify that "extradited" is after "arrested", which requires knowledge about prototypical event sequences.

(3) Eight months after the [transport fled] Ivory Coast when Gbagbo, the former president, was [End.Position ousted] by the French military. Blé Goudé was subsequently [Jail arrested] in Ghana and [transport extradited] Megrahi,[Jail jailed] for [Attack killing] 270 people in 1988.[6]

In our error analysis, we noticed that our system produces a large number of relations due to coreference propagation. One single wrong prediction can cause the error to propagate.

Besides memorizing the mention pairs, our model also tries to capture script compatibility through discourse signals. To further understand how much these signals help, we conduct an ablation study of the features in the discoursed based compatibility features (see §4.3.2). Similarly, we remove each feature group from the full feature set one by one and observe the performance change.

The results are reported in Table 4.5. While most of the features only affect the performance by less than 1 absolute F1 score, the feature sets after removing *mention* or *sentences* show a significant drop in both precision and recall. This shows that discourse proximity is the most significant ones among these features. In addition, the *mention* feature set captures the following *explain away* intuition: the event mentions A and B are less likely to be related if there are similar mentions in between. One such example can be seen in Figure 4.1, the event mention `fired` is

[6]The small red text indicates the event type for each mention.

	Prec.	Recall	F-Score
Oracle Cluster+Temporal	**46.21**	8.72	14.68
Our Model	18.28	**16.91**	**17.57**

Table 4.4: Test results for event sequencing. The Oracle Cluster+Temporal system is running CAEVO on the Oracle Clusters.

	Prec.	Recall	F-Score	Δ
Full	37.92	36.79	36.36	
- Mention Type	32.78	29.81	30.07	6.29
- Sentence	33.90	30.75	31.00	5.36
- Temporal	37.21	36.53	35.81	0.55
- Dependency	38.18	36.44	36.23	0.13
- Function words	38.08	36.51	36.18	0.18

Table 4.5: Ablation Study for Event Sequencing.

more likely to relate to the closest `killed`, instead of the other `killed` in the first paragraph.

In addition, our performance on the development set is higher than the test set. Further analysis reveals two causes: 1. the coreference propagation step causes the scores to be very unstable, 2. our model only learns limited common sense ordering based on lexical pairs, which can easily overfit to the small training corpus. Since the annotation is difficult to scale, it is important to use methods to harvest script common sense knowledge automatically, as in the script induction work [31].

4.5 Discussion

4.5.1 Event Coreference Challenges

Although we have achieved good performance on event coreference, upon closer investigation we found that most of the coreference decisions are still made based on simple word/lemma matching (note that the type and realis baseline is as high as 0.72 F1 score). The system exploits little semantic information to resolve difficult event coreference problems. A major challenge is that our system is not capable of utilizing event arguments: in fact, Hasler and Orasan [98] found that only around 20% of the arguments in the same event slot are actually coreferent for coreferential event pairs in the ACE 2005 corpus. Furthermore, the TAC-KBP corpus uses a relaxed participant identity requirement for event coreference, which makes argument-based matching more difficult.

4.5.2 Event Sequencing Challenges

Our event sequencing performance is still low despite the introduction of many features. This task is inherently difficult because it requires a system to solve both the script clustering and event ordering tasks. The former task requires both common-sense knowledge and discourse reasoning. Reasoning is more important for long-term links since there are no explicit clues like prepositions and dependencies to be exploited. The ablation study shows that discourse features like sentence distance are more effective, which indicates that our model mainly relies on surface clues and has limited reasoning power.

Furthermore, we observe a strong locality property of After links by skimming the training data: most After link relations are found in a small local region. Since reasoning and coreference based propagation will accumulate local decisions, a system must be accurate on them.

The Ambiguous Boundary of a Script

Besides the above-mentioned challenges, a more fundamental problem is to define the boundary of scripts. Since the definition of scripts is only prototypical event sequences, the boundaries between them are not clear. In Example 3, the event `jailed` is considered to belong to a "Judicial Process" script and `killing` is considered to belong to an"Attack" script[7]. No link is annotated between these two mentions since they are considered to belong to different clusters, even though the "jailed" event is to punish the "killing". Therefore essentially, the current Event Sequencing task simply requires the system to fit these human defined boundaries. In principle, the "Judicial Process" script and the "Attack" script can form a larger script structure, on a higher hierarchical level.

While it is possible to manually define scripts and what kind of events they may contain specifically in a controlled domain, it is difficult to generalize the relations. Most previous work on script induction [31, 42, 79, 166, 178] treats scripts as statistical models where probabilities can be assigned, thereby avoiding the boundary problem. While the script boundaries may be application dependent, a possible solution may rely on the "Goals" in Schank's script theory. The Goal of a script is the final state expected (by the script protagonist) from the sequence of events. Goal oriented scripts may be able to help us explain whether `killing` and `jailed` should be separate: if we take the"killer" as the protagonist, the goal of "kill" is achieved at the point of the victim dying. We leave the investigation on proper theoretical justification to future work.

4.6 Conclusion

In this chapter, we present a unified graph framework to conduct event coreference and sequencing. We have achieved state-of-the-art results on event coreference and report the first attempt at event sequencing. While we only studied two types of relations, we believe the method can be adopted in broader contexts.

[7]Script names are taken from the annotation guideline: `http://cairo.lti.cs.cmu.edu/kbp/2016/after/annotation`

In general, analyzing event structure can bring new aspects of knowledge from text. For instance, Event Coreference systems can help group scattered information together. Understanding Event Sequencing can help clarify the discourse structure, which can be useful in other NLP applications, such as solving entity coreference problems [163]. However, in our investigation, we find that the linguistic theory and definitions for events are not adequate for the computational setting. For example, proper theoretical justification is needed to define event coreference, which should explain the problems, such as argument mismatches. In addition, we also need a theoretical basis for script boundaries. In the future, we will devote our effort to understanding the theoretical and computational aspects of events relations, and utilizing them for other NLP tasks.

Chapter 5

Identifying Missing Information as Hierarchical Structures

5.1 Introduction

Humans often omit information from utterances to avoid redundancy when the context is clear enough to resolve the missing parts. This poses challenges for natural language processing (NLP) solutions. Two example problems in this direction are Verb Phrase Ellipsis (VPE) and Implicit Argument Detection (IAD). Both can be viewed as forms of cross-sentence ellipsis resolution or cross-sentence anaphora problems, where we need to find a phrase to be anaphoric to the elliptical slot. The key difference is that the missing information in VPE is part of the predicate, while the missing information in IAD is the argument.

Verb Phrase Ellipsis (VPE) is the anaphoric process where a verbal constituent is partially or totally unexpressed but can be resolved through an antecedent in the context. Consider the following examples:

(1) His wife also [antecedent *works for the paper*], as **did** his father.

(2) In particular, Mr. Coxon says, businesses are [antecedent *paying out a smaller percentage of their profits and cash flow in the form of dividends*] than they **have** historically.

In example 1, the light verb **did** represents the verb phrase *works for the paper*; example 2 shows a much longer antecedent phrase, which also differs in tense from the elided one. Following Dalrymple (1991), we refer to the full verb expression as the ”antecedent” and the anaphor as the ”target.”

Implicit Argument Detection (IAD) is a sub-problem of event argument extraction, which focuses on finding arguments that are mentioned across sentence boundaries. This task resembles implicit semantic role labeling (SRL), where the goal is to find argument spans to fill the roles of event frames. Event arguments can extend beyond sentence boundaries, introducing non-local or implicit arguments at the document level. Consider the following examples:

(3) The new computer cost 3000 dollars, while the old one cost 1000 dollars. Nevertheless, he still **bought** the more expensive one.

(4) The new computer cost 3000 dollars, while the old one cost 1000 dollars. Therefore, he **bought** the cheaper one.

In these examples, the *money* argument for the *purchase* event, triggered by the word "bought," appears in the previous sentence. This demonstrates the need to identify predicates and associate them with multi-word phrases across sentence boundaries.

Both VPE and IAD involve resolving missing information not explicitly stated within the current sentence, thereby requiring context from surrounding sentences for accurate comprehension. Understanding and modeling these hierarchical structures are crucial for tasks that demand deep text comprehension.

A Pipeline Approach to Reduce the Search Space. Semantically, Verb Phrase Ellipsis (VPE) and Implicit Argument Detection (IAD) both pose similar problems, requiring the resolution of missing information not explicitly stated within the current sentence. While VPE deals with the omission of predicate parts, IAD focuses on the omission of arguments. Despite their differences, both problems can benefit from a similar approach to decode the hierarchical structure. One common challenge in both cross-sentence and document-level problems is that the search and decoding space is much larger than a sentence-level solution, such as those used in regular semantic role labeling.

A pipeline solution is a natural choice for such problems where we identify the predicates and then search for the spans. However, considering all possible candidate spans that may occur in any sentences, their quadratic number still poses significant computational challenges. Moreover, the span detection needs to be accurate since it affects the semantics. For example, in Example 1, if the antecedent phrase is "works," then the semantics of "did" becomes "general working" instead of "working for the paper." Hence, we decompose the computation hierarchically.

1. Identify the predicate[1].
2. Identify the head word of the phrase to be extracted: The head word often contains crucial information about the phrase, narrowing down the search space.
3. Identify the correct boundary of the phrase: This step ensures that the entire relevant span is accurately captured, maintaining the intended semantics.

Predicate Identification. For VPE, this involves detecting the ellipsis targets where the verb phrase is omitted. For IAD, it involves identifying the event triggers that require argument resolution across sentence boundaries.

Two-Step Approach for Argument Expansion. For VPE, the next step is to identify the antecedents for the detected ellipsis targets. This step is decomposed into two subtasks: first, recognizing potential antecedents, and second, verifying their suitability. For IAD, the process involves detecting implicit arguments by first identifying candidate head-words and then expanding them to full argument spans.

Head Identification: Given the challenges posed by the quadratic number of candidate spans in IAD, we adopt a two-step approach. We hypothesize that syntactical head-words contain sufficient information to fill argument roles. Thus, we first detect the head-words of the arguments.

[1]the predicate is sometimes given in certain task settings

Head-to-Span Expansion: Once the head-words are identified, we expand them to full argument spans. This reduces the candidate space from quadratic to linear, making the detection process more efficient. By combining these methodologies, we address both VPE and IAD within a unified framework that leverages hierarchical structures and anaphora resolution. Our integrated approach not only improves performance on each individual task but also enhances the overall understanding of text in natural language processing.

Datasets. The availability of annotated datasets is crucial for training and evaluating our models. For VPE, there are only a few small datasets. While Bos and Spenader [22] provide publicly available VPE annotations for Wall Street Journal (WSJ) news documents, the annotations created by Nielsen [155] include a more diverse set of genres (e.g., articles and plays) from the British National Corpus (BNC).we transform annotations from the British National Corpus (BNC) into the format used by Bos et al. (2011) for the Wall Street Journal (WSJ) documents. This unified format allows for better benchmarking and facilitates meaningful comparisons.

While incorporating implicit arguments requires corresponding annotations, few exists in most of the widely used event datasets, like ACE2005 [109, 209] and RichERE [110]. Ebner et al. [66] create the Roles Across Multiple Sentences (*RAMS*) dataset, which covers multi-sentence implicit arguments for a wide range of event and role types. They further develop a span-based argument linking model and achieve relatively high scores.

In this chapter, we will describe the approaches, models and features of both problems separately, but following the same hierarchical approach presented here.

5.2 Related Work

5.2.1 Related Work on Verb Phrase Ellipsis

Considerable work has been done on VPE in the field of theoretical linguistics: e.g., [53, 187]; yet there is much less work on computational approaches to resolving VPE.

Hardt [1992, 1997] presents, to our knowledge, the first computational approach to VPE. His system applies a set of linguistically motivated rules to select an antecedent given an elliptical target. Hardt [97] uses Transformation-Based Learning to replace the manually developed rules. However, in Hardt's work, the targets are selected from the corpus by searching for "empty verb phrases" (constructions with an auxiliary verb only) in the gold standard parse trees.

Nielsen [2005] presents the first end-to-end system that resolves VPE from raw text input. He describes several heuristic and learning-based approaches for target detection and antecedent identification. He also discusses a post-processing substitution step in which the target is replaced by a transformed version of the antecedent (to match the context). We do not address this task here because other VPE datasets do not contain relevant substitution annotations. Similar techniques are also described in Nielsen [2003, 2004, 2004].

Results from this prior work are relatively difficult to reproduce because the annotations on which they rely are inaccessible. The annotations used by Hardt [96] have not been made available, and those used by Nielsen [155] are not easily reusable since they rely on some particular tokenization and parser. Bos and Spenader [22] address this problem by annotating a new corpus

of VPE on top of the WSJ section of the Penn Treebank, and propose it as a standard evaluation benchmark for the task. Still it is desirable to use Nielsen's annotations on the BNC which contain more diverse text genres with more frequent VPE.

5.2.2 Related Work on Implicit Argument Identification

Implicit arguments have been under-explored in event extraction. Most of previous systems [38, 117, 150, 211] only consider local arguments in the same sentence of the event trigger. While incorporating implicit arguments requires corresponding annotations, few exists in most of the widely used event datasets, like ACE2005 [109, 209] and RichERE [110]. There are several annotation efforts for implicit arguments in SRL, including *G&C* [85, 86], *SemEval-2010* [179, 180], and *80Days* [72]. Yet most are performed with different ontologies such as Nombank (*G&C*) and FrameNet (*SemEval-2010* and *80Days*); on different domains (e.g. novels); and in smaller scales (*G&C* and *80Days* only cover 10 types of predicates). The lack of annotations poses challenges to train and transfer implicit argument models for event extraction.

5.3 Modeling Verb Phrase Ellipsis

We focus on the problems of target detection and antecedent identification as proposed by Nielsen [155]. We propose a refinement of these two tasks, splitting them into these three:

1. **Target Detection (T)**, where the subset of VPE targets is identified.
2. **Antecedent Head Resolution (H)**, where each target is linked to the head of its antecedent.
3. **Antecedent Boundary Determination (B)**, where the exact boundaries of the antecedent are determined from its head.

The following sections describe each of the steps in detail.

5.3.1 Target Detection

Since the VPE target is annotated as a single word in the corpus[2], we model their detection as a binary classification problem. We only consider modal or light verbs (*be*, *do*, *have*) as candidates, and train a logistic regression classifier ($\mathbf{Log}^T$) with the following set of binary features:

1. The POS tag, lemma, and dependency label of the verb, its dependency parent, and the immediately preceding and succeeding words.
2. The POS tags, lemmas and dependency labels of the words in the dependency subtree of the verb, in the 3-word window, and in the same-size window after (as bags of words).
3. Whether the subject of the verb appears to its right (i.e., there is subject-verb inversion).

[2]All targets in the corpus of Bos and Spenader [22] are single-word by their annotation guideline.

5.3.2 Antecedent Head Resolution

For each detected target, we consider as potential antecedent heads all verbs (including modals and auxiliaries) in the three immediately preceding sentences of the target word[3] as well as the sentence including the target word (up to the target[4]). This follows Hardt [95] and Nielsen [155].

We perform experiments using a logistic regression classifier (**Log**^{H}), trained to distinguish correct antecedents from all other possible candidates. The set of features are shared with the Antecedent Boundary Determination task, and are described in detail in Section 5.3.3.

However, a more natural view of the resolution task is that of a ranking problem. The gold annotation can be seen as a partial ordering of the candidates, where, for a given target, the correct antecedent ranks above all other candidates, but there is no ordering among the remaining candidates. To handle this specific setting, we adopt a ranking model with domination loss [57].

Formally, for each potential target t in the determined set of targets T, we consider its set of candidates C_t, and denote whether a candidate $c \in C_t$ is the antecedent for t using a binary variable a_{ct}. We express the ranking problem as a bipartite graph $\mathcal{G} = (V^+, V^-, E)$ where vertices represent antecedent candidates:

$$\begin{aligned} V^+ &= \{(t,c) \mid t \in T, c \in C_t, a_{ct} = 1\} \\ V^- &= \{(t,c) \mid t \in T, c \in C_t, a_{ct} = 0\} \end{aligned}$$

and the edges link the correct antecedents to the rest of the candidates for the same target[5]:

$$E = \{((t,c^+),(t,c^-)) \mid (t,c^+) \in V^+, (t,c^-) \in V^-\}$$

We associate each vertex i with a feature vector $\mathbf{x}_i$, and compute its score s_i as a parametric function of the features $s_i = g(\mathbf{w}, \mathbf{x}_i)$. The training objective is to learn parameters $\mathbf{w}$ such that each positive vertex $i \in V^+$ has a higher score than the negative vertices j it is connected to, $V_i^- = \{j \mid j \in V^-, (i,j) \in E\}$.

The combinatorial domination loss for a vertex $i \in V^+$ is 1 if there exists any vertex $j \in V_i^-$ with a higher score. A convex relaxation of the loss for the graph is given by [57]:

$$f(w) = \frac{1}{|V^+|} \sum_{i \in V^+} \log(1 + \sum_{j \in V_i^-} \exp(s_j - s_i + \Delta))$$

Taking $\Delta = 0$, and choosing $\mathbf{g}$ to be a linear feature scoring function $s_i = \mathbf{w} \cdot \mathbf{x_i}$, the loss becomes:

$$f(w) = \frac{1}{|V^+|} \sum_{i \in V^+} \log \sum_{j \in V_i^-} \exp(\mathbf{w} \cdot \mathbf{x}_j) - \mathbf{w} \cdot x_i$$

The loss over the whole graph can then be minimized using stochastic gradient descent. We will denote the ranker learned with this approach as **Rank**^{H}.

[3]Only 1 of the targets in the corpus of Bos and Spenader [22], has an antecedent beyond that window.
[4]Only 1% of the targets in the corpus are cataphoric.
[5]During training, there is always 1 correct antecedent for each gold standard target, with several incorrect ones.

Algorithm 3 Candidate generation

Input: a, the antecedent head
Input: t, the target
Output: B, the set of possible antecedent boundaries (begin, end)

$a_s \longleftarrow \text{SemanticHeadVerb}(a)$
$E \longleftarrow \{a_s\}$ *// the set of ending positions*
for $ch \in \text{RightChildren}(a_s)$ **do**
  $e \leftarrow \text{RightMostNode}(ch)$
  if $e < t \wedge \text{ValidEnding}(e)$ **then**
    $E \longleftarrow E \cup \{e\}$
$B \longleftarrow \emptyset$
for $e \in E$ **do**
  $B \longleftarrow B \cup \{(a, e)\}$
  if $a ==$ "be" **then**
    if $\text{IsVerb}(a+1)$ **then**
      $A \longleftarrow A \cup \{(a+1, e)\}$
    for $s \in \{a+1, a+2 \ldots e-1\}$ **do**
      if $\text{IsAdverb}(s) \wedge \text{IsVerb}(s+1)$ **then**
        $B \longleftarrow B \cup \{(s+1, e)\}$
return B

5.3.3 Antecedent Boundary Determination

From a given antecedent head, the set of potential boundaries for the antecedent, which is a complete or partial verb phrase, is constructed using Algorithm 3.

Informally, the algorithm tries to generate different valid verb phrase structures by varying the amount of information encoded in the phrase. To do so, it accesses the semantic head verb a_s of the antecedent head a (e.g., *paying* for *are* in Example 2), and considers the rightmost node of each right child. If the node is a valid ending (punctuation and quotation are excluded), it is added to the potential set of endings E. The set of valid boundaries B contains the cross-product of the starting position $S = \{a\}$ with E.

For instance, from Example 2, the following boundary candidates are generated for *are*:

- are paying
- are paying out
- are paying out a smaller percentage of their profits and cash flow
- are paying out a smaller percentage of their profits and cash flow in the form of dividends

We experiment with both logistic regression (**Log**^{B}) and ranking (**Rank**^{B}) models for this task. The set of features is shared with the previous task, and is described in the following section.

Antecedent Features

The features used for antecedent head resolution and/or boundary determination try to capture aspects of both tasks. We summarize the features in Table 5.1. The features are roughly grouped by their type. **Labels** features make use of the parsing labels of the antecedent and target; **Tree** features are intended to capture the dependency relations between the antecedent and target; **Distance** features describe distance between them; **Match** features test whether the context of the antecedent and target are similar; **Semantic** features capture shallow semantic similarity; finally, there are a few **Other** features which are not categorized.

On the last column of the feature table, we indicate the design purpose of the feature: head selection (H), boundary detection (B) or both (B&H). However, we use the full feature set for all three tasks.

5.4 Does Joint Modeling work for VPE?

Here we consider the possibility that antecedent head resolution and target detection should be modeled jointly (they are typically separate). The hypothesis is that if a suitable antecedent for a target cannot be found, the target itself might have been incorrectly detected. Similarly, the suitability of a candidate as antecedent head can depend on the possible boundaries of the antecedents that can be generated from it.

We also consider the possibility that antecedent head resolution and antecedent boundary determination should be modeled independently (though they are typically combined). We hypothesize that these two steps actually focus on different perspectives: the antecedent head resolution (**H**) focuses on finding the correct antecedent position; the boundary detection step (**B**) focuses on constructing a well-formed verb phrase. We are also aware that **B** might be helpful to **H**, for instance, a correct antecedent boundary will give us correct context words, that can be useful in determining the antecedent position.

We examine the joint interactions by combining adjacent steps in our pipeline. For the combination of antecedent head resolution and antecedent boundary determination (**H+B**), we consider simultaneously as candidates for each target the set of all potential boundaries for all potential heads. Here too, a logistic regression model (**Log**$^{H+B}$) can be used to distinguish correct `(target, antecedent start, antecedent end)` triplets; or a ranking model (**Rank**$^{H+B}$) can be trained to rank the correct one above the other ones for the same target.

The combination of target detection with antecedent head resolution (**T+H**) requires identifying the targets. This is not straightforward when using a ranking model since scores are only comparable for the same target. To get around this problem, we add a "null" antecedent head. For a given target candidate, the null antecedent should be ranked higher than all other candidates if it is not actually a target. Since this produces many examples where the null antecedent should be selected, random subsampling is used to reduce the training data imbalance. The "null" hypothesis approach is used previously in ranking-based coreference systems [64, 172].

Most of the features presented in the previous section will not trigger for the null instance, and an additional feature to mark this case is added.

The combination of the three tasks (**T+H+B**) only differs from the previous case in that

Type	Feature Description	Purpose
Labels	The POS tag and dependency label of the antecedent head	H
	The POS tag and dependency label of the antecedent's last word	B
	The POS tag and lemma of the antecedent parent	H
	The POS tag, lemma and dependency label of within a 3 word around around the antecedent	B
	The pair of the POS tags of the antecedent head and the target, and of their auxiliary verbs	H
	The pair of the lemmas of the auxiliary verbs of the antecedent head and the target	H
Tree	Whether the antecedent and the target form a comparative construction connecting by *so*, *as* or *than*	H&B
	The dependency labels of the shared lemmas between the parse tree of the antecedent and the target	H
	Label of the dependency between the antecedent and target (if exists)	H
	Whether the antecedent contains any descendant with the same lemma and dependency label as a descendant of the target.	H
	Whether antecedent and target are dependent ancestor of each other	H
	Whether antecedent and target share prepositions in their dependency tree	H
Distance	The distance in sentences between the antecedent and the target (clipped to 2)	H
	The number of verb phrases between the antecedent and the target (clipped to 5)	H
Match	Whether the lemmas of the heads, and words in the the window (=2) before the antecedent and the target match respectively	H
	Whether the lemmas of the ith word before the antecedent and $i-1$th word before the target match respectively (for $i \in \{1,2,3\}$, with the 0th word of the target being the target itself)	H&B
Semantic	Whether the subject of the antecedent and the target are coreferent	H
Other	Whether the lemma of the head of the antecedent is *be* and that of the target is *do* (be-do match, used by Hardt and Nielsen)	H
	Whether the antecedent is in quotes and the target is not, or vice versa	H&B

Table 5.1: Antecedent Features

all antecedent boundaries are considered as candidates for a target, in addition to the potential antecedent heads.

	Documents		VPE Instances	
	Train	Test	Train	Test
WSJ	1999	500	435	119
BNC	12	2	641	204

Table 5.2: Corpus statistics

5.5 VPE Experiments

5.5.1 Datasets

We conduct our experiments on two datasets (see Table 5.2 for corpus counts). The first one is the corpus of Bos and Spenader [22], which provides VPE annotation on the WSJ section of the Penn Treebank. Bos and Spenader [22] propose a train-test split that we follow[6].

To facilitate more meaningful comparison, we converted the sections of the British National Corpus annotated by Nielsen [155] into the format used by Bos and Spenader [22], and manually fixed conversion errors introduced during the process[7] (Our version of the dataset is publicly available for research[8].) We use a train-test split similar to Nielsen [155][9].

5.5.2 Evaluation

We evaluate and compare our models following the metrics used by Bos and Spenader [22].

VPE target detection is a per-word binary classification problem, which can be evaluated using the conventional precision (Prec), recall (Rec) and F1 scores.

Bos and Spenader [22] propose a token-based evaluation metric for antecedent selection. The antecedent scores are computed over the correctly identified tokens per antecedent: precision is the number of correctly identified tokens divided by the number of predicted tokens, and recall is the number of correctly identified tokens divided by the number of gold standard tokens. Averaged scores refer to a "macro"-average over all antecedents.

Finally, in order to asses the performance of antecedent head resolution, we compute precision, recall and F1 where credit is given if the proposed head is included inside the golden antecedent boundaries.

[6]Section 20 to 24 are used as test data.

[7]We also found 3 annotation instances that could be deemed errors, but decided to preserve the annotations as they were.

[8]`https://github.com/hunterhector/VerbPhraseEllipsis`

[9]Training set is CS6, A2U, J25, FU6, H7F, HA3, A19, A0P, G1A, EWC, FNS, C8T; test set is EDJ, FR3

5.5.3 Baselines and Benchmarks

We begin with simple, linguistically motivated baseline approaches for the three subtasks. For target detection, we re-implement the heuristic baseline used by Nielsen [155]: take all auxiliaries as possible candidates and eliminate them using part-of-speech context rules (we refer to this as $\mathbf{Pos}^T$). For antecedent head resolution, we take the first non-auxiliary verb preceding the target verb. For antecedent boundary detection, we expand the verb into a phrase by taking the largest subtree of the verb such that it does not overlap with the target. These two baselines are also used in Nielsen [155] (and we refer to them as $\mathbf{Prev}^H$ and $\mathbf{Max}^B$, respectively).

To upper-bound our results, we include an oracle for the three subtasks, which selects the highest scoring candidate among all those considered. We denote these as $\mathbf{Ora}^T$, $\mathbf{Ora}^H$, $\mathbf{Ora}^B$.

We also compare to the current state-of-the-art target detection results as reported in Nielsen [155] on the BNC dataset ($\mathbf{Nielsen}^T$)[10].

5.6 VPE Results

The results for each one of the three subtasks in isolation are presented first, followed by those of the end-to-end evaluation. We have not attempted to tune classification thresholds to maximize F1.

5.6.1 Target Detection

Table 5.3 shows the performance of the compared approaches on the Target Detection task. The logistic regression model $\mathbf{Log}^T$ gives relatively high precision compared to recall, probably because there are so many more negative training examples than positive ones. Despite a simple set of features, the F1 results are significantly better than Nielsen's baseline $\mathbf{Pos}^T$.

Notice also how the oracle $\mathbf{Ora}^T$ does not achieve 100% recall, since not all the targets in the gold data are captured by our candidate generation strategy. The loss is around 7% for both corpora.

The results obtained by the joint models are low on this task. In particular, the ranking models $\mathbf{Rank}^{T+H}$ and $\mathbf{Rank}^{T+H+B}$ fail to predict any target in the WSJ corpus, since the null antecedent is always preferred. This happens because joint modeling further exaggerates the class imbalance: the ranker is asked to consider many incorrect targets coupled with all sorts of hypothesis antecedents, and ultimately learns just to select the null target. Our initial attempts at subsampling the negative examples did not improve the situation. The logistic regression models $\mathbf{Log}^{T+H}$ and $\mathbf{Log}^{T+H+B}$ are most robust, but still their performance is far below that of the pure classifier $\mathbf{Log}^T$.

5.6.2 Antecedent Head Resolution

Table 5.4 contains the performance of the compared approaches on the Antecedent Head Resolution task, assuming oracle targets ($\mathbf{Ora}^T$).

[10]The differences in the setup make the results on antecedent resolution not directly comparable.

	WSJ			BNC		
	Prec	Rec	F1	Prec	Rec	F1
Ora^{T}	100.00	93.28	96.52	100.00	92.65	96.18
Log^{T}	80.22	61.34	69.52	80.90	70.59	75.39
Pos^{T}	42.62	43.7	43.15	35.47	35.29	35.38
Log^{T+H}	23.36	26.89	25.00	12.52	38.24	18.86
Rank^{T+H}	0.00	0.00	0.00	15.79	5.88	8.57
Log^{T+H+B}	25.61	17.65	20.90	21.50	32.35	25.83
Rank^{T+H+B}	0.00	0.00	0.00	16.67	11.27	13.45
Nielsen^{T}	—	—	—	72.50	72.86	72.68

Table 5.3: Results for Target Detection

	WSJ			BNC		
	Prec	Rec	F1	Prec	Rec	F1
Ora^{H}	94.59	88.24	91.30	79.89	74.02	76.84
Rank^{H}	70.27	65.55	67.83	52.91	49.02	50.89
Prev^{H}	67.57	63.03	65.22	39.68	36.76	38.17
Log^{H}	59.46	55.46	57.39	38.62	35.78	37.15
Rank^{H+B}	68.47	63.87	66.09	51.85	48.04	49.87
Log^{H+B}	39.64	36.97	38.26	30.16	27.94	29.01

Table 5.4: Results for Antecedent Head Resolution

First, we observe that even the oracle **Ora**^{H} has low scores on the BNC corpus. This suggests that some phenomena beyond the scope of those observed in the WSJ data appear in the more general corpus (we developed our system using the WSJ annotations and then simply evaluated on the BNC test data).

Second, the ranking-based model **Rank**^{H} consistently outperforms the logistic regression model **Log**^{H} and the baseline **Prev**^{H}. The ranking model's advantage is small in the WSJ, but much more pronounced in the BNC data. These improvements suggest that indeed, ranking is a more natural modeling choice than classification for antecedent head resolution.

Finally, the joint resolution models **Rank**$^{H+B}$ and **Log**$^{H+B}$ give poorer results than their single-task counterparts, though **Rank**$^{H+B}$ is not far behind **Rank**^{H}. Joint modeling requires more training data and we may not have enough to reflect the benefit of a more powerful model.

5.6.3 Antecedent Boundary Determination

Table 5.5 shows the performance of the compared approaches on the Antecedent Boundary Determination task, using the soft evaluation scores (the results for the strict scores are omitted for brevity, but in general look quite similar). The systems use the output of the oracle targets (**Ora**^{T}) and antecedent heads (**Ora**^{H}).

Regarding boundary detection alone, the logistic regression model **Log**^{B} outperforms the ranking model **Rank**^{B}. This suggests that boundary determination is more a problem of determining the compatibility between target and antecedent extent than one of ranking alternative boundaries. However, the next experiments suggest this advantage is diminished when gold targets and antecedent heads are replaced by system predictions.

Non-Gold Antecedent Heads

Table 5.6 contains Antecedent Boundary Determination results for systems which use oracle targets, but system antecedent heads. When **Rank**^{H} or **Log**^{H} are used for head resolution, the difference between **Log**^{B} and **Rank**^{B} diminishes, and it is even better to use the latter in the BNC corpus. The models were trained with gold annotations rather than system outputs, and the ranking model is somewhat more robust to noisier inputs.

On the other hand, the results for the joint resolution model **Rank**$^{H+B}$ are better in this case than the combination of **Rank**^{H}**+Rank**^{B}, whereas **Log**$^{H+B}$ performs worse than any 2-step combination. The benefits of using a ranking model for antecedent head resolution seem thus to outperform those of using classification to determine its boundaries.

5.6.4 End-to-End Evaluation

Table 5.7 contains the end-to-end performance of different approaches, using the soft evaluation scores.

The trends we observed with gold targets are preserved: approaches using the **Rank**^{H} maintain an advantage over **Log**^{H}, but the improvement of **Log**^{B} over **Rank**^{B} for boundary determination is diminished with non-gold heads. Also, the 3-step approaches seem to perform slightly better

	WSJ			BNC		
	Prec	Rec	F1	Prec	Rec	F1
Ora^{B}	95.06	88.67	91.76	85.79	79.49	82.52
Log^{B}	89.47	83.46	86.36	81.10	75.13	78.00
Rank^{B}	83.96	78.32	81.04	75.68	70.12	72.79
Max^{B}	78.97	73.66	76.22	73.70	68.28	70.88

Table 5.5: Soft results for Antecedent Boundary Determination

	WSJ			BNC		
	Prec	Rec	F1	Prec	Rec	F1
Ora^{H}**+Ora**^{B}	95.06	88.67	91.76	85.79	79.49	82.52
Rank^{H}**+Log**^{B}	64.11	59.8	61.88	47.04	43.58	45.24
Rank^{H}**+Rank**^{B}	63.90	59.6	61.67	49.11	45.5	47.24
Log^{H}**+Log**^{B}	53.49	49.89	51.63	34.77	32.21	33.44
Log^{H}**+Rank**^{B}	53.27	49.69	51.42	36.26	33.59	34.88
Rank^{H+B}	67.55	63.01	65.20	50.68	46.95	48.74
Log^{H+B}	40.96	38.20	39.53	30.00	27.79	28.85

Table 5.6: Soft results for Antecedent Boundary Determination with non-gold heads

	WSJ			BNC		
	Prec	Rec	F1	Prec	Rec	F1
Ora^T**+Ora**^{H}**+Ora**^{B}	95.06	88.67	91.76	85.79	79.49	82.52
Log^T**+Rank**^{H}**+Rank**^{B}	52.68	40.28	45.65	43.03	37.54	40.10
Log^T**+Rank**^{H}**+Log**^{B}	52.82	40.40	45.78	40.21	35.08	37.47
Log^T**+Log**^{H}**+Rank**^{B}	49.45	37.82	42.86	33.12	28.90	30.86
Log^T**+Log**^{H}**+Log**^{B}	49.41	37.79	42.83	31.32	27.33	29.19
Pos^T**+Prev**^{H}**+Max**^{B}	19.04	19.52	19.27	12.81	12.75	12.78
Log^T**+Rank**$^{H+B}$	54.82	41.92	47.51	41.86	36.52	39.01
Log^T**+Log**$^{H+B}$	38.85	29.71	33.67	26.11	22.78	24.33

Table 5.7: Soft end-to-end results

than the 2-step ones. Together with the fact that the smaller problems are easier to train, this appears to validate our decomposition choice.

5.7 Discussion of VPE Results

In this chapter we have explored a decomposition of Verb Phrase Ellipsis resolution into subtasks, which splits antecedent selection in two distinct steps. By modeling these two subtasks separately with two different learning paradigms, we can achieve better performance then doing them jointly, suggesting they are indeed of different underlying nature.

Our experiments show that a logistic regression classification model works better for target detection and antecedent boundary determination, while a ranking-based model is more suitable for selecting the antecedent head of a given target. However, the benefits of the classification model for boundary determination are reduced for non-gold targets and heads. On the other hand, by separating the two steps, we lose the potential joint interaction of them. It might be possible to explore whether we can bring the benefits of the two side: use separate models on each step, but learn them jointly. We leave further investigation of this to future work.

We have also explored jointly training a target detection and antecedent resolution model, but have not been successful in dealing with the class imbalance inherent to the problem.

Our current model adopts a simple feature set, which is composed mostly by simple syntax and lexical features. It may be interesting to explore more semantic and discourse-level features in our system. We leave these to future investigation.

All our experiments have been run on publicly available datasets, to which we add our manually aligned version of the VPE annotations on the BNC corpus. We hope our experiments, analysis, and more easily processed data can further the development of new computational approaches to the problem of Verb Phrase Ellipsis resolution.

(a)	The new computer cost 3000 dollars, while the old one cost 1000 dollars. Nevertheless, he still **bought** the more expensive one.
(b)	The new computer cost 3000 dollars, while the old one cost 1000 dollars. Therefore, he **bought** the cheaper one.

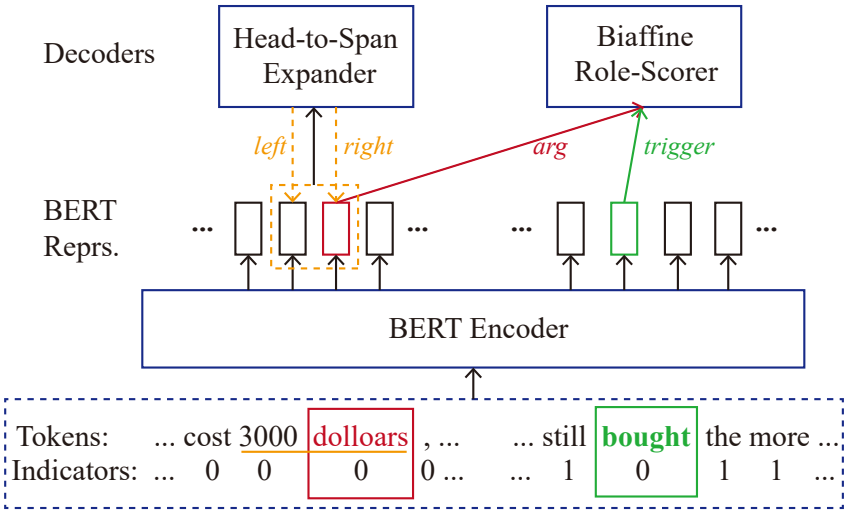

Figure 5.1: Examples of implicit arguments and model illustration. The **bold** text indicates the trigger word for the *purchase* event, while the underlined text indicates its non-local "*money*" argument in the previous sentence. Our model first detects the head-word "dollars", and then expands it to the whole span.

5.8 Modeling Implicit Argument Identification

The goal of event argument detection is to create labeled links between argument spans and the predicate (event trigger). Recent state-of-the-art solutions for sentence-level SRL perform the detection in an end-to-end setting, such as span-based [99, 158], and sequence labeling models [100, 186]. However, span-based models face great challenges when considering arguments across sentence boundaries, since the computational complexity of such models grows quadratically to deal with $O(N^2)$ span candidates given N tokens. While traditional sequence labeling models can run in linear-time, they are less flexible and extensible in complex scenarios like overlapping mentions and multiple roles for one mention.

In this work, we take a two-step approach that decomposes the problem explicitly into two sub-problems, based on the hypothesis that head-words can usually capture the information of the mention spans. Figure 5.1 illustrates the three main modules of our model: 1) BERT-based Encoder, 2) Argument Head-Word Detector, and 3) Head-to-span Expander.

5.8.1 BERT-based Encoder

Our encoding module is a BERT-based contextualized encoder. The input contains a predicate word (or occasionally a span), which triggers an event, together with its multi-sentence context. We refer to the sentence containing the event trigger as the *center sentence*. We concatenate the tokens within the 5-sentence window (the window size used in *RAMS* annotation) of the center sentences, and feed them to BERT to obtain the contextual representation $\mathbf{e}$ of each token. In addition, we add special `token_type_ids` indicators: tokens of the event trigger are assigned 0, other tokens in the center sentence get 1, and tokens in surrounding sentences get 0[11]. We only adopt the indicators when fine-tuning BERT, since the pre-trained BERT originally uses them as segment ids.

5.8.2 Argument Head-word Detector

Instead of directly deciding argument spans, we first identify the head-words of the arguments. The hypothesis is that the head-word is able to represent the meaning of the whole span. In this way, this sub-problem mimics a token-pairwise dependency-parsing problem. Following [61, 62], we adopt a biaffine module to calculate $\Pr_r(p, c)$: the probability of a candidate word c filling an argument role r in the frame governed by a predicate p. We first take the contextualized representations of the candidate ($\mathbf{e}_c$) and the predicate ($\mathbf{e}_p$), which are calculated by BERT as described in §5.8.1. "Biaffine_r" further gives the pairwise score based on these representations, and $\Pr_r(p, c)$ is then given by softmax with the scores:

$$\Pr_r(p, c) = \frac{\exp \text{Biaffine}_r(\mathbf{e}_p, \mathbf{e}_c)}{\sum_{c' \in \mathcal{C} \cup \{\epsilon\}} \exp \text{Biaffine}_r(\mathbf{e}_p, \mathbf{e}_{c'})}$$

where the normalization is done over the argument candidate set $\mathcal{C}$ (or null ϵ, whose score is fixed to 0) for each role, following [66, 158]. During training, we use the cross-entropy loss to guide the network to pick head-words of gold arguments (or ϵ if there are no arguments for this role). If there are multiple arguments for one role, we view them as individual instances and sum the losses. At inference time, we simply pick the maximumly-scored argument (or ϵ) for each role.

5.8.3 Head-to-span Expander

The second module expands each head-word of the argument to its full span. We view it as a combination of left and right boundary classification problems. Taking the left-expanding scenario (L) as example, for each head-word h, we generate a set of candidate spans by adding words one by one on the left up to K words (we empirically set $K = 7$), and calculate the probability of word b being the boundary as follow:

$$\Pr_L(h, b) = \frac{\exp \text{MLP}_L(\mathbf{e}_h, \mathbf{e}_b)}{\sum_{b' \in (h-K, h]} \exp \text{MLP}_L(\mathbf{e}_h, \mathbf{e}_{b'})}$$

[11]We overload 0 because pre-trained BERT only has two types of `token_type_id`. Nevertheless, the trigger words are still distinguishable since they appear inside center sentences, and are separated from other sentences.

	+TCD	Dev. F1	Test P	Test R	Test F1
Span	no	69.9	62.8	74.9	68.3
	yes	75.1	78.1	69.2	73.3
Head	no	71.0	71.5	66.2	68.8
	yes	74.3	81.1	66.2	73.0

Table 5.8: Comparison of Span-based [66] and Head-based (ours) models on *RAMS*, given gold argument spans. "+TCD" indicates whether applying type-constrained decoding based on gold event types.

Here, the input to the Multi-layer Perceptron (MLP) is again the contextualized representations as depicted in §5.8.1. During training, we minimize cross-entropy losses on the left and right respectively. At test time, we expand to the maximumly-scored boundary words on both sides.

5.9 Experiment on IAP

We conduct all experiments[12] on the *RAMS* (v1.0) dataset and focus on the event argument detection task: given (gold) event triggers and their multi-sentence contexts, predicting the argument spans from raw input tokens. Following [66], we only use gold event types in the type-constrained decoding (TCD) setting.

Through our experiments, we adopt the pre-trained `bert-base-cased` model. We train all the models for maximumly 20 epochs. If fine-tuning BERT, we set the initial learning rate to 5e-5; otherwise, it is set to 2e-4. We jointly train our argument-detector and span-expander, with loss multipliers of 1.0 and 0.5, respectively.

Since head-words are not annotated, we apply a simple rule: utilizing predicted dependency trees, we heuristically pick the word that has the smallest arc distance to the dependency root as the head. Ties are broken by choosing the rightmost one. There are cases where this procedure does not always give the perfect head, or there is no single head-word for a span (e.g., in multi-word expressions or conjunction). Nevertheless, we find this strategy works well in practice.

5.9.1 Argument Linking with Gold Spans

Setting To compare our model with span-based models, we first evaluate in the same setting of [66] that assumes gold argument spans. We directly apply the head rule on the gold spans and consider the head-words as candidates. We also adopt the same BERT setting: learning a linear combination of layers 9, 10, 11 and 12, and applying neither the special input indicators nor fine-tuning.

[12]Our implementation is publicly available at `https://github.com/zzsfornlp/zmsp`

	+TCD	Dev.		Test	
		Span F1	Head F1	Span F1	Head F1
Seq.	no	$38.1_{\pm 0.7}$	$45.7_{\pm 0.7}$	$39.3_{\pm 0.4}$	$47.1_{\pm 0.7}$
	yes	$39.2_{\pm 0.7}$	$46.7_{\pm 0.8}$	$40.5_{\pm 0.4}$	$48.0_{\pm 0.5}$
Head	no	$38.9_{\pm 0.6}$	$46.4_{\pm 0.7}$	$40.1_{\pm 0.7}$	$47.7_{\pm 0.9}$
	yes	$40.3^{*}_{\pm 0.6}$	$48.0^{*}_{\pm 0.7}$	$41.8^{*}_{\pm 0.6}$	$49.7^{*}_{\pm 0.8}$

Table 5.9: Comparison of the sequence-labeling model (Seq.) and our Head-based model for argument detection on *RAMS* v1.0. All results are averaged over five runs, '$*$' denotes that the result of Head model is significantly better than the corresponding Seq. model (by paired randomization test, $p < 0.05$).

Results Table 5.8 compares our results with the reported results of the span-based model from [66]. The results show that the head-word approach can get comparable results to the span-based counterpart. This matches our hypothesis that head-words contain sufficient information of surrounding words using contextualized embedding, making them reasonable alternatives to full argument spans.

5.9.2 Full Argument Detection

Setting This setting considers all arguments from any spans in the multi-sentence context. Unless otherwise noted, here we use the last layer of BERT and apply fine-tuning for the whole model. We compare with a strong BERT-based BIO-styled sequence labeling model [186]. We adopt a modified version[13] from AllenNLP and re-train it on *RAMS* with similar settings: adopting special input indicators and fine-tuning BERT. For arguments that have multiple roles labels, we simply concatenate the labels as a new class.

Results Table 5.9 shows the main results for full argument detection. Since the criterion of full-span matching might be too strict in some way, we also report head-word based F1 scores by evaluating solely on head-word matches (obtained using the same head rules). The results show that our head-word based approach gets better results on average without type-constrained decoding and significantly better results after adopting type-constrained decoding with gold event types. Our head-driven approach is also flexible and easily extensible to more complex scenarios like nesting mentions or multiple roles, while keeping the linear complexity.

Ablation Table 5.10 lists the ablation results on the encoder. The results show that the BERT encoder contributes much to the performance of our full model. Fine-tuning BERT and the special indicator inputs can provide further improvements.

[13] `https://github.com/allenai/allennlp/blob/b89ff098372656b674ec71457dda071222fd05ae/allennlp/models/srl_bert.py`

	SpanF1	HeadF1
BERT-Full	$38.9_{\pm 0.6}$	$46.4_{\pm 0.7}$
No-Indicator	$35.6_{\pm 0.4}$	$42.9_{\pm 0.4}$
No-FineTuning	$34.4_{\pm 0.5}$	$40.0_{\pm 0.4}$
LSTM	$26.6_{\pm 0.4}$	$31.9_{\pm 0.6}$

Table 5.10: Ablation on the encoder for the head-based argument detection model (on development set, no type-constrained decoding). "BERT-Full" is our full fine-tuned BERT encoder, "No-Indicator" ablates indicating inputs, "No-FineTuning" freezes all pre-trained parameters of BERT, and "LSTM" replaces the BERT with a bi-directional LSTM encoder.

	d=-2 (3.6%)	d=-1 (7.5%)	d=0 (82.8%)	d=1 (4.0%)	d=2 (2.1%)
Seq.	$14.0_{\pm 0.6}$	$14.0_{\pm 2.4}$	$41.2_{\pm 0.9}$	$15.7_{\pm 1.0}$	$4.2_{\pm 2.5}$
Head	$15.6_{\pm 1.7}$	$15.3_{\pm 1.0}$	$43.4_{\pm 0.7}$	$17.8_{\pm 2.6}$	$8.5_{\pm 6.2}$

Table 5.11: Performance breakdown for Span-F1 by argument-trigger distance d (on development set, no type-constrained decoding). Numbers in parentheses at the second row indicate the distribution over distance d.

On Sentence Distances Table 5.11 lists the performance breakdown on different sentence distances between arguments and triggers. As opposed to the relative consistent performance in the gold span setting, as shown in [66], we notice a dramatic performance drop on non-local arguments. There may be two main reasons: 1) data imbalance, since non-local implicit arguments appear much less frequently (only around 18% in *RAMS*) than local ones; 2) lack of direct syntax signals, making the connections between the implicit arguments and event triggers much weaker than the local ones.

On Argument Roles We also investigate performance breakdowns on different argument roles. The results are shown in Figure 5.2, where we take the top-20 frequent roles to get more robust results. We can observe that our model performs better on core roles such as "*communicator*", "*employee*" and "*victim*" (with F1 $>$ 50), but struggles on non-core roles, like "*instrument*", "*origin*" and "*destination*", with F1 scores of around 20 to 30. The F1 scores correlate well (with Pearson and Spearman correlation coefficients of 0.64 and 0.70, respectively) with the local percentages: the more often one role appears locally around the event trigger, the better results it can obtain. These patterns are not surprising if we consider the possible underlying reasoning. The non-core arguments are not closely related with the event trigger, and thus can appear more freely at other places (or sometimes even be omitted), leading to a lower local percentage and also being harder to detect.

Category	Description	Example	Count (Percentage)
Correct	Correct	-	348 (38.6%)
Span	Unimportant span mismatch	The [monument]$_{\text{artifact}}$ to fallen Soviet sailors$_{\text{artifact}}$ in Limbazi, was **demolished**$_{\text{Destroy}}$ by activists.	82 (9.1%)
Coref.	Co-references	The United States$_{\text{destination}}$ gets more energy domestically, as [the country]$_{\text{destination}}$ continues to rely on oil **imports**$_{\text{Tranport}}$ from elsewhere.	60 (6.7%)
Possi.	Possible annotation problems	A Chinese official$_{\text{participant}}$ said **dialogue**$_{\text{Discussion}}$ was needed to resolve issues on the Korean peninsula.	44 (4.9%)
Partial	Partially correct	[His]$_{\text{recipient}}$ family, advisers and allies$_{\text{recipient}}$ set about **acquiring**$_{\text{Purchase}}$ expensive overseas homes and positions in the country.	26 (2.9%)
Frame	Frame errors	Relation was wrecked last November when [Turkey]$_{\text{killer attacker}}$ **shot**$_{\text{LifeDie}}$ down a fighter jet over the boarder.	31 (3.4%)
Others	Other errors	-	310 (34.4%)

Table 5.12: Examples and results of error analysis. In the examples, the **bold** text indicates the trigger word, followed by its event type noted in green. Arguments in gold annotations are indicated by the underlined spans with red role types, while the predicted arguments are indicated by [bracketed] spans with blue role types.

5.9.3 Manual Analysis

To further investigate in detail what type of errors the model makes, we sample 200 event frames from the development set and manually compare our model's predictions with the gold annotations. Overall, there are 459 annotated arguments and 442 predicted ones. For both annotated and predicted arguments, we assign them to one of seven categories, and the results are listed in Table 5.12. Here, the "Span" errors denote unimportant span mismatches, and they take nearly 9% of all items. If we ignore these errors, the performance can reach around 47%, which roughly matches the automatically evaluated Head-F1 scores. In some way, this supports our intuition to adopt a two-step approach, since the decisions of the span ranges may be separated from the core problem of argument detection, where head-words can be reasonable representatives. Another major source of errors comes from "Coref.", which is not surprising since the same entities can have multiple appearances at the document level. Our analysis indicates that

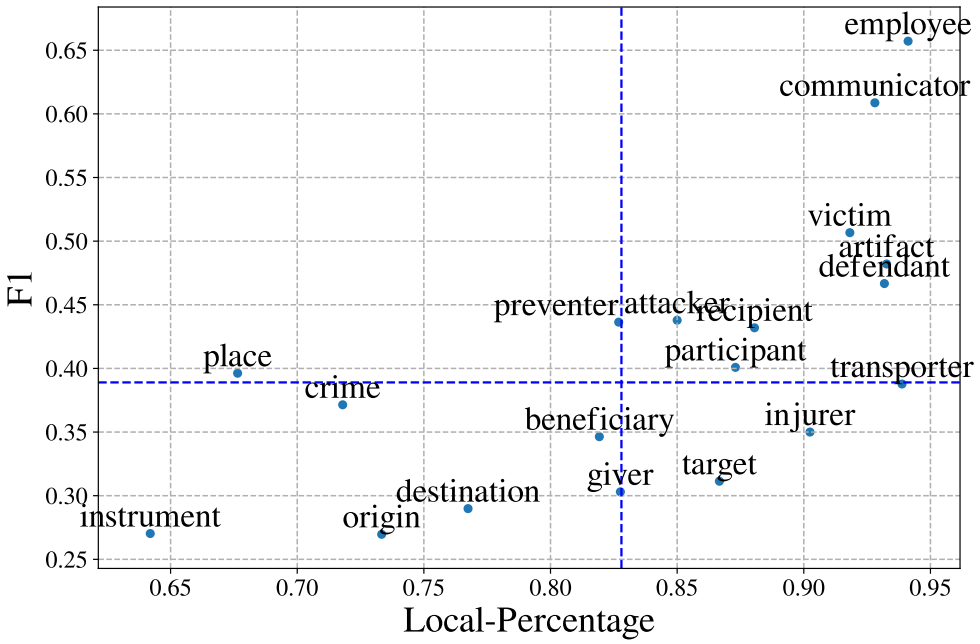

Figure 5.2: Performance breakdown of Span-F1 on the top-20 frequent roles (on development set, no type-constrained decoding). x-axis represents the percentage of local arguments for this role, while y-axis denotes the role specific Span-F1 scores. The two blue dashed lines denote the overall F1 scores (0.389) and local percentage (82.8%).

this is a problem that should be further investigated for both modeling and evaluation. Another notable type of error is frame mismatch ("Frame"). In the main setting (without type-constrained decoding), our model neither utilizes nor predicts event frame types, meaning that the frame information purely comes from the trigger words. Therefore, roles belonging to other event frames may be predicted. Finally, the "Others" category includes the ones where we cannot find obviously intuitive patterns. We would identify most of them as the more difficult cases, whose error breakdown follows similar patterns to the overall ones as shown in Figure 5.2.

Part II

Scaling up Data for Event Semantics

In Part II of this thesis, we explored supervised approaches to event structure prediction, relying on expert-annotated datasets to develop algorithms that decode various event structures. However, the limitations in the availability and scale of these datasets pose significant challenges for achieving comprehensive semantic understanding. To address these limitations, Part II of this thesis focuses on scaling up data to enhance the robustness and generalizability of event semantics models. This part investigates three distinct approaches to expand and utilize data effectively, uncovering new insights and phenomena in event semantics.

§7 proposes a crowdsourcing method for Cross-Document Event Coreference. In this chapter, we introduce a novel crowdsourcing workflow designed to handle the complexity of annotating cross-document event coreference. By breaking down the annotation tasks into simpler steps and incorporating follow-up questions to gather evidence on event mention, time, location, and participant overlap, we enable crowdworkers to contribute effectively. This approach not only scales the data but also leads to the discovery of a new type of partial identity, termed *spatiotemporal continuity*. The chapter details the methodology, implementation, and results of this crowdsourcing approach.

§6 utilizes indirect supervision for Event Salience Detection. Here, we explore the use of indirect supervision signals to create a large-scale event salience dataset. By leveraging summarization as a proxy task, we automatically generate event salience annotations using the Annotated New York Times corpus. This chapter discusses the process of dataset creation, the model development, and the observed semantic phenomena that emerge from this approach. We demonstrate how models trained with indirect supervision can capture script-related event mentions and correlated event arguments, enhancing our understanding of event salience.

§8 analyzes Language Models on Coreference and Winograd Schemas. This chapter delves into the capabilities of Large Language Models (LLMs) in solving coreference and related tasks, with a particular focus on the tasks similar to the Winograd Schema Challenge. We analyze the performance and developmental trajectory of LLMs using model checkpoints from the LLM360 project. Through circuit analysis methods, we investigate the underlying mechanisms that enable LLMs to address these complex tasks. This analysis reveals the strengths and limitations of current LLMs, highlighting areas where further advancements are needed to achieve human-level performance.

Chapter 6

Event Salience

6.1 Introduction

Automatic extraction of prominent information from text has always been a core problem in language research. While traditional methods mostly concentrate on the word level, researchers start to analyze higher-level discourse units in text, such as entities [63] and events [45].

Events are important discourse units that form the backbone of our communication. They play various roles in documents. Some are more central in discourse: connecting other entities and events, or providing key information of a story. Others are less relevant, but not easily identifiable by NLP systems. Hence it is important to be able to quantify the "importance" of events. For example, Figure 6.1 is a news excerpt describing a debate around a jurisdiction process: "*trial*" is central as the main discussing topic, while "*war*" is not.

Researchers are aware of the need to identify central events in applications like detecting salient relations [222], and identifying climax in storyline [207]. Generally, the salience of discourse units is important for language understanding tasks, such as document analysis [15], information retrieval [220], and semantic role labeling [41]. Thus, proper models for finding important events are desired.

In this chapter, we study the task of **event salience detection**, to find events that are most relevant to the main content of documents. To build a salience detection model, one core observation is that salient discourse units are forming discourse relations. In Figure 6.1, the "*trial*" event is connected to many other events: "*charge*" is pressed before "*trial*"; "*trial*" is being "*delayed*".

We present two salience detection systems based on the observations. First is a feature based learning to rank model. Beyond basic features like frequency and discourse location, we design features using cosine similarities among events and entities, to estimate the *content organization* [89]: how lexical meaning of elements relates to each other. Similarities from within-sentence or across the whole document are used to capture interactions on both local and global aspects (§6.4). The model significantly outperforms a strong "Frequency" baseline in our experiments.

However, there are other discourse relations beyond lexical similarity. Figure 6.1 showcases

Federal prosecutors **urged** a **trial** **judge** today to **deny** defense **requests** to **delay** the **trial** of Zacarias Moussaoui and suggested that Mr. Moussaoui, the only person **charged** in the Sept. 11 **attacks**, was to **blame** for many of the **delays** so far. The **attacks** ''were volleys in a declared war against the United States and were more than just acts of terror,'' the prosecutors said in a filing to the Federal District Court in Alexandria, Va. ''Thus, the victims' and the nation's interest in a fair and speedy **trial** is beyond dispute.''.Last week, court-appointed defense lawyers asked that the starting date of the **trial**, now set for Sept. 30, be **delayed** by at least two months to allow them to wade through volumes of evidence that prosecutors have presented to them, including more than 1,300 computer discs.

Figure 6.1: Examples annotations. Underlying words are annotated event triggers; the red bold ones are annotated as salient.

some: the script relation [185][1] between "*charge*" and "*trial*", and the frame relation [12] between "*attacks*" and "*trial*" ("*attacks*" fills the "charges" role of "*trial*"). Since it is unclear which ones contribute more to salience, we design a Kernel based Centrality Estimation (KCE) model (§6.5) to capture salient specific interactions between discourse units automatically.

In KCE, discourse units are projected to embeddings, which are trained end-to-end towards the salience task to capture rich semantic information. A set of soft-count kernels are trained to weigh salient specific latent relations between discourse units. With the capacity to model richer relations, KCE outperforms the feature-based model by a large margin (§6.7.1). Our analysis shows that KCE is exploiting several relations between discourse units: including script and frames (Table 6.5). To further understand the nature of KCE, we conduct an *intrusion test* (§6.6.2), which requires a model to identify events from another document. The test shows salient events form tightly related groups with relations captured by KCE.

The notion of salience is subjective and may vary from person to person. We follow the empirical approaches used in entity salience research [63]. We consider the *summarization test*: an event is considered salient if a summary written by a human is likely to include it, since events about the main content are more likely to appear in a summary. This approach allows us to create a large-scale corpus (§6.3).

This chapter makes three main contributions. First, we present two event salience detection systems, which capture rich relations among discourse units. Second, we observe interesting connections between salience and various discourse relations (§6.7.1 and Table 6.5), implying potential research on these areas. Finally, we construct a large scale event salience corpus,

[1]Scripts are prototypical sequences of events: a *restaurant* script normally contains events like "order", "eat" and "pay".

providing a testbed for future research. Our code, dataset and models are publicly available[2].

6.2 Related Work

Events have been studied on many aspects due to their importance in language. To name a few: event detection [118, 151, 164], coreference [121, 128], temporal analysis [30, 59], sequencing [5], script induction [13, 31, 165, 178].

However, studies on event salience are premature. Some previous work attempts to approximate event salience with word frequency or discourse position [207, 222]. Parallel to ours, Choubey et al. [45] propose a task to find the most dominant event in news articles. They draw connections between event coreference and importance, on hundreds of closed-domain documents, using several oracle event attributes. In contrast, our proposed models are fully learned and applied on more general domains and at a larger scale. We also do not restrict to a single most important event per document.

There is a small but growing line of work on entity salience [60, 63, 167, 220]. In this work, we study the case for events.

Text relations have been studied in tasks like text summarization, which mainly focused on cohesion [92]. Grammatical cohesion methods make use of document level structures such as anaphora relations [14] and discourse parse trees [133]. Lexical cohesion based methods focus on repetitions and synonyms on the lexical level [71, 146, 188]. Though sharing similar intuitions, our proposed models are designed to learn richer semantic relations in the embedding space.

Comparing to the traditional summarization task, we focus on events, which are at a different granularity. Our experiments also unveil interesting phenomena among events and other discourse units.

6.3 The Event Salience Corpus

This section introduces our approach to construct a large-scale event salience corpus, including methods for finding event mentions and obtaining saliency labels. The studies are based on the Annotated New York Times corpus [183], a newswire corpus with expert-written abstracts.

6.3.1 Automatic Corpus Creation

Event Mention Annotation: Despite many annotation attempts on events [25, 170], automatic labeling of them in general domain remains an open problem. Most of the previous work follows empirical approaches. For example, Chambers and Jurafsky [31] consider all verbs together with their subject and object as events. Do et al. [58] additionally include nominal predicates, using the nominal form of verbs and lexical items under the *Event* frame in FrameNet [12].

There are two main challenges in labeling event mentions. First, we need to decide which lexical items are event triggers. Second, we have to disambiguate the word sense to correctly identify events. For example, the word "phone" can refer to an entity (a physical phone) or an

[2] `https://github.com/hunterhector/EventSalience`

	Train	Dev	Test
# Documents	526126	64000	63589
Avg. # Word	794.12	790.27	798.68
Avg. # Events	61.96	60.65	61.34
Avg. # Salience	8.77	8.79	8.90

Table 6.1: Dataset Statistics.

event (a phone call event). We use FrameNet to solve these problems. We first use a FrameNet based parser: Semafor [56], to find and disambiguate triggers into frame classes. We then use the FrameNet ontology to select event mentions.

Our frame based selection method follows the Vendler classes [204], a four way classification of eventuality: *states*, *activities*, *accomplishments* and *achievements*. The last three classes involve state change, and are normally considered as events. Following this, we create an "event-evoking frame" list using the following procedure:

1. We keep frames that are subframes of *Event* and *Process* in the FrameNet ontology.
2. We discard frames that are subframes of state, entity and attribute frames, such as *Entity*, *Attributes*, *Locale*, etc.
3. We manually inspect frames that are not subframes of the above-mentioned ones (around 200) to keep event related ones (including subframes), such as *Arson*, *Delivery*, etc.

This gives us a total of 569 frames. We parse the documents with Semafor and consider predicates that trigger a frame in the list as candidates. We finish the process by removing the light verbs[3] and reporting events[4] from the candidates, similar to previous research [176].

Salience Labeling: For all articles with a human written abstract (around 664,911) in the New York Times Annotated Corpus, we extract event mentions. We then label an event mention as salient if we can find its lemma in the corresponding abstract (Mitamura et al. [143] showed that lemma matching is a strong baseline for event coreference.). For example, in Figure 6.1, event mentions in bold and red are found in the abstract, thus labeled as salient. Data split is detailed in Table 6.1 and §6.6.

6.3.2 Annotation Quality

While the automatic method enables us to create a dataset at scale, it is important to understand the quality of the dataset. For this purpose, we have conducted two small manual evaluation study.

Our lemma-based salience annotation method is based on the assumption that lemma matching being a strong detector for event coreference. In order to validate this assumption, one of the

[3]Light verbs carry little semantic information: "appear", "be", "become", "do", "have", "seem", "do", "get", "give", "go", "have", "keep", "make", "put", "set", "take".

[4]Reporting verbs are normally associated with the narrator: "argue", "claim", "say", "suggest", "tell".

Name	Description
`Frequency`	The frequency of the event lemma in document.
`Sentence Location`	The location of the first sentence that contains the event.
`Event Voting`	Average cosine similarity with other events in document.
`Entity Voting`	Average cosine similarity with other entities in document.
`Local Entity Voting`	Average cosine similarity with entities in the sentence.

Table 6.2: Event Salience Features.

authors manually examined 10 documents and identified 82 coreferential event mentions pairs between the text body and the abstract. The automatic lemma rule identifies 72 such pairs: 64 of these matches human decision, producing a precision of 88.9% $(64/72)$ and a recall of 78% $(64/82)$. There are 18 coreferential pairs missed by the rule.

The next question is: *is an event really important if it is mentioned in the abstract?* Although prior work [63] shows that the assumption to be valid for entities, we study the case for events. We asked two annotators to manually annotate 10 documents (around 300 events) using a 5-point Likert scale for salience. We compute the agreement score using Cohen's Kappa [46]. We find the task to be challenging for human: annotators don't agree well on the 5-point scale (Cohen's Kappa = 0.29). However, if we collapse the scale to binary decisions, the Kappa between the annotators raises to 0.67. Further, the Kappa between each annotator and automatic labels are 0.49 and 0.42 respectively. These agreement scores are also close to those reported in the entity salience tasks [63].

While errors exist in the automatic annotation process inevitably, we find the error rate to be reasonable for a large-scale dataset. Further, our study indicates the difficulties for human to rate on a finer scale of salience. We leave the investigation of continuous salience scores to future work.

6.4 Feature-Based Event Salience Model

This section presents the feature-based model, including the features and the learning process.

6.4.1 Features

Our features are summarized in Table 6.2.

Basic Discourse Features: We first use two basic features similar to Dunietz and Gillick [63]: *Frequency* and *Sentence Location*. *Frequency* is the lemma count of the mention's syntactic head word [132]. *Sentence Location* is the sentence index of the mention, since the first few sentences are normally more important. These two features are often used to estimate salience [15, 207].

Content Features: We then design several lexical similarity features, to reflect Grimes' content relatedness [89]. In addition to events, the relations between events and entities are also important. For example, Figure 6.1 shows some related entities in the legal domain, such as "*prosecutors*" and "*court*". Ideally, they should help promote the salience status for event "*trial*".

Lexical relations can be found both within-sentence (local) or across sentence (global) [92]. We compute the local part by averaging similarity scores from other units in the same sentence. The global part is computed by averaging similarity scores from other units in the document. All similarity scores are computed using cosine similarities on pre-trained embeddings [141].

These lead to 3 content features: *Event Voting*, the average similarity to other events in the document; *Entity Voting*, the average similarity to entities in the document; *Local Entity Voting*, the average similarity to entities in the same sentence. Local event voting is not used since a sentence often contains only 1 event.

6.4.2 Model

A Learning to Rank (`LeToR`) model [120] is used to combine the features. Let ev_i denote the ith event in a document d. Its salience score is computed as:

$$f(ev_i, d) = W_f \cdot F(ev_i, d) + b \tag{6.1}$$

where $F(ev_i, d)$ is the features for ev_i in d (Table 6.2); W_f and b are the parameters to learn.

The model is trained with pairwise loss:

$$\sum_{ev^+, ev^- \in d} \max(0, 1 - f(ev^+, d) + f(ev^-, d)), \tag{6.2}$$
$$\text{w.r.t. } y(ev^+, d) = +1 \ \& \ y(ev^-, d) = -1.$$

$$y(e_i, d) = \begin{cases} +1, & \text{if } e_i \text{ is a salient entity in } d, \\ -1, & \text{otherwise.} \end{cases}$$

where ev^+ and ev^- represent the salient and non-salient events; y is the gold standard function. Learning can be done by standard gradient methods.

6.5 Neural Event Salience Model

As discussed in §6.1, the salience of discourse units is reflected by rich relations beyond lexical similarities, for example, script ("*charge*" and "*trial*") and frame (a "*trial*" of "*attacks*"). The relations between these words are specific to the salience task, thus difficult to be captured by raw cosine scores that are optimized for word similarities. In this section, we present a neural model to exploit the embedding space more effectively, in order to capture relations for event salience estimation.

6.5.1 Kernel-based Centrality Estimation

Inspired by the kernel ranking model [219], we propose Kernel-based Centrality Estimation (KCE), to find and weight semantic relations of interests, in order to better estimate salience.

Formally, given a document d, the set of annotated events $\mathbb{V} = \{ev_1, \dots ev_i \dots, ev_n\}$, KCE first embed an event into vector space: $ev_i \xrightarrow{Emb} \overrightarrow{ev_i}$. The embedding function is initialized with pre-trained embeddings. It then extract K features for each ev_i:

$$\Phi_K(ev_i, \mathbb{V}) = \{\phi_1(\overrightarrow{ev_i}, \mathbb{V}), \dots, \phi_k(\overrightarrow{ev_i}, \mathbb{V}), \dots, \phi_K(\overrightarrow{ev_i}, \mathbb{V})\}, \tag{6.3}$$

$$\phi_k(\overrightarrow{ev_i}, \mathbb{V}) = \sum_{ev_j \in \mathbb{V}} \exp\left(-\frac{(\cos(\overrightarrow{ev_i}, \overrightarrow{ev_j}) - \mu_k)^2}{2\sigma_k^2}\right). \tag{6.4}$$

$\phi_k(\overrightarrow{ev_i}, \mathbb{V})$ is the k-th Gaussian kernel with mean μ_k and variance σ_k^2. It models the interactions between events in its kernel range defined by μ_k and σ_k. $\Phi_K(ev_i, \mathbb{V})$ enforces multi-level interactions among events — relations that contribute similarly to salience are expected to be grouped into the same kernels. Such interactions greatly improve the capacity of the model with negligible increase in the number of parameters. Empirical evidences [219] have shown that kernels in this form are effective to learn weights for task-specific term pairs.

The final salience score is computed as:

$$f(ev_i, d) = W_v \cdot \Phi_K(ev_i, \mathbb{V}) + b, \tag{6.5}$$

where W_v is learned to weight the contribution of the certain relations captured by each kernel.

We then use the exact same learning objective as in equation (6.2). The pairwise loss is first back-propagated through the network to update the kernel weights W_v, assigning higher weights to relevant regions. Then the kernels use the gradients to update the embeddings, in order to capture the meaningful discourse relations for salience.

Since the features and KCE capture different aspects, combining them may give superior performance. This can be done by combining the two vectors in the final linear layer:

$$f(ev_i, d) = W_v \cdot \Phi_K(ev_i, \mathbb{V}) + W_f \cdot F(ev_i, d) + b \tag{6.6}$$

6.5.2 Integrating Entities into KCE

KCE is also used to model the relations between events and entities. For example, in Figure 6.1, the entity "*court*" is a frame element of the event "*trial*"; "*United States*" is a frame element of the event "*war*". It is not clear which pair contributes more to salience. We again let KCE to learn it.

Formally, let $\mathbb{E}$ be the list of entities in the document, i.e. $\mathbb{E} = \{en_1, \dots, en_i, \dots, en_n\}$, where en_i is the ith entity in document d. KCE extracts the kernel features about entity-event relations as follows:

$$\Phi_K(ev_i, \mathbb{E}) = \{\phi_1(\overrightarrow{ev_i}, \mathbb{E}), \ldots, \phi_k(\overrightarrow{ev_i}, \mathbb{E}), \ldots, \phi_K(\overrightarrow{ev_i}, \mathbb{E})\}, \tag{6.7}$$

$$\phi_k(\overrightarrow{ev_i}, \mathbb{E}) = \sum_{en_j \in \mathbb{E}} \exp\left(-\frac{(\cos(\overrightarrow{ev_i}, \overrightarrow{en_j}) - \mu_k)^2}{2\sigma_k^2}\right) \tag{6.8}$$

similarly, en_i is embedded by: $en_i \xrightarrow{Emb} \overrightarrow{en_i}$, which is initialized by pre-trained entity embeddings.

We reach the full KCE model by combining all the vectors using a linear layer:

$$f(ev_i, d) = W_e \cdot \Phi_K(ev_i, \mathbb{E}) + W_v \cdot \Phi_K(ev_i, \mathbb{V}) + W_f \cdot F(ev_i, d) + b \tag{6.9}$$

The model is again trained by equation (6.2).

6.6 Experimental Methodology

This section describes our experiment settings.

6.6.1 Event Salience Detection

Dataset: We conduct our experiments on the salience corpus described in §6.3. Among the 664,911 articles with abstracts, we sample 10% of the data as the test set and then randomly leave out another 10% documents for development. Overall, there are 4359 distinct event lexical items, at a similar scale with previous work [31, 58]. The corpus statistics are summarized in Table 6.1.
Input: The inputs to models are the documents and the extracted events. The models are required to rank the events from the most to least salience.
Baselines: Three methods from previous researches are used as baselines: *Frequency*, *Location* and *PageRank*. The first two are often used to simulate saliency [15, 207]. The *Frequency* baseline ranks events based on the count of the headword lemma; the *Location* baseline ranks events using the order of their appearances in discourse. Ties are broken randomly.

Similar to entity salience ranking with PageRank scores [220], our *PageRank* baseline runs PageRank on a fully connected graph whose nodes are the events in documents. The edges are weighted by the embedding similarities between event pairs. We conduct supervised PageRank on this graph, using the same pairwise loss setup as in KCE. We report the best performance obtained by linearly combining *Frequency* with the scores obtained after a one-step random walk.
Evaluation Metric: Since the importance of events is on a continuous scale, the boundary between "important" and "not important" is vague. Hence we evaluate it as a ranking problem. The metrics are the precision and recall value at 1, 5 and 10 respectively. It is adequate to stop at 10 since there are less than 9 salient events per document on average (Table 6.1). We also report Area Under Curve (AUC). Statistical significance values are tested by permutation (randomization) test with $p < 0.05$.

Implementation Details: We pre-trained word embeddings with 128 dimensions on the whole Annotated New York Times corpus using Word2Vec [141]. Entities are extracted using the TagMe entity linking toolkit [75]. Words or entities that appear only once in training are replaced with special "unknown" tokens.

The hyper-parameters of the KCE kernels follow previous literature [219]. There is one exact match kernel ($\mu = 1, \sigma = 1e^{-3}$) and ten soft-match kernels evenly distributed between $(-1, 1)$, i.e. $\mu \in \{-0.9, -0.7, \ldots, 0.9\}$, with the same $\sigma = 0.1$.

The parameters of the models are optimized by Adam [108], with batch size 128. The vectors of entities are initialized by the pre-trained embeddings. Event embeddings are initialized by their headword embedding.

6.6.2 The Event Intrusion Test: A Study

KCE is designed to estimate salience by modeling relations between discourse units. To better understand its behavior, we design the following **event intrusion test**, following the word intrusion test used to assess topic model quality [33].

Event Intrusion Test: The test will present to a model a set of events, including: the **origins**, all events from one document; the **intruders**, some events from another document. Intuitively, if events inside a document are organized around the core content, a model capturing their relations well should easily identify the intruder(s).

Specifically, we take a bag of unordered events $\{O_1, O_2, \ldots, O_p\}$, from a document O, as the origins. We insert into it intruders, events drawn from another document, I: $\{I_1, I_2, \ldots, I_q\}$. We ask a model to rank the mixed event set $M = \{O_1, I_1, O_2, I_2, \ldots\}$. We expect a model to rank the intruders I_i below the origins O_i.

Intrusion Instances: From the development set, we randomly sample 15,000 origin and intruding document pairs. To simplify the analysis, we only take documents with at least 5 salient events. The intruder events, together with the entities in the same sentences, are added to the origin document.

Metrics: AUC is used to quantify ranking quality, where events in O are positive and events in I are negative. To observe the ranking among the salient origins, we compute a separate AUC score between the intruders and the salient origins, denoted as SA-AUC. In other words, SA-AUC is the AUC score on the list with non-salient origins removed.

Experiments Details: We take the full KCE model to compute salient scores for events in the mixed event set M, which are directly used for ranking. Frequency is recounted. All other features (Table 6.2) are set to 0 to emphasize the relational aspects,

We experiment with two settings: 1. adding only the salient intruders. 2. adding only the non-salient intruders. Under both settings, the intruders are added one by one, allowing us to observe the score change regarding the number of intruders added. For comparison, we add a *Frequency* baseline, that directly ranks events by the Frequency feature.

6.7 Evaluation Results

This section presents the evaluations and analyses.

Method	P@01		P@05		P@10		AUC	
Location	0.3555	–	0.3077	–	0.2505	–	0.5226	–
PageRank	0.3628	–	0.3438	–	0.3007	–	0.5866	–
Frequency	0.4542	–	0.4024	–	0.3445	–	0.5732	–
LeToR	$0.4753^{\dagger}$	+4.64%	$0.4099^{\dagger}$	+1.87%	$0.3517^{\dagger}$	+2.10%	$0.6373^{\dagger}$	+11.19%
KCE (-EF)	0.4420	−2.69%	0.4038	+0.34%	$0.3464^{\dagger}$	+0.54%	$0.6089^{\dagger}$	+6.23%
KCE (-E)	$0.4861^{\dagger\ddagger}$	+7.01%	$0.4227^{\dagger\ddagger}$	+5.04%	$0.3603^{\dagger\ddagger}$	+4.58%	$0.6541^{\dagger\ddagger}$	+14.12%
KCE	$0.5049^{\dagger\ddagger}$	+11.14%	$0.4277^{\dagger\ddagger}$	+6.29%	$0.3638^{\dagger\ddagger}$	+5.61%	$0.6557^{\dagger\ddagger}$	+14.41%

Method	R@01		R@05		R@10		W/T/L
Location	0.0807	–	0.2671	–	0.3792	–	–/–/–
PageRank	0.0758	–	0.2760	–	0.4163	–	–/–/–
Frequency	0.0792	–	0.2846	–	0.4270	–	–/–/–
LeToR	$0.0836^{\dagger}$	+5.61%	$0.2980^{\dagger}$	+4.70%	$0.4454^{\dagger}$	+4.31%	8037 / 48493 / 6770
KCE (-EF)	0.0714	−9.77%	0.2812	−1.18%	$0.4321^{\dagger}$	+1.20%	6936 / 48811 / 7553
KCE (-E)	$0.0925^{\dagger\ddagger}$	+16.78%	$0.3172^{\dagger\ddagger}$	+11.46%	$0.4672^{\dagger\ddagger}$	+9.41%	11676 / 43294 / 8330
KCE	$0.0946^{\dagger\ddagger}$	+19.44%	$0.3215^{\dagger\ddagger}$	+12.96%	$0.4719^{\dagger\ddagger}$	+10.51%	12554 / 41461 / 9285

Table 6.3: Event salience performance. (-E) and (-F) marks removing Features and Entity information from the full KCM model. The relative performance differences are computed against Frequency. W/T/L are the number of documents a method wins, ties, and loses compared to Frequency. † and ‡ mark the statistically significant improvements over $\texttt{Frequency}^{\dagger}$, $\texttt{LeToR}^{\ddagger}$ respectively.

6.7.1 Event Salience Performance

We summarize the main results in Table 6.3.

Baselines: *Frequency* is the best performing baseline. Its precision at 1 and 5 are higher than 40%. *PageRank* performs worse than *Frequency* on all the precision and recall metrics. *Location* performs the worst.

Feature Based: LeToR outperforms the baselines significantly on all metrics. Particularly, its P@1 value outperforms the *Frequency* baseline the most (4.64%), indicating a much better estimation on the most salient event. In terms of AUC, LeToR outperforms *Frequency* by a large margin (11.19% relative gain).

Feature Ablation: To understand the contribution of individual features, we conduct an ablation study of various feature settings in Table 4.5. We gradually add feature groups to the *Frequency* baseline. The combination of *Location* (sentence location) and *Frequency* almost sets the performance for the whole model. Adding each voting feature individually produces mixed results. However, adding all voting features improves all metrics. Though the margin is small, 4 of them are statistically significant over *Frequency+Location*.

Kernel Centrality Estimation: The KCE model further beats LeToR significantly on all metrics, by around 5% on AUC and precision values, and by around 10% on the recall values. Notably, the P@1 score is much higher, reaching 50%. The large relative gain on all the recall metrics and

Feature Groups	P@1	P@5	P@10	R@1	R@5	R@10	AUC
SL	0.3548	0.3069	0.2497	0.0807	0.2671	0.3792	0.5226
Frequency	0.4536	0.4018	0.3440	0.0792	0.2846	0.4270	0.5732
+ SL	0.4734	0.4097	0.3513	0.0835	0.2976	0.4436	0.6354
+ SL + Event	0.4726	0.4101†	0.3516	0.0831	0.2969	0.4431	0.6365†
+ SL + Entity	0.4739	0.4100	0.3518	0.0812	0.2955	0.4418	0.6374
+ SL + Entity + Event	0.4739	0.4100	0.3518†	0.0832	0.2974	0.4452†	0.6374†
+ SL + Entity + Event + Local	0.4754†	0.4100	0.3517†	0.0837	0.2981	0.4454†	0.6373†

Table 6.4: Event Salience Feature Ablation Results. The + sign indicates adding feature groups to Frequency. SL is the sentence location feature. Event is the event voting feature. Entity is the entity voting feature. Local is the local entity voting feature. † marks the statistically significant improvements over +SL.

the high performance on precision show that KCE works really well on the top of the rank list.

Kernel Ablation: To understand the source of performance gain of KCE, we conduct an ablation study by removing its components: −E removes of entity kernels; −EF removes the entity kernels and the features. We observe a performance drop in both cases. Without entities and features, the model only using event information still performs similarly to *Frequency*. The drops are also a reflection of the small number of events ($\approx$ 60 per document) comparing to entities ($\approx$ 200 per document). The study indicates that the relational signals and features contain different but both important information.

Discussion: The superior results of KCE demonstrate its effectiveness in predicting salience. So what additional information does it capture? We revisit the changes made by KCE: 1. it adjusts the embeddings during training. 2. it introduces weighted soft count kernels. However, the *PageRank* baseline also does embedding tuning but produces poor results, thus the second change should be crucial. We plot the learned kernel weights of KCE in Figure 6.2. Surprisingly, the salient decisions are not linearly related, nor even positively correlated to the weights. In fact, besides the "Exact Match" bin, the highest absolute weights actually appear at 0.3 and -0.3. This implies that embedding similarities do not directly imply salience, breaking some assumptions of the feature based model and *PageRank*.

Case Study: We inspect some pairs of events and entities in different kernels and list some examples in Table 6.5. The pre-trained embeddings are changed a lot. Pairs of units with different raw similarity values are now placed in the same bin. The pairs in Table 3 exhibit interesting types of relations: e.g.,"*arrest-charge*" and "*attack-kill*" form script-like chains; "911 attack" forms a quasi-identity relation [174] with "attack"; "business" and "increase" are candidates as frame-argument structure. While these pairs have different raw cosine similarities, they are all useful in predicting salience. KCE learns to gather these relations into bins assigned with higher weights, which is not achieved by pure embedding based methods. The KCE has changed the embedding space and the scoring functions significantly from the original space after training.

		Word2Vec	Kernel
attack	kill	0.69	0.3
arrest	charge	0.53	0.3
USA (**E**)	war	0.46	0.3
911 attack (**E**)	attack	0.72	0.3
attack	trade	0.42	0.9
hotel (**E**)	travel	0.49	0.9
charge	murder	0.49	0.7
business(**E**)	increase	0.43	0.7
attack	walk	0.44	-0.3
people (**E**)	work	0.40	-0.3

Table 6.5: Examples of pairs of Events/Entities in the kernels. The **Word2vec** column shows the cosine similarity using pre-trained word vectors. The **Kernel** column shows the closest kernel they belong after training. Items marked with (**E**) are entities.

This partially explains why the raw voting features and PageRank are not as effective.

6.7.2 Intrusion Test Results

Figure 6.3 plots results of the intrusion test . The left figure shows the results of setting 1: adding non-salient intruders. The right one shows the results of setting 2: adding salient intruders. The AUC is 0.493 and the SA-AUC is 0.753 if all intruders are added.

The left figure shows that KCE successfully finds the non-salient intruders. The SA-AUC is higher than 0.8. Yet the AUC scores, which include the rankings of non-salience events, are rather close to random. This shows that the salient events in the origin documents form a more cohesive group, making them more robust against the intruders; the non-salient ones are not as cohesive.

In both settings, KCE produces higher SA-AUC than *Frequency* at the first 30%. However, in setting 2, KCE starts to produce lower SA-AUC than *Frequency* after 30%, then gradually drops to 0.5 (random). This phenomenon is expected since the asymmetry between origins and intruders allow KCE to distinguish them at the beginning. When all intruders are added, KCE performs worse because it relies heavily on the relations, which can be also formed by the salient intruders. This phenomenon is observed only on the salient intruders, which again confirms the cohesive relations are found among salient events.

In conclusion, we observe that the salient events form tight groups connected by discourse relations while the non-salient events are not as related. The observations imply that the main scripts in documents are mostly anchored by small groups of salient events (such as the "Trial" script in Example 6.1). Other events may serve as "backgrounds" [42]. Similarly, Choubey et al. [45] find that relations like event coreference and sequence are important for saliency.

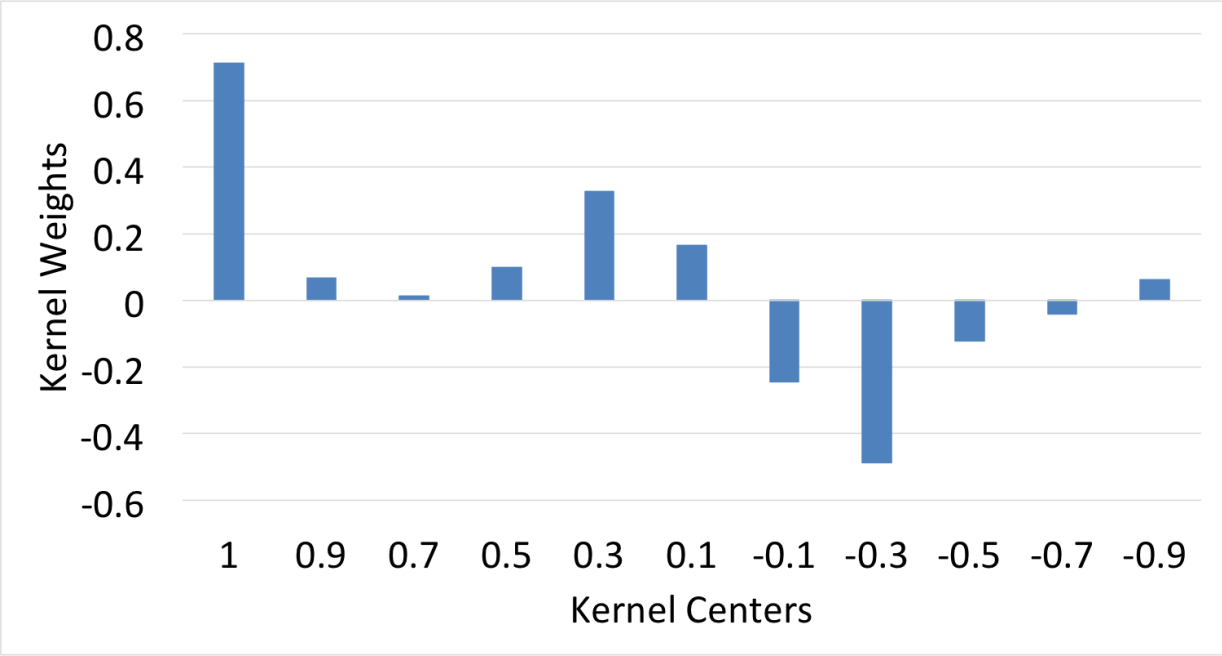

Figure 6.2: Learned Kernel Weights of KCE

6.8 Conclusion

In this chapter we describe two salient detection models, based on lexical relatedness and semantic relations. The feature-based model with lexical similarities is effective, but cannot capture semantic relations like scripts and frames. The KCE model uses kernels and embeddings to capture these relations, thus outperforms the baselines and feature-based models significantly. All the results are tested on our newly created large-scale event salience dataset. While the automatic method inevitably introduces noises to the dataset, the scale enables us to study complex event interactions, which is infeasible via costly expert labeling.

Our case study shows that the salience model finds and utilize a variety of discourse relations: script chain (*attack* and *kill*), frame argument relation (*business* and *increase*), quasi-identity (*911 attack* and *attack*). Such complex relations are not as prominent in the raw word embedding space. The core message is that a salience detection module automatically discovers connections between salience and relations. This goes beyond prior centering analysis work that focuses on lexical and syntax and provide a new semantic view from the script and frame perspective.

In the intrusion test, we observe that the small number of salient events are forming tight connected groups. While KCE captures these relations quite effectively, it can be confused by salient intrusion events. The phenomenon indicates that the salient events are tightly connected, which form the main scripts of documents.

In this chapter, we have shown that we can use indirect supervision signals to reveal many interesting semantic relations between discourse phenomena and salience. Some of them are not directly related to our proposed task. For example, our study suggests that core script information

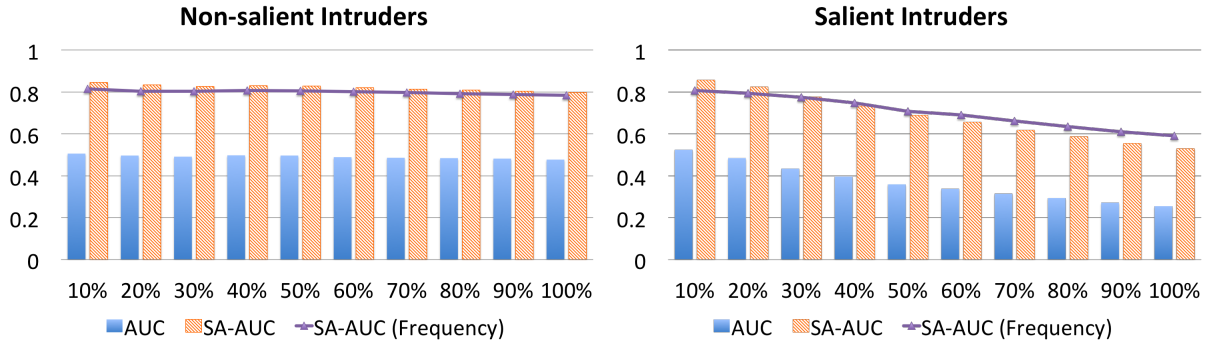

Figure 6.3: Intruder study results. X-axis shows the percentage of intruders inserted. Y-axis is the AUC score scale. The left and right figures are results from salient and non-salient intruders respectively. The blue bar is AUC. The orange shaded bar is SA-AUC. The line shows the SA-AUC of the frequency baseline.

may reside mostly in the salient events. In this proposal, we plan to use similar methods to reveal other semantic knowledge from data, with a focus on frames and scripts.

Chapter 7

Cross-document Event Identity via Dense Annotation

7.1 Introduction

Coreference resolution is the task of identifying events (or entities) that refer to the same underlying activity (or objects). Accurately resolving coreference is a prerequisite for many NLP tasks, such as question answering, summarization, and dialogue understanding. For instance, to get a holistic view of an ongoing natural disaster, we need to aggregate information from various sources (newswire, social media, public communication, etc.) over an extended period. Often this requires resolving coreference between mentions across documents.[1]

Recasens et al. [175] defines coreference as "identity of reference". Therefore, modeling event coreference requires understanding the extent of the shared identity between event mentions. Numerous factors determine this identity, including the semantics of the event mention, its arguments, and the document context. Resolving coreference across documents is more challenging, as it requires modeling identity over a much longer context. To this end, we identify two major issues with existing cross-document event coreference (CDEC) datasets that limit the progress on this task. First, many prior datasets often annotate coreference *only on a restricted set of event types*, limiting the coverage of mentions in the dataset. Second, many datasets and models *insufficiently tackle the concept of event identity*. As highlighted by Hovy et al. [102], the decision of whether two mentions refer to the same event is often non-trivial. Occasionally, event mentions only share a partial identity (*quasi-identity*). In this work, we present a new dataset for CDEC that attempts to overcome both issues.

Earlier efforts on CDEC dataset collection were limited to specific pre-defined event types, restricting the scope of event mentions that could be studied. In this work, we instead annotate mentions of all types, i.e., open-domain events [6], and provide a *dense annotation* [28] by checking for coreference relationship between every mention pair in all underlying document pairs. We compile documents from the publicly available English Wikinews.[2] To facilitate our goal of dense annotation of mentions and their coreference, we develop and release a new

[1] A mention is a linguistic expression in text that denotes a specific instance of an event.

[2] `https://en.wikinews.org/`

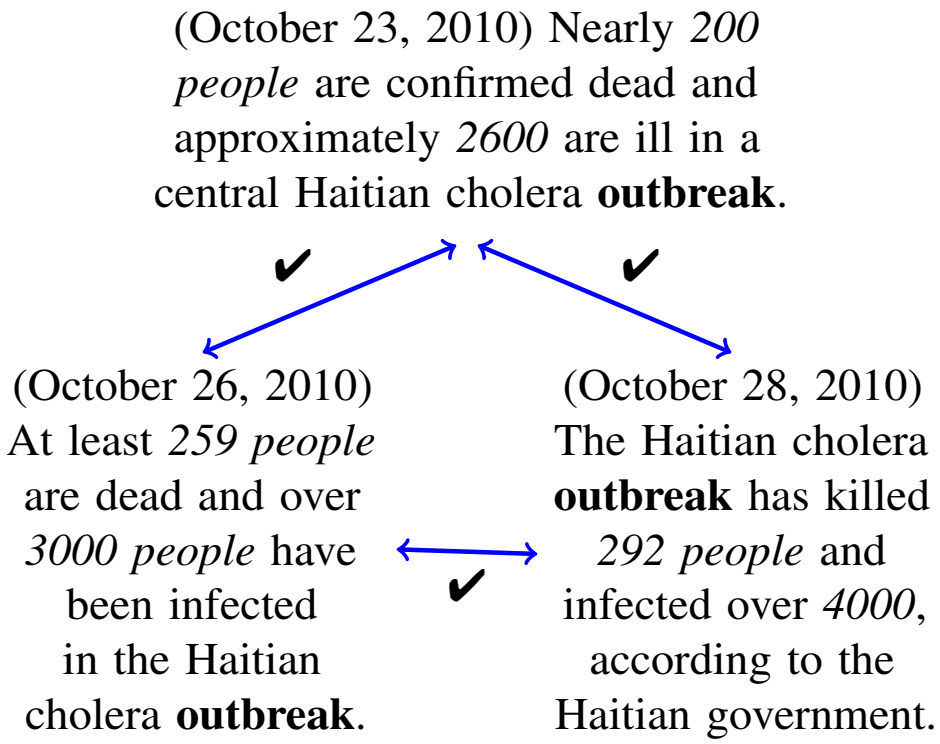

Figure 7.1: An illustration of the quasi-identity nature of events. The event [Haitian cholera] 'outbreak' is expressed by instances with varying counts of infections and deaths. The identity of this event continuously evolves over space and time, attributed to a new type of quasi-identity, spatiotemporal continuity.

easy-to-use annotation tool that allows linking text spans across documents. We crowdsource coreference annotations on Mechanical Turk.[3]

Prior work has attributed the quasi-identity behavior of events to two specific phenomena, membership and subevent [102]. However, its implications in cross-document settings remain unclear. In this work, we specifically focus on a cross-document setup. As highlighted by Recasens et al. [177], a direct annotation of quasi-identity relations is hard because annotators might not be familiar with the phenomenon. Therefore, we propose a new annotation workflow that allows for easy determination of quasi-identity links. To this end, we collect evidence for time, location, and participant(s) overlap between corefering mentions. We also collect information regarding any potential inclusion relationship between the mention pair.

Our workflow allowed us to empirically identify a new type of quasi-identity, *spatiotemporal continuity*, in addition to the existing types defined by Hovy et al. [102]. Figure 7.1 illustrates this phenomenon using the case of [Haitian cholera] outbreak. The event gradually evolves over space and time, leading to cases of partial coreference. Additionally, traditional coreference annotations

[3] `https://www.mturk.com/`

cluster mentions together. However, this methodology can be misleading when dealing with cases of quasi-identity (see §7.5). To overcome this limitation, we frame our annotation task as a (cross-document) mention pair linking. The proposed task simplifies the annotation process by avoiding merging quasi-identical mentions into a single cluster.

The main contributions of our work can be summarized as follows,

- We present an empirical study of the quasi-identity of events in the context of CDEC. In addition to providing evidence for previously studied types of quasi-identity (membership, subevent), we identify a novel type relating to the spatiotemporal continuity of events.
- We release a densely annotated CDEC dataset, CDEC-WN, spanning 198 document pairs across 55 subtopics from English Wikinews. The dataset is available under an open license. To serve as a benchmark for future work, we provide two baselines, lemma-match, and a BERT-based cross-encoder.
- To efficiently collect evidence for quasi-identity, we develop a novel annotation workflow built upon a custom-designed annotation tool. We deploy the workflow to crowdsource CDEC annotations from Mechanical Turk.

In the upcoming sections, we first position our work within the existing CDEC literature (§7.2). We then describe our methodology for preparing the source corpus (§7.3), and our crowdsourcing setup for collecting coreference annotations on this corpus (§7.4). In §7.5, we present a study of quasi-identity of events in our dataset. Finally, in §7.6, we present two baselines models for the proposed dataset.

7.2 Related Work

Event Coreference: Widely studied in the literature, with datasets curated for both within and cross-document tasks. ACE 2005 [209], OntoNotes [101], and TAC-KBP [145] are commonly used benchmarks for within-document coreference. For cross-document coreference, ECB+ [52] is a widely popular benchmark and is an extended version of the original ECB dataset [16]. ECB+ suffers from a major limitation with coreference annotations restricted to only the first few sentences in the documents. However, CDEC is a long-range phenomenon, and there is a need for more densely annotated datasets.

Many other datasets have since been curated for the task of CDEC. Some related works include, MEANTIME [142], Event hoppers [189], Gun Violence Corpus (GVC) [208], Football Coreference Corpus (FCC) [27], and Wikipedia Event Coreference (WEC) [68]. However, most CDEC systems are still evaluated primarily on ECB+. Additionally, all of these datasets do not account for the quasi-identity nature of events.

Though compiled from Wikinews, CDEC annotations in the MEANTIME corpus were limited to events with participants from a pre-defined list of 44 seed entities. While the FCC corpus was also crowdsourced, the annotation unit was an entire sentence instead of a single event mention. WEC corpus uses hyperlinks from Wikipedia but primarily handles referential events. In this work, we use open-domain events and treat an event mention as our annotation unit. We collect coreference links across all the mention pairs from all the underlying document pairs.

Event Identity: Recasens et al. [175] postulated entity coreference as a continuum, with identity, non-identity and near-identity relations. In a follow-up work [177], they identify near-identity relations using the disagreement between annotators. They say subjects are not fully aware of the near-identity behavior, therefore making direct annotation collection hard. The continuum idea has since extended to events [102]. Determining if two event mentions are identical is not a trivial decision. It depends on the arguments of the mentions (often underspecified in the local context), the semantics of the mention, and the document context. In this work, we are specifically interested in cross-document coreference. Wright-Bettner et al. [216] studied the impact of the subevent relationship on quasi-identity, but a more general annotation framework is missing. Accurately capturing event identity is critical to CDEC dataset construction and the subsequent modeling. Therefore, we qualitatively study this phenomenon by collecting supplementary information with each coreference link.

7.3 Corpus Preparation

In our goal of curating a CDEC dataset, we first needed to identify documents that exhibit cross-document coreference. We now describe our document collection process and our methodology for annotating event mentions in these documents.

Document Selection: To facilitate the redistribution of the documents under an open license, we prioritized collecting the documents from publicly available news sources. We chose Wikinews for three key reasons. First, the news articles were sourced from trusted news outlets and reported impartially. Second, these articles are available under an open license (CC BY 2.5), allowing easy redistribution. Finally, each article is human-labeled with categories (e.g., Disaster and accidents, Health, Sports, etc.),[4] as we describe later, this meta-information plays a significant role in our dataset collection. We use the July 1st, 2020 dump of English Wikinews, which contains a total of 21k titles (or articles/documents). These news articles are timestamped from November 2004 to July 2020. Annotating coreference between every document pair in Wikinews is infeasible. Therefore, we first identify groups of related news articles. Articles within a given group usually describe a part of a developing news story or storyline.

Identifying Storylines: To identify these latent storylines, we first construct an undirected Wikinews graph (W) with articles as nodes and add an edge between two nodes if one is mentioned under the "Related News" section in the other. We then identify cliques (C_W) (i.e., fully connected sub-graphs) in the Wikinews graph, which constitute our potential set of storylines. While the articles within each clique are related, we also want to minimize the relatedness of articles across cliques. Therefore, we construct a new graph (M), where each clique ($\in C_W$) is a node, and an edge is added between two nodes if the two cliques are not disjoint or if any two articles in the two cliques share an edge in the Wikinews graph (W). Finally, we extract maximal independent sets from M that correspond to separate storylines. Among the multiple feasible maximal independent sets, we optimize for maximum overlap in Wikinews categories of articles within each clique.

[4] `https://en.wikinews.org/wiki/Wikinews:Categories_and_topic_pages`

# topics	1
# subtopics	55
# documents	176
# sentences per doc (avg.)	14.6
# tokens per doc (avg.)	344
# event mentions	7220
# mentions per doc (avg.)	41
# document pairs	198
# CDEC links	4282
# CDEC links per document pair	21.6
# full coreference links	2914
# partial coreference links	1368

Table 7.1: An overview of the compiled CDEC dataset.

This algorithm satisfies two requirements of a CDEC dataset. First, within each storyline, all articles are related to each other. Second, articles from different storylines aren't adjacent in the Wikinews graph (W); thereby, they are very likely unrelated.

For this work, we narrow our focus only to articles in the "Disaster and Accidents" category on Wikinews.[5] Following the terminology of prior work, our dataset constitutes of a single topic (Disaster and accidents) and 55 subtopics (individual storylines). We restrict CDEC annotations to subtopics that contain 3 or 4 documents. Our algorithm aims for completeness of the CDEC dataset by maximizing for intra-subtopic and minimizing inter-subtopic coreference.

Event Mention Identification: To annotate the event mentions in the above-collected documents, we first run a combination of mention detection systems. Specifically, we use the OpenIE system [190] from AllenNLP [84] and an open-domain event extraction system [6]. The former is effective at extracting verbal events, whereas the latter is good at nominal events. In contrast to most prior work, we do not restrict the mentions to specific event types or salient events. We believe it is important to study all underlying events to achieve a complete understanding of the corpus. Since the quality of mention identification is critical to our CDEC dataset, we ask an expert to go through the automatically identified mentions and add/edit/delete mentions using the Stave annotation tool [123].[6]

Table 7.1 presents the overall statistics of our document corpus. Our documents are $\sim$14.6 sentences long, comparable to prior work, ECB+ (16.6), GVC (19.2), and FCC (34.4). However, our documents are significantly more dense in terms of event mentions. Our documents contain $\sim$41 mentions (on avg.), much higher compared to prior work, ECB+ (15.3), GVC (14.3), FCC

[5] `https://en.wikinews.org/wiki/Category:Disasters_and_accidents`
[6] the expert annotator is an author of this work.

(5.8). Given the dense nature of our documents, we appropriately design our annotation task and interface.

7.4 Annotating Coreference via Crowdsourcing

Corefering event mentions share their identity. However, the extent of sharing for them to be considered coreferential is unclear. To empirically study this behavior, we crowdsource annotations on Mechanical Turk. We use the crowd workers' responses to analyze the influence of quasi-identity on coreference decisions.

7.4.1 Annotation Task

The input to our annotation task constitutes a pair of documents, with all event mentions pre-identified. Annotator iterates through every mention on the left document and select corefering mentions from the right document. We also provide the document titles and publication dates to help set the context for the articles. Note that we focus solely on cross-document coreference in this work and leave the addition of within-document links to future work.

Prior work has highlighted the difficulty in capturing event coreference, specifically in cases where the mentions are only quasi-identical [102]. Notably, Recasens et al. [177] found direct annotation of partial identity to be a difficult task. Therefore, we propose to analyze this behavior by collecting supplementary information from the annotators. For each coreference link created by an annotator, we ask them four *follow-up questions*, 1. overlap in location, 2. overlap in time, 3. overlap in participants, and 4. potential inclusion relationship.[7] Annotators implicitly consider these aspects when making a coreference decision; therefore, responding to these questions won't increase the annotators' cognitive load significantly. As we show in §7.5, the responses to these questions help us tease apart the cases of partial identity.

Unlike within-document coreference, disjoint narratives between documents often complicate CDEC annotation tasks. Wright-Bettner et al. [217] analyzed this behavior in detail and proposed a new `contains-subevent` label for within-document links that improved annotator agreement and reduced inconsistencies. However, they rely on experts to create the within-doc `contains-subevent` label beforehand. Instead, we focus solely on cross-document links and frame the task as a simple pair-wise classification. Our framing allows non-expert annotators to make decisions without concern for complex granularity issues. Our follow-up question regarding inclusion facilitates a post hoc analysis of the event granularities in our dataset.

To ensure completeness of our CDEC dataset, we collect annotations for each pair of documents in a given subtopic (§7.3). As highlighted earlier, the quasi-identity of events may or may not allow for the application of transitivity property. Therefore, in our dataset, we cannot expand coreference links using transitivity. So collecting annotations between each pair in a given subtopic is necessary.

[7] see Table A.11 in Appendix for the exact formulation of these follow-up questions.

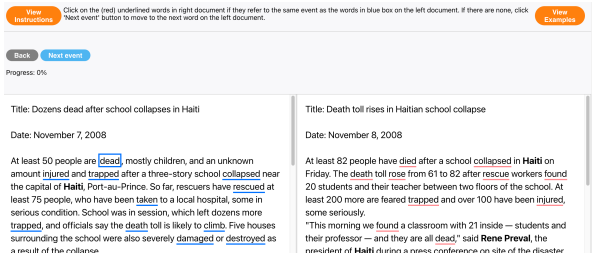

Figure 7.2: Tool for annotating cross-document event coreference. The two documents are shown side-by-side, with event mentions pre-highlighted. We provide on-screen instructions as well as dedicated pages for viewing detailed instructions and examples. As seen in the example here, we allow annotation of every pair of mentions in the given document pair. In our annotation effort, we present every pair of related documents on this tool, leading to a *densely* annotated dataset.

Annotation Guidelines: Events are commonplace in the newswire; therefore, it is feasible to explain the concept of events and their coreference via simple example-based guidelines. In our guidelines, we first define *events* and then provide numerous examples of identical and non-identical event mentions, with detailed explanations. Following prior work [189], we rely on the annotator's intuition to decide coreference.[8]

7.4.2 Annotation Tool

To efficiently crowdsource annotations, we require a tool that is both easy-to-use and customizable to our workflow. For this purpose, we build upon the Forte[9] and Stave[10] toolkits [123]. We extend both the toolkits to support cross-document linking as required by our annotation task. Figure 7.2 presents a snapshot of our annotation interface. We highlight event mentions in both the documents and allow the annotator to iterate through each mention on the left document. In addition to dedicated links to instructions and examples, we provide on-screen instructions to assist the annotator in real-time. We also use an English NER tool [131] to highlight the named entities in the documents. These entities help the annotator keep track of various event participants in the two documents.

We utilize this tool for our entire dataset collection. While we show an application of our annotation tool for CDEC, we believe it's adaptable to other cross-document tasks like entity coreference and event/entity relation labeling tasks. We will release our toolkit to encourage future work on cross-document NLP tasks.

[8] see A.2 in Appendix for complete guidelines.
[9] `https://github.com/asyml/forte`
[10] `https://github.com/asyml/stave`

7.4.3 Collecting CDEC annotations

We crowdsource annotations for CDEC using Amazon Mechanical Turk (MTurk). Each Human Intelligence Task (HIT) constitutes annotating cross-document links for one pair of documents. We obtained IRB approval and set our HIT price based on preliminary studies.[11] On MTurk, we restricted our HITs to crowd workers from the US and set our qualification thresholds for % HITs, and total HITs approved as 95% and 1000 respectively. We paid a fair compensation of $10.9/hour on average.[12] Our annotation task requires proficiency in English, as well as a good understanding of event coreference. To this end, we attach a qualification test with eight yes/no questions regarding event coreference, with a qualification threshold of 75%.[13]

For each document pair, we collected annotations from three different crowd workers. In each task, crowd workers go through the two documents and develop a high-level understanding of the news story. They then iterate through the mentions in the left document, in the narrative order, to identify potential cross-document coreference links. From our preliminary studies, we found that annotators spend considerable time reading the two documents. Therefore, to make the best use of the crowd workers' time and effort, we group HITs that constitutes document pairs from the same subtopic. This way, if the crowd worker chooses to, they can annotate the entire subtopic in one sitting, sharing their understanding of a document from one HIT to the next. In total, we collected annotations for 198 document pairs, spanning 176 unique documents and 55 subtopics from 46 crowd workers.

Inter Annotator Agreement (IAA): For each pair of documents, we collect annotations from three crowd workers. Our setup allows the annotator to decide coreference for every mention pair. To measure IAA, we associate a value to each mention pair (corefering or non-corefering) and compute Krippendorff's α. For coreference links, we observed an α of 0.46, indicating moderate agreement [7].[14] Additionally, we compare the impact of the quasi nature of coreference on the annotator agreement. In our dataset, 31% of the full-coreference links have a perfect majority (3/3 annotators). However, only 13% of the partial-coreference links have the same (see section 7.5 for the methodology used to determine partial coreference). This sharp contrast illustrates the difficulty in capturing partial coreference links.

Selecting CDEC links: For each pair of mentions, we take a majority vote on the three crowdsourced annotations. In our preliminary analysis, we found many valid coreference links annotated by just one crowd worker. While we encourage the crowd workers to annotate every pair of corefering mentions, they occasionally miss links. Therefore, to ensure completeness of our dataset, we use an adjudicator to go through the single-annotator links to decide if they are in-fact corefering or not.

[11] see A.1 in Appendix for more details.

[12] The median pay was slightly higher at $16.3/hour. Both mean and median pay are above the current minimum wage requirements in the United States.

[13] see A.4 in Appendix for the test format and the questions.

[14] It's important to note that we compute IAA on our entire dataset. Our IAA score is comparable to those of quasi-relations from Hovy et al. [102].

Table 7.1 presents an overview of the compiled CDEC dataset. Unlike prior work, we do not create mention clusters by expanding the links via transitive closure. As we show in §7.5, quasi-identity of events warrants the need to analyze coreference at the level of mention pairs instead of clusters.

7.4.4 Dataset Validation

To facilitate benchmarking future coreference resolution models, we split our dataset into train and test. Of the 55 subtopics, 40 are for model training and development, and 15 are for the unseen test set. Given the importance of the test set quality, we perform expert validation on a randomly selected subset of 18 document pairs from our test set. The expert inspected the annotated coreference links in the subset and found 97.5% precision (549/563 were corefering). On the other hand, measuring the recall is hard due to a large number of mention pairs. Therefore, we specifically focus on two types of potentially missing coreference links, 1. mention pairs that share the same head lemma (but not annotated as corefering), 2. mention pairs that are part of a non-transitive triplet.[15] Upon inspection by the expert, we find that majority of lemma-match links are non-corefering (50/565 were corefering), while a majority of non-transitive pairs are corefering (149/173 were corefering). This result indicates the scope for improvement in tackling missing coreference links. We leave this extension to future work.

7.5 Studying Quasi-Identity of Events

Numerous factors determine the identity of an event mention, including the semantics of the mention, arguments (place, time, and participants), and the overall document context. Therefore, overlap in these factors determines the extent of coreference between two given mentions. This overlap leads to cases of partial (quasi-) identity. Our annotation workflow allows for empirical investigation of this phenomenon, and we summarize our observations through a taxonomy of event identity in Figure 7.3. Except for Wright-Bettner et al. [217], prior CDEC datasets do not account for the partial identity during the annotation process. Hovy et al. [102] have previously proposed two types of partial identity, membership, and subevent. In addition to providing evidence for these two types in our dataset, we also identify a novel type of partial identity termed as *spatiotemporal continuity*.

Collecting Partial Identity: We use the responses to follow-up questions for qualitatively analyzing cases of partial identity. We consider a link to be a case of partial identity if a strict majority of annotators indicate one of the following. First, there is an inclusion relationship between corefering mentions. Second, the two overlap in place, time, or participants. With this screening methodology, we found $\sim$32% of the total CDEC links to be candidates for partial identity (Table 7.1). We qualitatively analyze the dataset and identify three types of partial identity, 1. Membership, 2. Subevent, and 3. Spatiotemporal continuity. Table 7.2 illustrates each type with examples from our compiled dataset.

[15] (E_A, E_B, E_C) is a non-transitive event triplet if E_A corefers with E_B, E_B corefers with E_C, but E_A and E_C are non-corefering.

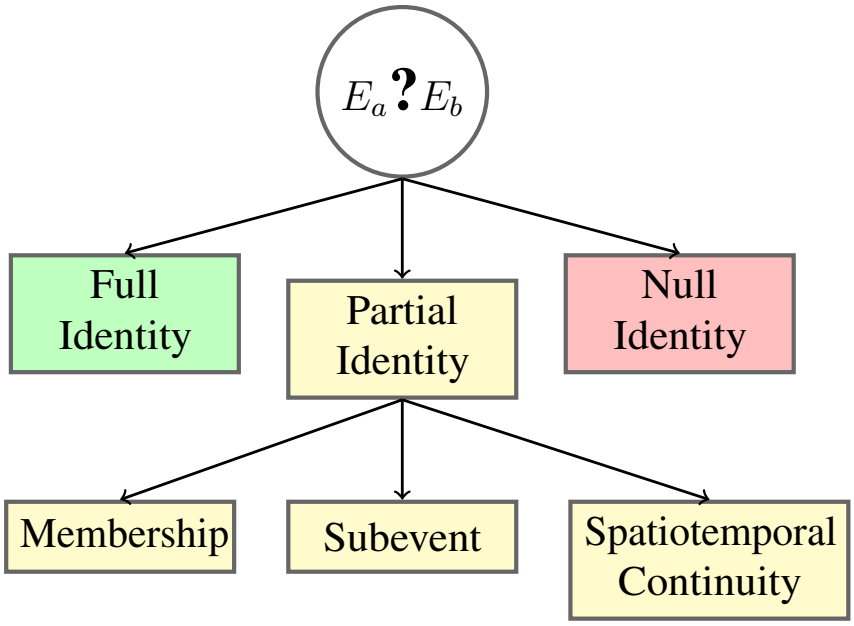

Figure 7.3: A taxonomy of event identity. While full and null identities are well understood, the definition of partial identity is still evolving. We present the three types of partial identity found in our dataset.

Membership: An event mention E_a is a member of event mention E_b. Consider the two sentences, 1a, and 1b. The mention 'fire' (1a) denotes a specific wildfire, whereas 'wildfires' (1b) denotes a group of wildfires, including the one in 1a. The concept of partial identity often challenges the transitivity assumption of coreference. For instance, the mentions [smaller] 'aftershocks' (2b) and [7.1] 'aftershock' (2a) share no identity, thereby, non-coreferential. However, both the mentions partially corefer with [60] 'aftershocks' from 2c.

Subevent: An event mention E_a is a subevent of event mention E_b. This behavior can be seen in the coreference between the 'crash' event from 3a, and the 'accident' event from 3b. While the 'accident' event involves many individual events, derailed, caught fire, spill chemical, and release fumes, it partially corefers with the event 'crash' from 3a that likely refers only to the derailment. Similarly, consider the case of the Boston Marathon Bombing in examples 4a and 4b. The 'bombing' event from 4a refers to the whole incident, whereas the 'explosions' in 4b refers to specific subevents of the 'bombing'.

Spatiotemporal Continuity: The identity of an event can *continuously* evolve over space and time. Consider the two mentions, 'storm' and 'Hurricane' from Table 7.2 (5a, 5b). At a high level, these mentions are corefering because they denote the same event (storm Richard). However, the expressions of this event differ slightly across the two documents. In the former, it's a storm (with 70mph winds) having an impact in Honduras, whereas, in the latter, it's a hurricane (with

90mph winds) impacting Belize. Similar behavior is visible with the [Haitian cholera] 'outbreak' event from Figure 7.1. The outbreak gradually evolves, with growing infection (2600 → 3000 → 4000) and deaths (200 → 259 → 292). In both of these examples, we observe the event changes gradually and is always continuous in both space and time dimensions.[16]

In line with prior work on entities [173], we believe identity and coreference of events to be a continuum. Our dataset already includes many instances of partial identity to support this hypothesis. The above-described cases of partial identity (membership, subevent, and spatiotemporal continuity) will pose new challenges to future dataset collection efforts. We believe our annotation workflow and guidelines will be of use to future work.

In this section, we establish a clear case for tackling partial identity within the coreference resolution task. However, in practical settings, the boundaries between full, partial, and null identities remain fuzzy. As seen in our analysis on the inter-annotator agreement, humans find it hard to identify cases of partial coreference. In the downstream coreference resolution task, users are primarily interested in knowing if two given mentions share an identity or not. Therefore, we propose to view both full and partial identity under a single coreference label ('coreference') and contrast them against cases with no shared identity ('non-coreference'). Compared to prior datasets, this presents new challenges in tackling partial identity within the 'coreference' label.

7.6 Baselines

We define the task as a mention pair classification problem. Due to the quasi-identity nature of event mentions (§7.5), we do not cluster mentions in coreference groups. Additionally, we consider both full and partial identity under the coreference label. We present two baseline models, lemma-match, and a cross-encoder model. We split the dataset of 55 subtopics into train and test, with 40 subtopics for training and development, and 15 subtopics for the held-out test set. For our experiments, we assume gold mentions and subtopic information.[17]

Lemma-match: For our first baseline, we implement the traditional lemma-match baseline. We use spacy's large model[18] to extract the head lemma of the event mentions, and consider two mentions corefering if the lemma's match. Following Upadhyay et al. [199], we also experiment with a Lemma-δ baseline. In our experiments, we found the best dev performance with δ=0, resolving to a simple lemma baseline. This could be due to our assumption of access to gold subtopic information.

Cross-Encoder: As a second baseline, we implement BERT-based cross-encoder model. The input consists of a pair of sentences with both mentions highlighted using special tokens to indicate the start and end of mention spans (¡E¿, ¡/E¿). We first concatenate the two event-tagged sentences (with [SEP] token) and pass it through a bert-base-uncased encoder. We then perform mean pooling on the event start tags (¡E¿), and pass the pooled embedding through a linear

[16]We borrow the term spatiotemporal continuity from the Philosophy literature. It describes the properties of well-behaved objects [212]. A similar treatment for entities is presented in Recasens and Hovy [173].

[17]topic-level performance [29]

[18]`en_core_web_lg` from `https://spacy.io`

classification layer to predict coreference vs. non-coreference. For training the cross-encoder, in addition to the positive coreference pairs, we generate two types of negative mention pairs. For the first type, we collect non-coreference mention pairs from sentences that have a coreference link between a different mention pair. For the second type, we extract non-coreference mention pairs from random sentence pairs between the documents. During training, we use a dataset ratio of 1:5:5 (positive:negative-I:negative-II). We use huggingface transformers [215], and train the model using AdamW [127] with an initial learning rate of 2e-5. We also use a linear warmup scheduler, with 10% of training steps for warmup. We finetune the # epochs and positive:negative dataset ratio during the development stage (5-fold cross-validation) and use the best configuration when training on the entire train set.

Results: Table 7.3 presents the results of our baselines. For model development, we perform 5-fold cross-validation on the training set (40 subtopics). To report the results on the held-out test set (15 subtopics), we train the model's best configuration on the entire training set. We report precision, recall, and F1 scores of the coreference label averaged on five different runs. The lemma baseline only achieves an F1 score of 48.2, indicating that the proposed dataset is lexically diverse. The cross-encoder improves upon the lemma baseline, especially on the recall. Upon inspection of development set predictions, we observe two possible error cases for the cross-encoder model. First, the model struggles at the cases of partial identity ('explosion' vs. 'incident' and 'evacuate' vs. 'evacuations'). This drawback of cross-encoder indicates that the model requires a deeper understanding of event identity. Second, the cross-encoder model is often limited by the information available in a single sentence. It is known the event arguments are often under-specified in the local context [67]; therefore, increasing the context to a paragraph or the entire document might help improve the performance.

Membership	
1a	The **fire** has burned about 4400 acres so far and 15 homes have been lost, however there have been no reported injuries or deaths.
1b	Reports say that the amount of people fleeing from their homes in California located in the United States due to **wildfires** has reached the 1,000,000 mark as the fires continue to grow.
2a	Several aftershocks have rocked the same area, the latest measuring 7.1, had a depth of 10 km. It was first reported to be a 7.3 **aftershock**.
2b	Some smaller **aftershocks** with magnitudes between 5.2 and 5.7 were also reported in the region.
2c	That quake was followed by as many as 60 **aftershocks** for at least a week, with some ranging as high as magnitude 7.8.
Subevent	
3a	A freight train in Lviv, Ukraine derailed, caught fire, and spilled a toxic chemical, releasing dangerous fumes into the air early Tuesday morning (local time), and people who live near the site of the **crash** are still becoming sick.
3b	The available information about the phosphorous cloud following the railway **accident** in the Ukraine last Monday is becoming more and more cryptic.
4a	During the fifteen days of the trial, the prosecutors called 92 witnesses to testify as to the chaotic scenes following the **bombing**.
4b	Two **explosions** within seconds of each other tore through the finish line at the Boston Marathon, approximately four hours after the start of the men's race.
Spatiotemporal Continuity	
5a	Tropical **storm** Richard is nearing hurricane strength with winds of 70 mph (115 kph) as it lashes Honduras with heavy rains
5b	**Hurricane** Richard made landfall in Belize about 20 mi (35 km) south-southeast of Belize City with winds of 90 mph (150 kph) at approximately 6:45 local time (0045 UTC) according to the National Hurricane Center (NHC)

Table 7.2: An illustration of quasi-identity of event mentions across documents. These examples cover the three identified types of quasi-identity, membership, subevent, and spatiotemporal continuity.

Model	**Dev**			**Test**		
	P	R	F1	P	R	F1
Lemma-match	46.6	54.9	49.9	42.3	56.0	48.2
Cross-Encoder	43.1 $\pm$ 0.6	75.4 $\pm$ 0.5	54.3 $\pm$ 0.5	45.9 $\pm$ 0.8	77.3 $\pm$ 1.1	57.6 $\pm$ 0.6

Table 7.3: Baseline results on development and test sets. For cross-encoder, we report the average scores and their standard deviation across five runs.

Chapter 8

How do Language Models Learn about Coreference

8.1 Introduction

Throughout this thesis, we have dived deeply into the complexities of event semantics and their interactions with other discourse elements, recognizing their crucial role in natural language understanding. We have also investigated ways to increase data availability for these tasks through crowdsourcing and indirect supervision. In this chapter, we focus on Large Language Models (LLMs), one of the most prominent methods for generalizing model ability across tasks via scaling. LLMs can solve many NLP tasks very well, even achieves good performance on complex tasks. Building on these advancements, we seek to answer the following research questions here:

1. *How (well) do large language models (LLMs) manage the complexities inherent in event semantics and anaphora resolution?*
2. *Given that models capable of solving complex semantic tasks tend to be large and hence challenging to analyze, can we come up with effective analysis methods?*

Winograd Schemas. We choose to use the Winograd Schemas to study these problems. The Winograd Schema Challenge (WSC) [114] is designed to evaluate a model's ability to resolve coreference with common sense reasoning. Each schema consists of sentence pairs where a single word change, often a key verb, leads to a completely different interpretation. Here is a typical question from the Winograd Schemas: "*The city councilmen refused the demonstrators a permit because they (**feared / advocated**) violence.*" Here, the pronoun "they" would be "the city councilmen" if the verb is 'feared' by "the demonstrators" if the verb is "advocated. We argue that the WSC provides an excellent venue to study our research questions.

Firstly, the schemas are challenging for language models but straightforward for humans. For instance, Llama3 8B scores 76.1% on the Winogrande dataset, whereas human performance is 94.0%. Secondly, studying the schemas allows us to examine how LLMs handle coreference beyond simple surface matches. Notably, events play a crucial role here, as the key changes between pairs often involve event verbs, aligning with our research theme. Additionally, observing how LLMs handle the WSC reveals how small textual changes lead to different predictions,

providing insights into the propagation of these perturbations through network layers. Lastly, the pairwise and subtle differences between schema pairs offer a unique opportunity to design efficient analysis methods.

Circuit Analysis. The recent mechanistic interpretability line of research is to reverse engineer the computation carried out by powerful models such as the Transformer. One particular direction is to identify circuits: subgraphs that implement human-interpretable algorithms [69]. Recent studies show promising results in revealing interpretable algorithms that implement specific functionalities. For example, the circuit analysis of an Indirect Object Identification (IOI) [210] tasks reveals LLMs use a simple algorithm with many functional attention heads to solve the task.

An IOI task is based on sentences with a dependent clause: "When Mary and John went to the store" and a main clause: "John gave a bottle of milk to". And the task is to complete this sentence by choosing between the two names. The more probable answer is "Mary". It is shown that GPT2 implements a circuit that approximate the following simple algorithm with a combination of attention heads (Fig. 8.1).

1. Identify all previous names in the sentence (Mary, John, John).
2. Remove all names that are duplicated (in the example above: John).
3. Output the remaining name.

Although simple, this algorithm demonstrates how language models can propagate information and form algorithms with network structures such as residual streams and attention heads. It remains an interesting question how a circuit would look like for semantic reasoning.

Our Approach. While complex LLM models can handle real semantic tasks more effectively, their complexity poses a significant challenge for circuit analysis. As shown in Fig. 8.1, the search space and resulting circuits of a 7-billion-parameter model is much larger than that of GPT-2-small (117 million parameters). Furthermore, while all prior study mostly focus on one specific algorithm or function, we believe there will be a diverse set of "algorithms" for different Winograd schema task. The extended search space from both angle renders a complete manual circuit analysis almost impossible. To address this challenge, we propose the following methods:

1. Narrow Down Data Samples: The structure of the Winograd Schemas allows us to focus on samples that the model can answer reliably. We only target schemas where the model is correct on the pairs. Additionally, we perform a longitude study by evaluating the model at various training checkpoints to find samples it consistently answers correctly over time. The checkpoints are resources provided by LLM360 [125], which we will introduce later in session 8.3.2. By narrowing down the samples, we can focus on the ones where we can potentially find interesting circuits.
2. Automatic Circuit Finding: IOI circuits are initially found using a technique called activation patching, which is costly and requires extensive human intervention, making it impractical for larger LLMs on real tasks. Instead, we adopt recent automatic circuit discovery methods. Specifically, we use the Information Flow Routes (IFR) algorithm [78], an attribution method that replaces the need for patching. As demostrainted in [78] and Fig. 8.1, this method can successfully identify the IOI circuit automatically.

3. Zoom in Key Subgraphs: We enhance the IFR algorithm with a simple contrastive method that replaces the activation with the contrast activation value between pairs. This method effectively highlights the key computational differences within a Winograd Schema pair, revealing whether an LLM correctly picks up the context clues. We find that this technique is important since the key circuit edges are often not picked up by the original IFR algorithm.

With these tools, resources, and techniques, we are able to conduct model analysis. To gain insights into how the model's abilities emerge, we analyze the performance and circuits of LLMs at various stages of pretraining. By comparing the circuits obtained at different checkpoints, we hope to observe whether and when the model begins to form robust algorithms.

The rest of the chapter will detail our experiment setup and findings. We find that language models can quickly develop "shortcut" strategics for certain data samples, but also form seemly complex algorithms to solve Winograd Schemas. We hope the findings could contribute to a deeper understanding of how language models learn coreference resolution, offering insights for interpretability and model development for NLP.

8.2 Related Work

Interpretability and Circuit Study of LLMs. Mechanistic interpretability aims to reverse engineer neural networks, focusing on understanding the internal mechanisms at work. A significant paradigm in this field is the analysis of circuits, which originated with vision models and has extended to transformer language models [93, 119, 138, 139, 156, 195, 203, 210]. Increasingly, research has sought to characterize the individual components within these circuits, examining attention heads [35, 87, 136, 157], neurons [1, 81, 90, 181, 205, 206], and use methods such as dictionary learning and sparse autoencoders [24, 104, 134] to learn interpretable features at scale. And more recently, advancements have been made to conduct interpretation with scalable [78, 94, 147] and automated methods [20, 49, 193], in place of conducting activation patching and examination manually [157, 205]. However, due to the complexity of both the networks and the actual language problems, most of the methods that tries to uncover the LLM "algorithms" focus on small or artificial tasks, or individual functional heads, where a simple human interpretable algorithm is available, such as the Indirect Object Identification circuit [210], the Greater Than circuit [93], the Correct Letter circuit [119], the Copy Suppression Head [136], the Successor Head [87], and Gendered-Pronoun Resolution [205]. Most circuit studies are on small or artificial tasks or individual functional heads. In this chapter, we study a real complex semantic task, the Winograd Schemas.

Developmental Study of LLMs. To understand the development and emergence of abilities in LLMs, extensive research has been conducted on their evolution throughout the pre-training process. Recently, [194] examined whether circuit analyses remain consistent across different stages of training and model scales. However, most prior work on developmental studies has focused on making high-level claims about model performance and abilities, such as investigating linguistic capabilities in syntax acquisition [218], word acquisition [34], and general skills like memorization [18]. While these behavioral studies may guide research at a high level, they do not reveal the internal implementation of the models.

In this chapter, we examine LLMs through both mechanical analysis and developmental study, providing a more holistic view of LLM behaviors. Additionally, rather than focusing on small or artificial problems, we study a challenging and complex linguistic phenomenon like the Winograd Schema, using a modern 7B parameter model.

8.3 Experiment Setup

In this section, we describe experimental setup, including the dataset we used, details of the algorithms and our solution to narrow down the search space.

8.3.1 Dataset

The Winogrand dataset. Our primary dataset for this study is Winogrande, a large dataset with Winograd Schemas questions, created via adversarial methods to reduce simple surface-level biases. The original Winograd Schema Challenge [114] dataset is not used because many of the samples are available online and may have been seen by the model during pretraining. In our preliminary analysis, we found that the AMBER model achieved a 78% 0-shot accuracy on WSC, which should be too high for the model. Moreover, the Winogrande dataset is larger, offering more samples for study. We run the experiments on the 1267 test samples from Winogrande.

We further categorized the dataset samples based on their difficulty for the LLM. We categorize the samples into three buckets: *simple bucket* (correctly answered by checkpoints from first 30% of pretraining), *medium bucket* (consistently answered correctly by checkpoints that go through more than 40% of pretraining), and *difficult bucket* (not consistently answered correctly by any checkpoints, until the end of training). The medium bucket should be the most interesting segment since we believe the model may have found non-trivial circuits.

To study circuits in auto-regressive models, one constraint is that the task has to be "next token prediction". Most Winograd Schema data samples are formed as "filling the blank" and the important clues to the answer are often towards the end. If we convert the task to a multiple-choice question and ask the model to output the option, this will further involve confounding factors such as attention heads that copy options [119]. To get around this problem, we manually curate a few sentences from each bucket to move the "blank" to the end of the sentence. For example, we convert the clause of the original sentence: "*In the hotel laundry room, Felicia burned Mary's shirt while ironing it, so the manager gave () a refund.*" into "so the manager refunded ()". Note that this process is not always possible for all sentences.

After these steps, we end up with a small set of data samples for further investigation, from which we selected 15 samples (Fig. 8.1).

8.3.2 LLM360

In order to perform the longitudinal study, we leverage the LLM360 checkpoints to examine how language models learn about coreference, particularly through analyzing circuits formed during the training process. Specifically, we use the AMBER model, we take 40 checkpoints evenly over

the whole pretraining process, including the final checkpoint. We provide an brief overview of this model in this section.

Model Configurations. We trained a 7B parameter model named AMBER trained on approximately 1.26 trillion tokens (see Table B.2 for the dataset breakdown). The model shares architectural similarities with LLaMA 7B, including model dimensions, use of RMSNorm [221], and rotary positional embeddings (RoPE) at each layer of the network [191], a summary of model specification is in Table B.2.

Model Checkpoints. LLM360 models are released with all intermediate checkpoints saved during training, including model weights and optimizer states. These checkpoints enable continued training from various starting points and facilitate post-training research.

Metrics. The LLM360 project also provides access to detailed logs and intermediate metrics, including system statistics, training logs, and evaluation metrics. Fig. 8.2 shows the model's performance improvement on Winogrande over time, landing at 64.24% at the end.

The availability of detailed metrics and intermediate checkpoints from the LLM360 project enables this study. The metrics and checkpoints allow us to observe the development of specific circuits over the course of training, providing insights into how the model learns to resolve complex anaphora problems. Note that though the Pythia [19] project also provides model checkpoints over the pretraining lifetime, AMBER is a modern language model that is larger and have better performance on difficult datasets like the Winogrande.

8.3.3 Automatic Circuit Discovery

We adopt the Information Flow Routes algorithm (IFR) [78] to conduct automatic circuit analysis. The Information Flow Routes algorithm extracts important subgraphs in a top-down manner by tracing information back through the network, starting from the final token representation. In the information flow graph, nodes represent token representations and edges represent operations that move information across these nodes. For example, token representations at different layers are connected through attention and residual stream edges. This method uses attribution to determine edge importance, which is significantly faster than patching techniques.

The key of the algorithm is attribution, which is based on ALTI (Aggregation of Layer-Wise Token-to-Token Interactions) [76], ALTI estimates the importance of each vector (edge) to the overall sum (node) is proportional to its proximity to the resulting sum. Formally, the edge values as z_i, then the sum is $y = z_1 + \cdots + z_m$,

$$\text{importance}(z_j, y) = \frac{\text{proximity}(z_j, y)}{\sum_k \text{proximity}(z_k, y)}, \tag{8.1}$$

$$\text{proximity}(z_j, y) = \max(-\|z_j - y\|_1 + \|y\|_1, 0). \tag{8.2}$$

In a Transformer model, the key edges exist in the feedforward layers and the attention layers. For multi-head attention, we consider each head as independent edges. After computing the edge

importance, we simply take a subgraph by use a cutoff threshold. Currently we do not have a way to find the threshold automatically, we find that 0.02 and 0.04, similar to the values used in [78] are reasonable and show interesting subgraphs with the regular IFR run. However, we find that the contrastive method yields edge weights in much smaller granularity, and find that a threshold around 0.002 to be more informative. Note that there are no method to guarantee finding the minimal viable circuit. We choose these values to highlight the insights only, they may not generate faithful circuits (a circuit is faithful if all edges outside the circuits would not affect the performance for this task [94, 210]). Detailed formulations of the attention and feedforward layer edges are provided in the Appendix, and we recommend readers refer to [76, 78] for more details.

Narrow Down the Circuits. We further implement a contrastive method utilizing the pair structure in Winograd Schemas. The algorithm computes the edge values for the LLM running through both sentences in the pair. We denote the function M that takes a sentence S and compute all the edge values with an LLM $M(S) = E = \{e_1, e_2, \ldots, e_n\}$, then for a pair of Winograd Schema sentences S_1 and S_2, we apply the following contrastive method:

$$E_c = M(S_1) - M(S_2) \tag{8.3}$$

We replace the edge values E_1 in the IFR algorithm with E_c. This method effectively cancels out the activation values that are the same across both runs. In the Winograd Schema setting, this will cancel most of the activations since the two input sentences differ slightly, leaving only the values and networks impactful to the change of prediction. Note that we find that the scale of these values are much smaller, hence we use 0.002, 10% of the regular threshold.

8.4 How an LLM Solves Winograd Schemas

In this section, we present the experiment results and findings on how large language models (LLMs) solve Winograd Schemas. Our analysis reveals the mechanisms and processes LLMs use to resolve coreferences and understand event semantics. We leverage the Information Flow Routes algorithm and our proposed methods to uncover critical circuits and provide insights into the capabilities and limitations of LLMs in handling complex linguistic tasks.

We run zero-shot evaluation and circuit analysis over the 40 intermediate model checkpoints (spaced evenly) from the AMBER checkpoint series, on the 15 selected pairs, as shown in Table 8.1. The consecutive correct count indicates the its bucket (simple, medium or hard). Overall, the final checkpoint's accuracy is 0.64, and Figure 8.2 shows the performance improvement over time. The model reaches around 0.65 at halfway through the training and does not significantly improve afterwards. We find that a 7B model trained on 1.2 trillion tokens still struggles to predict the pairs consistently. In the Winogrande dataset, we can form 273 pairs in the test set, and the model can solve 75 of them with the final checkpoint, which is 27% of the total pairs. If we consider only the pairs that the model can robustly solve (consecutive count more than 5), there are only 44 left, representing 16% of the total. The fact that the model struggles with consistent predictions and its final accuracy plateaus at halfway through training, indicating that this dataset is still challenging to the model.

Sentence 1	Sentence 2	Options	#Correct
Carla start doing sit-ups and pushups for her weak spots. Her abs needs the	Carla start doing sit-ups and pushups for her weak spots. Her chest needs the	situps/ pushups	38
They had to eat a lot to gain the strength they had lost and be able to work, they had too much	They had to eat a lot to gain the strength they had lost and be able to work, they had too little	work/ strength	30
The stain in the bucket could not be cleaned with the brush because of the soft	The stain in the bucket could not be cleaned with the brush because of the tough	stain/ brush	30
Billy liked watching foreign movies with subtitles unlike Jason because the original language was hated by	Billy liked watching foreign movies with subtitles unlike Jason because the original language was loved by	Lawrence/ Jason	25
Emma had to pay less tax than Mary because less money were made by	Emma had to pay less tax than Mary because more money were made by	Emma/ Mary	22
Keeping the doors closed and the windows opened kept the apartment cool, because the heat was let out by the	Keeping the doors closed and the windows opened kept the apartment cool, because the heat was kept out by the	doors/ windows	18
The gas was not smelling out of the tank but out of the hose because the leaky	The gas was not smelling out of the tank but out of the hose because the sealed	hose/ tank	15
Of the two owners, Jessica was far worse than Jennifer, because dogs will get beaten by	Of the two owners, Jessica was far worse than Jennifer, because dogs will get treated by	Jessica/ Jennifer	12
The woman kept the bikini but returned the top, because the wrong size of the	The woman kept the bikini but returned the top, because the right size of the	bikini/ top	1
In the hotel laundry room, Emma burned Mary's shirt while ironing it, so the manager scolded	In the hotel laundry room, Emma burned Mary's shirt while ironing it, so the manager refunded	Emma/ Mary	1
The wooden doors at my friends work are worse than the wooden desks at my work, because the cheaper wood of the	The wooden doors at my friends work are worse than the wooden desks at my work, because the better wood of the	doors/ desks	0
The trader decided to buy wool and sell cotton because the low price of the	The trader decided to buy wool and sell cotton because the high price of the	wool/ cotton	0
The room at the hotel cost more than the room at the inn because the nasty room at the	The room at the hotel cost more than the room at the inn because the lovely room at the	hotel/ inn	0
The musician liked playing at the auditorium more than at the park because he sounded quieter at the	The musician liked playing at the auditorium more than at the park because he sounded louder at the	auditorium/ park	0
The clothing in the north was warmer than the clothing in the south because there was more snow in the	The clothing in the north was warmer than the clothing in the south because there was more sun in the	south/ north	0

Table 8.1: The selected Winogrande samples, all formatted as next token prediction tasks. The correct counts measure the number of consecutive checkpoints at which the model consistently answers the question correctly until the final checkpoint.

8.4.1 Circuit Development

We compute the circuit subgraphs over all the checkpoints with an edge filtering threshold of 0.02. We can compute how the circuit changes over time by measuring the similarity between the graphs. We use the Jaccard similarity ($J(A, B) = \frac{|A \cap B|}{|A \cup B|}$) of the graph edges as a measure for graph similarity. For each subgraph of the intermediate checkpoint, we compute its Jaccard similarity with the subgraph of the final checkpoint. Figure 8.3 plots the Jaccard similarites of the following sentences.

(4) In the hotel laundry room, Emma burned Mary's shirt while ironing it, so the manager scolded/refunded (Emma/Mary)

(5) They had to eat a lot to gain the strength they had lost and be able to work, they had too much/little (work/strength)

Example 5 is from the simple bucket, where it consistently predicts the pair correctly after 25% of training progress. This is expected since making the prediction here does not require extensive contextual reasoning. The shortcut strategy is that "too much work" is a much more common phrase than "too little work". In the next section, we show that the circuit subgraph of the model focuses on a narrow part of the graph. The Jaccard similarity computed for its subgraph is shown on the right, which is consistent most of the time but remains only slightly above 0.6.

Example 4 is a challenging sample from the medium bucket but on the edge of the hard one, where the model only predicts it correctly at the end of the training. This indicates that the model might not be able to form a consistent algorithm for this sample in the early phase. This can be observed from the Jaccard similarity graph, where the graph similarity fluctuates significantly and is often below 0.5 in the early stages of training. The similarity increases for the final few checkpoints, reaching as high as 0.8-0.9. In the next section, we find that the model seems to find semantically meaningful circuits for this sentence. The circuit for this pair may start to form and stabilize late in the training.

8.4.2 A Closer Look into the Circuits

Now we zoom into specific samples and study their circuit structures. We compute and visualize the circuit subgraph with the original IFR method, and also with our constrastive method. We show three examples, from each bucket respectively.

"Too much work" Circuit. We show the circuits found for Example 5 mentioned above. This example is easily solved by the model. We suspect that the model may use a shortcut algorithm that simply predicting "work" by only reading the clue "too much", since "too much work" is a much more common phrase compared with "too little work". As expected, almost no edges from the early context words such as "gain strength" are linked to the final prediction. The model simply focus on the final two tokens "too much" or "too little". We computed the contrastive graph for this pair[1], and the graph edges are almost empty except for the last token "little", confirming this hypothesis.

[1]This graph is omitted because it is basically empty.

"Keep the Heat" Circuit. We suspect that the circuits for a pair from the medium bucket would be reveal more interesting circuits. Among the samples, we find some interesting circuits for the "keep the heat" sentence pair (Example 6).

(6) Keeping the doors closed and the windows opened kept the apartment cool , because the heat was let/kept out by the doors/windows

We find that the model can successfully predict the right answer consitently, with only half way of the training progress. By examining the base circuit as well as the contrastive circuit, we find that the model effectively pickup semantic related context clues from the sentence. The contrastive circuit especially highlights that the model select signals from many semantic related words, using edges from different layers. For example, the model focuses on keywords like "kept", "open", "heat" etc. Note that although these attention heads should be important for predicting the final output correctly, they are not picked up by the base IFR algorithm.

To take a close look at the model's behavior, we plot the attention and contribution map[2] at a few attention edges in Fig. 8.6. Note that while the attention map exhibits regular behaviors such as diagonal and attending on the first token, the contribution map captures the important information source more clearly. The attention head at (L7, H26)[3] shows contribution mainly on the two different verbs "kept" and "let", potentially used to capture the differences of the events, while the head (L13, H25) shows contribution mainly focus on the two different states "open" and "close". Potentially, the algorithm that solves this pair is to associate "kept" with "close" and "let out" with "open", then find the closest previous entity (door/window). The circuit in this example is slightly more complex than the previous one, requiring the model to at least associate the related verbs. But the model may still rely on simply using the previous word (or the closest entity) to resolve the final entity.

"Scold and refund" Circuit. Example 4 is a more challenging one for the model. But from the Jaccard similarity plot in the previous session, we observe that the circuit starts to become stable at the end of the training. Would this be a signal that the model starts to pick up relevant semantic information? We show in Figure 8.7 the circuits for it, the base circuit and the contrastive one.

The base circuit doesn't highlight links to important contextual information, such as the names of the people and the key event "burned". However, the contrastive circuit shows multiple edges to the key tokens, including the names, the key event "burned". The algorithm to implement this pair might be more complex and inspecting the graph doesn't show a simple insight. Another observation is that the internal activation that favors the the correct answer is also quite small, and the two names have very similar output logit values. In other words, the model rank both answers closely. We suspect the model prediction here can be very fragile. We replace one of the entity with a more rare name (e.g., Cynthia), and we can successfully confused the model to always output the more popular name.

Remarks on Circuit Analysis. From the two of the examples with non-trivial circuits, we find that regular attribution algorithm does not capture the key context clues needed for the Winograd

[2]Contribution is the value computed by Equation 8.1
[3]This means layer 7, head 26.

Schemas. This is not surprising since the differences between the two sentences are very small, hence their signal will be overwhelmed by the internal activation from the rest of the words. This may also due to that our task setup does not require the model to focus on predicting the entity correctly, a different task setup might present different observations.

With the case study on circuits of the samples at different difficulty levels[4], we observe that the model is picking up interesting clues from the context. The contrastive method can help find out the more context-relevant edges comparing to the base IFR method. We hope the proposed methods can shed some light on this direction and help establish a feasible experimental setup.

8.5 Conclusion

In this chapter, we have explored how large language models (LLMs) solve Winograd Schemas, providing insights into both their developmental trajectories and internal mechanisms. By leveraging the special structure of the Winograd Schemas, we identified more robust pairs and conducted contrastive analysis to reveal key circuit components. Our adoption of the Information Flow Route algorithm allowed us to automatically discover circuits. By narrowing down these circuits using contrastive methods and temporal analyses, we pinpointed specific pathways that contribute to the model's understanding of ambiguous anaphora problems.

Overall, our findings highlight both the successes and limitations of LLMs in handling challenging linguistic phenomena. While the models show promise in solving certain tasks, their performance on more complex and nuanced examples remains inconsistent. The analysis also shows that the formation of robust algorithms in LLMs can be a gradual and unstable process, especially for more difficult tasks.

Limitations. The setup used in this chapter is very specific and cannot be easily generalized to other tasks. This setup restricts our study to data samples where the prediction word needs to be at the end. This limitation narrows the scope of our analysis, reducing the applicability of our findings and methods to a wider range of scenarios.

Large Language Models also perform better with certain techniques, such as chain-of-thought prompting and few-shot examples. Reasoning tasks typically benefit from these techniques. However, in this chapter, we study the scenario without any additional prompting to avoid introducing confounding factors. Task-specific circuits would be harder to analyze due to the introduction of in-context learning heads [69, 157]. Nevertheless, we consider the inability to analyze the model with its full potential unleashed a limitation.

The language models we study are also not the most performant ones to date, since the total FLOPs of the model is far behind the best open sourced models. However, we choose the model based on the availability of the full checkpoint sequence, and limitations on computation cost. We believe analysis on larger models with the full sequences, such as LLM360 K2-65B [126].

Future Work. Future research should focus on in-depth circuit analysis to uncover detailed interactions within models. This will involve not only using current methods but also exploring

[4]More circuits can be found in Appendix B.2

new automatic techniques to address scalability issues, such as methods that can explain the circuits automatically [20]. Understanding these detailed interactions can lead to better model designs and improved interpretability of LLMs.

Moreover, expanding the scope of this study to include different datasets and tasks could provide a broader understanding of LLMs' capabilities and limitations. Investigating the effects of different training regimes and architectures on the formation of circuits could also yield valuable insights. For example, there are hypothesis that models trained with programming language exhibit strong reasoning and reference ability. It would be very interesting to apply the circuit methods on models that are stronger at coding.

In conclusion, we hope the methods and resources from this study can provide valuable directions for future research in model interpretability and the development of more robust language understanding systems. By continuing to explore and refine these approaches, we can move closer to creating models that are not only powerful but also transparent and reliable in their decision-making processes.

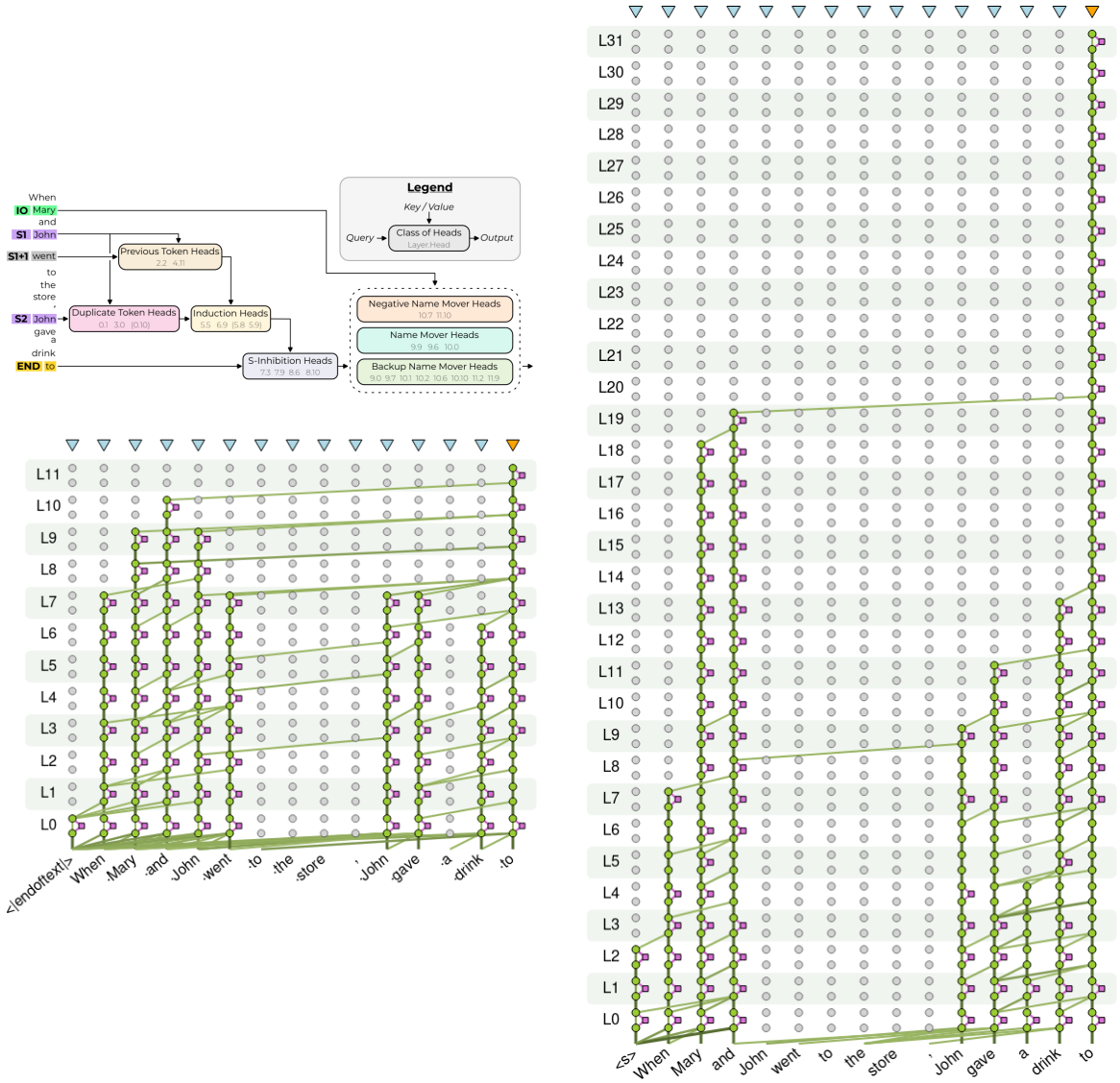

Figure 8.1: **Top Left**: the original IOI circuits identified by Wang et al. [210] on GPT2-small [171]; **Bottom Left**: IOI circuit on GPT-2 using Information Flow Routes [78]; **Right**: the IOI circuit on Amber-7B [125] using Information Flow Routes [78].

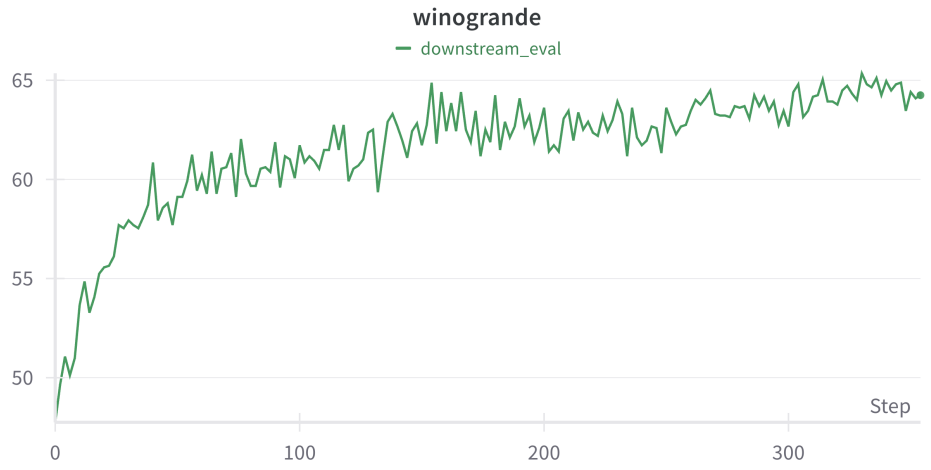

Figure 8.2: AMBER's performance on the Winogrande dataset (5-shot).

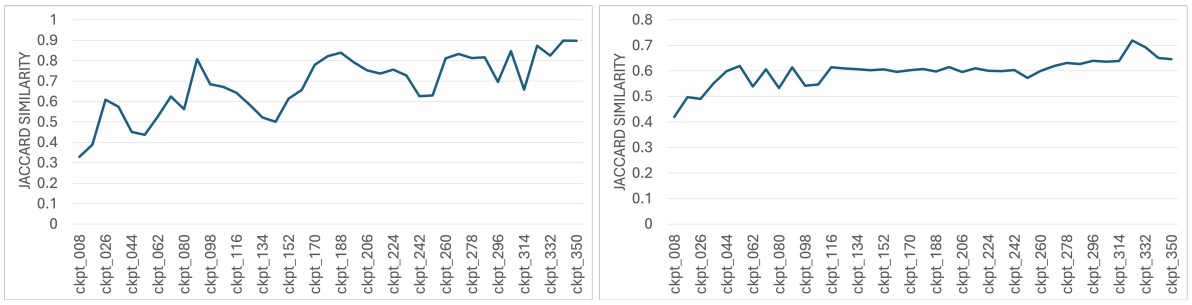

Figure 8.3: Jaccard Similarity of the intermediate checkpoint circuit vs. the final checkpoint circuit. The left figure is on a sentence from the medium bucket, the right figure is computed on a sentence from the simple bucket.

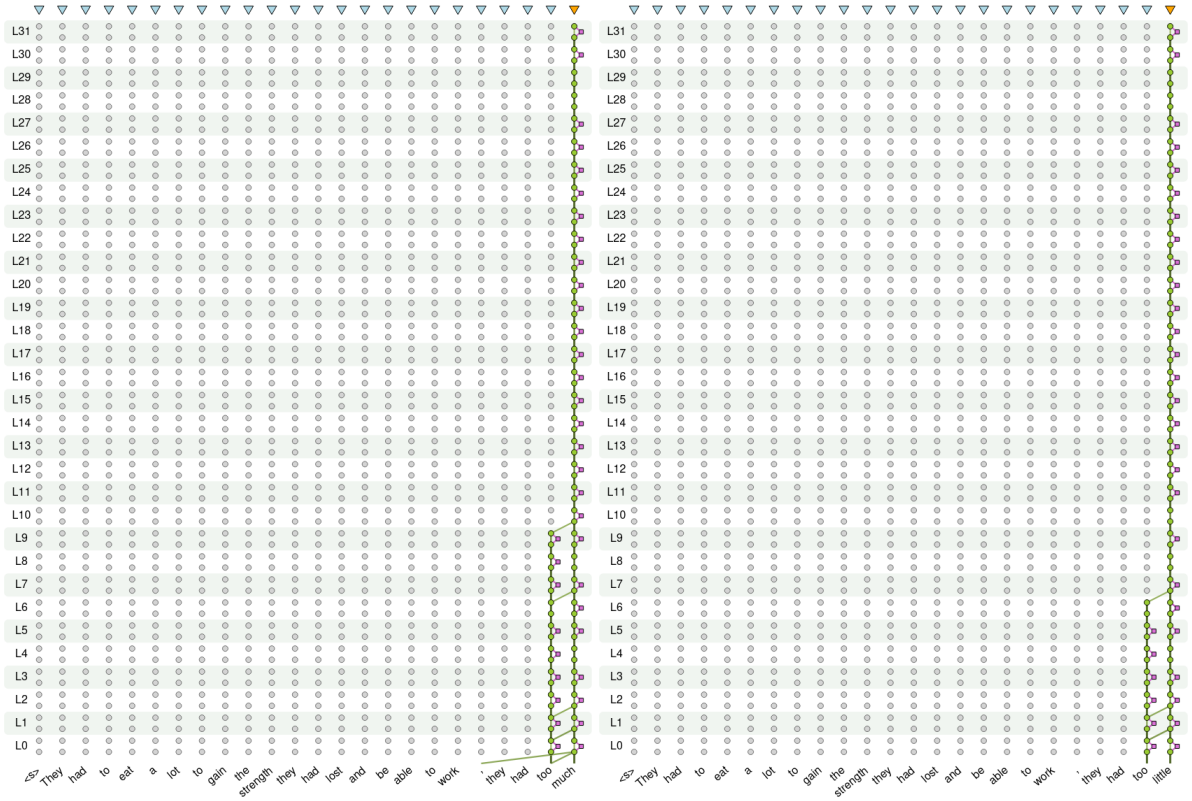

Figure 8.4: Circuits for the ”too much work”/”too little strength” example pair.

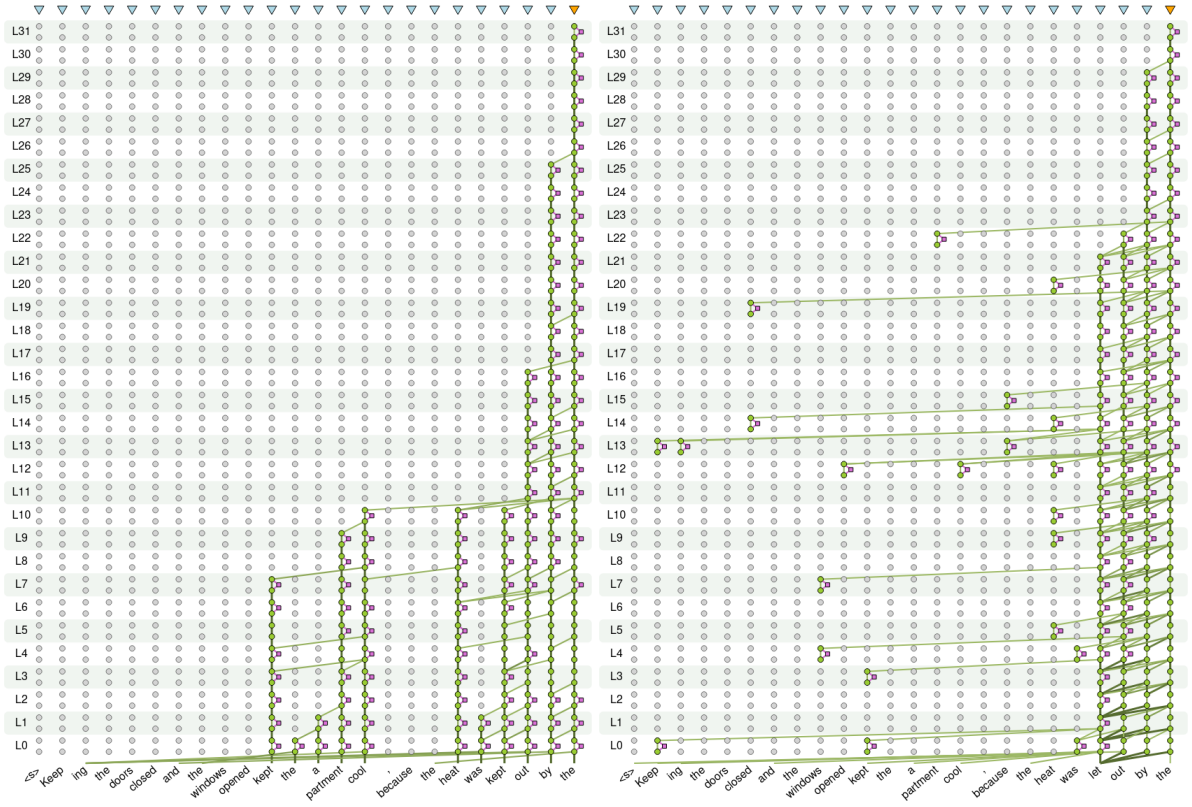

Figure 8.5: Circuits for the “keep heat”/“let out heat” example pair. Left is the circuit of the “kept heat” sentence, Right is the circuit created via the contrastive method

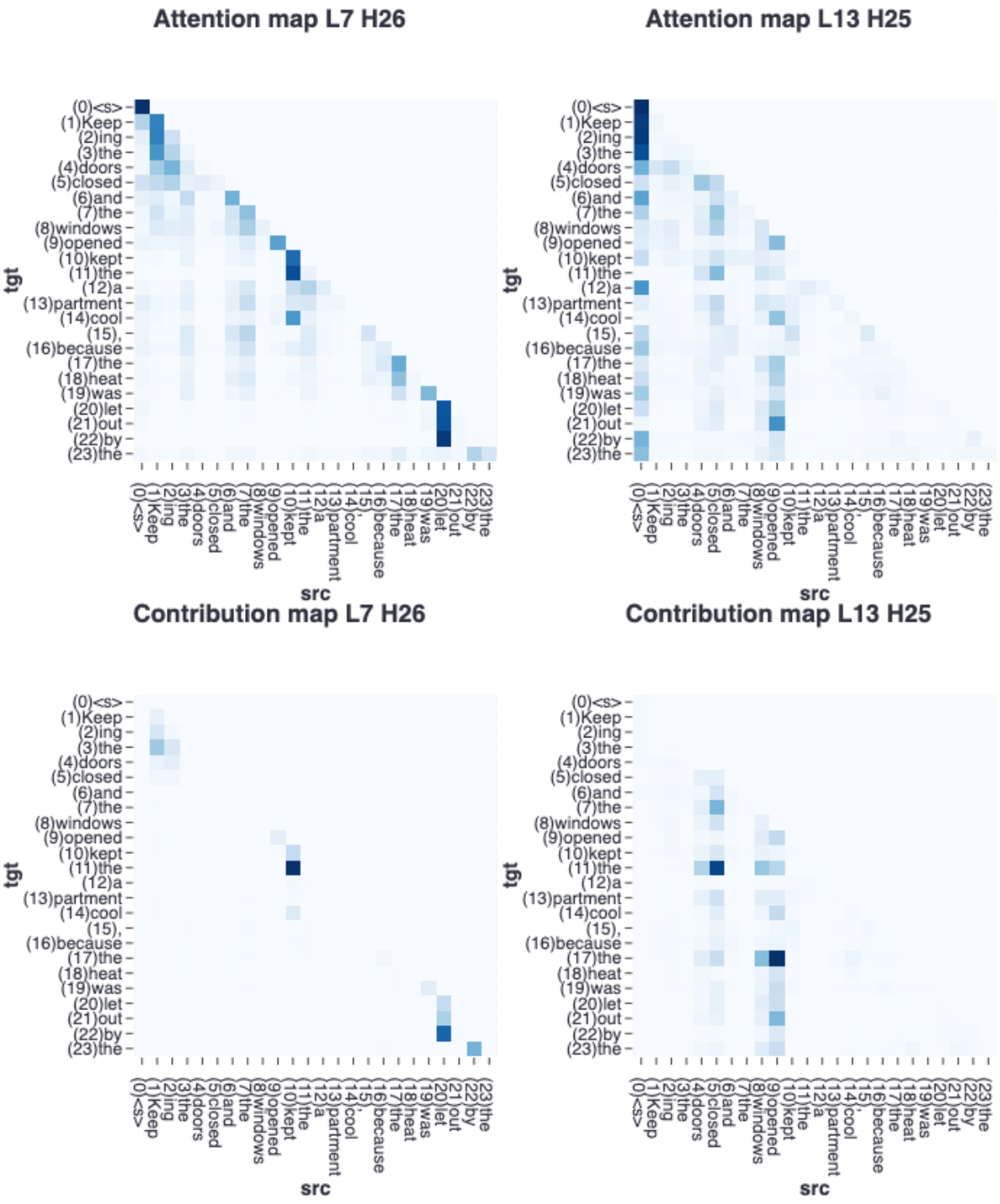

Figure 8.6: Attention and Contribution Graph of selected attention edges from the “Keep the Heat” contrastive circuit.

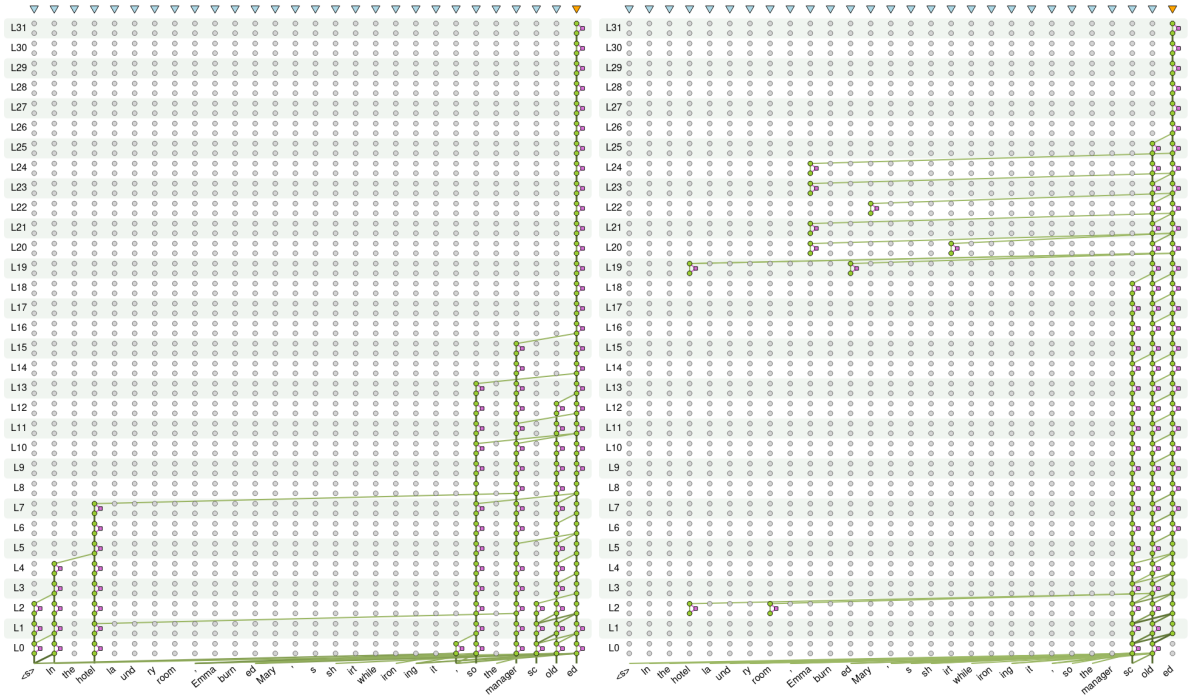

Figure 8.7: Circuits for the “scolded/refund” example pair. Left is the circuit of the “scolded” sentence, Right is the circuit created via the contrastive method

Chapter 9

Conclusion

This thesis has explored the domain of event semantics in natural language processing, focusing on various event structures and their interactions within discourse. By addressing the challenges of limited annotated data and complex event relationships, we have made progresses in understanding and modeling event semantics.

Key Learnings. Throughout the course of this thesis, we have gained significant insights into the complexities of event semantics and the methodologies required to effectively model and understand them. We outline the key learnings from this thesis, highlighting the strengths and limitation of the current research.

Supervised Structure Prediction: We developed algorithms to decode various event structures using expert-annotated datasets. These efforts revealed the rich properties and complex relations of event mentions, such as coreference and subevents. Supervised learning approaches were effective in structured prediction tasks like Event Detection, Event Coreference, Event Sequencing, and Ellipsis Resolution. However, the limited scope and scale of annotated datasets posed challenges to the generalizability of these models.

Scaling Up Data: This thesis shows promising evidence that scaling up the data can lead to more generalizable findings in semantics, where:

1. **Indirect Supervision**: Using summarization as a proxy task, we created a large-scale event salience dataset. This method demonstrated how indirect supervision signals can be harnessed to train robust models that capture intricate event-related phenomena. Additionally, we observed that scaling up the data allows the model to uncover other event structures.
2. **Crowdsourcing**: By breaking down complex annotation tasks into simpler steps, we successfully leveraged crowdsourcing to scale up data annotation. This approach not only increased the volume of annotated data but also led to the discovery of novel partial identity types, such as spatiotemporal continuity.
3. **Language Models**: Analyzing the performance of large language models on tasks like the Winograd Schema Challenge provided insights into their capabilities and limitations. Although LLMs exhibit significant advancements, they still have gaps in fully understanding and resolving complex event semantics. Moreover, it remains challenging for humans to interpret and analyze the diverse circuits identified by the algorithms.

Future Work. The research conducted in this thesis has laid a foundation for understanding and modeling event semantics in natural language processing. However, several areas deserve further exploration. In this section, we outline a few directions for future work that can build on the insights gained and address the limitations encountered in this thesis.

Enhanced Annotation Techniques: Further refine crowdsourcing workflows to handle more complex annotation tasks, potentially incorporating active learning to guide crowdworkers and improve annotation quality. Develop semi-automated annotation tools that leverage existing LLMs to assist human annotators, thereby increasing efficiency and accuracy.

Expanding Indirect Supervision: Explore additional proxy tasks and indirect supervision signals for other aspects of event semantics, such as event causality and temporal relations. Integrate multiple indirect supervision sources to create more comprehensive and diverse datasets, enhancing the robustness of event semantic models.

Interpretation Study of Language Models: Another crucial area for future work involves the interpretation and mechanism analysis of large language models (LLMs). Utilizing resources like the LLM360 project, further dissect the specific pathways and computations LLMs use to resolve complex event-related tasks. Expanding on the circuit analysis methods used in this thesis, develop frameworks for increased transparency and explainability to ensure that LLMs' decisions regarding event semantics can be understood and justified by end-users. Additionally, robust testing procedures and better evaluation metrics from this research can enhance LLMs' robustness in diverse real-world scenarios.

Cross-Linguistic and Cross-Domain Generalization: Extend the research to multiple languages and domains, assessing the generalizability of the developed models and approaches. This includes creating multilingual and cross-domain datasets for event semantics. Study the transferability of event semantic models across different linguistic and cultural contexts to build more universally applicable NLP systems.

Final Remarks. We hope that this thesis and the insights gained from the research pave the way for future advancements, bringing us closer to achieving a comprehensive and nuanced understanding of events in natural language. We encourage researchers, especially those in NLP and computational linguistics, to further explore how semantics can be represented by modern sophisticated models.

Appendices

Appendix A

Appendix for Chapter 7

A.1 Ethical Considerations

In our dataset construction, we follow the standard norms for ethical research involving human participants. We obtained IRB approval before starting our study. Our pilot study indicated that each HIT takes ∼10-15 minutes; therefore, we set the price of individual HIT to be $2.3. Overall, we paid a fair compensation of $10.9/hour (with median pay of $16.3/hour). For each HIT, the crowd workers on Mechanical Turk have signed the informed consent form before starting the task (see A.3 in Appendix). We provided clear instructions for using our annotation tool, both within and through an instructional video. We provide positive and negative examples to illustrate event coreference to the crowd workers (see A.2 in Appendix). Our dataset is limited to the English language, specifically for text documents relating to Disasters and accidents. While we have taken specific steps to improve the quality of our dataset, there might be incorrect or missing coreference links. However, we believe that such incorrect/missing links will not create additional risks to the models trained on our dataset.

A.2 Annotation Guidelines

To explain the task of cross-document event coreference to crowd workers on Mechanical Turk, we present detailed example-based guidelines (Table A.3, Table A.4). Additionally, we provide crowd workers with detailed instructions to our annotation interface (Table A.1, Table A.2). Workers view these instructions before the start of each task and optionally during the task. In our HIT, we also link to a 1-minute video tour of our annotation interface.

In our guidelines, we only present examples of full and null coreference. While we consider membership a form of coreference (partial), we don't train the crowd workers on full and partial identity.

A.3 MTurk Consent Form

A consent form is attached to the start of each HIT. Crowd workers are required to go through the form and provide their consent before starting the task. Anonymized version of the consent form is presented in Table A.5 and Table A.6. We anonymize the document for the conference review process.

A.4 MTurk Qualification Test

To identify high-quality crowd workers, we design a qualification test and add it as an additional requirement to solving our HITs.

A.4.1 Test Questions

In the qualification test on MTurk, we randomly select eight questions from a pool of 20 questions. Table A.7 and Table A.8 list all the questions.

A.4.2 Test Format

Table A.9 presents the format of the qualification test used for screening crowd workers.

A.5 HIT Template

Table A.10 presents our HIT layout. Our layout is simple, and all of our annotations are collected using our custom-designed annotation tool.

A.6 Follow-up Questions

Table A.11 lists the four follow-up questions. We present these questions for each coreference link annotated by the crowd worker.

Instructions for using the tool

This tool can be used to select events that are the same across the two given documents.

How to open instructions
(embedded GIF)
How to annotate one pair of events
(embedded GIF)
How to delete previous annotations
(embedded GIF)
How to proceed to the next event
(embedded GIF)

At any point during the task, you can click on the "View Instructions" button to read these instructions.

What is this task about?

- Two related documents are presented side-by-side on the tool.
- A few words in both the documents are underlined and these are referred to as events.
- The task is to select events from the right document that are the same as the currently highlighted event in the left document.

How should I solve this task?

- When you first start the task, make sure you read through both the left and right documents to get an overall understanding of the two documents.
- At each step, an event is highlighted in a blue box on the left document (aka. target event). Now, your goal is to identify underlined events from the right document that are the same as the target event from the left document.
- Once you select an event from the right document (an annotation), you are presented a few follow-up questions. Make sure you answer these questions to the best of your knowledge.
- If you change your mind while answering the questions, you can click the "Cancel" button to remove your annotation.
- After you have identified all possible same events from the right document (if any), please use the "Next event" button to move to the next target event on the left document.

Table A.1: Instructions as shown to the annotators on the interface.

Instructions for using the tool (contd.)

FAQs

Q: I made a mistake and incorrectly marked two events as the same. How do I correct this?
If you are still answering the follow-up questions, you can just click on the "Cancel" button. If you have already moved to the next target event, you can use the "Back" button to move back the previously finished target events.

Q: I am not sure how to respond to the follow-up questions. How should I proceed?
The follow-up questions help us understand more about your decision that two events are the same. It is important to note that the response to these questions need not always be "Yes". In fact, in many cases, you may not have enough information to respond with a definite "Yes" or "No", then please feel free to select "Not enough information".

Q: How do I decide if two events are the same or different?
We understand that this decision is not always easy. To help you with this, we compiled a bunch of examples. You can quickly glance through them using the "View Examples" button on the tool.

Q: How do I contact the authors of the task?
For any comments, feedback and/or suggestions, please use this form (XXXX). We strive to make this a great experience for you.

Table A.2: Instructions as shown to the annotators on the interface. (contd)

Examples

Goal of the Task
You will help us identify the same events from different documents.
What is an event?
People use text to describe what happen(ed) in the world. These are called events in text. We often use verbs, sometimes even (pro)nouns, and adjectives as events. For example:
- It rained a lot yesterday.
- There was a fire last night.
- He got sick.

How do we know that the two events are the same?
In the following examples (1 to 5), two events are the same.

1. When two events refer to the same thing, they should be the same in terms of meaning, or semantically identical.
   - Taken as a whole, the evidence suggests that the plan to bomb the Boston Marathon took shape over three months.
   - Dzhokhar Tsarnaev apologized for suffering caused by the Boston Marathon bombing.

2. When two events are the same, one event may be the synonym for the other.
   - A 16-year-old southern Utah boy was accused of bringing a homemade bomb to his high school.
   - The teen was charged Monday with attempted murder and use of a weapon of mass desctuction, both first-degreen felonies.

3. Sometimes one event may be the pronoun (e.g.,it) or the anaphora (e.g., this, that) of the other, when they are the same.
   - Both drones carried explosives, and no YPF ("People's Defence Units") fighters were injured in the incident.
   - This would not be the first terrorist drone strike.

4. The same events do not have to take place at the same time. In the following example, one event ("go") would happen in the future, while the other ("went") did occur.
   - The couple had been planning to go to Paris for a long time.
   - They finally went there last month.

5. Sometimes the same events are described from different perspectives. The following example refers to the exchange of the gift from two perspectives.
   - John gave a gift to Mary.
   - Mary received a gift from John.

Table A.3: Examples for coreference and non-coreference, as shown to the annotators on the interface.

Examples (contd.)

In the following examples (6 to 8), two events are not the same.

6. When one event is a part of the other larger event, they are not the same.

- Following the trial of Mahammed Alameh, the first suspect in the bombing, investigators discovered a jumble of chemicals, chemistry implements and detonating materials.
- The explosion killed at least five people. ("bombing" refers to the entire process which starts with making a bomb and ends with destructions, damages and injuries, while "explosion" is a smaller event that occurs in that processes)

7. Two events are not the same even if they are the same semantically. The first example refers to the general bomb-making process, while the second one indicates a particular bomb-making event that took place in the garage.

- They obtained the online manual of bomb-making. (general bomb-making process)
- They made a bomb in the garage. (specific bomb-making event that happened in the specific place)

8. When one event consists of, or is a member of the other event, they are not the same. The first example refers to the specific death of a 44-year-old man, while the second one refers to the deaths of 305 people.

- The government announced that a 44-year-old man died from the COVID. (death of a 44-year-old man)
- There are more than 14,300 confirmed COVID cases, and 305 people have died. (deaths of 305 people)

Table A.4: Examples for coreference and non-coreference, as shown to the annotators on the interface. (contd)

Consent From

This task is part of a research study conducted by XXX at XXX and is funded by XXX.

Purpose
The goal of this study is to collect datasets of coreference-labeled pairs sampled from public online news articles through the help of crowd workers.

Procedures
You will be directed to a website implemented by the research team to complete the task. You will be asked to read upto 3 pairs of articles. For each pair of articles, you will need to label pieces of text that refer to the same event, and answer additional questions about your labeling. Labeling one pair of articles whose length sums up to 40 sentences is expected to take around 15 minutes.

Participant Requirements
Participation in this study is limited to individuals age 18 and older, and native English speakers.

Risks
The risks and discomfort associated with participation in this study are no greater than those ordinarily encountered in daily life or during other online activities.

Benefits
There may be no personal benefit from your participation in the study but the knowledge received may be of value to humanity.

Compensation & Costs
For this task, you will receive between $2 to $3 for annotating each pair of articles. The exact reward for each pair depends on the length of corresponding articles. You will not be compensated if you provide annotations of poor quality.
There will be no cost to you if you participate in this study.

Future Use of Information and/or Bio-Specimens
In the future, once we have removed all identifiable information from your data (information or bio-specimens), we may use the data for our future research studies, or we may distribute the data to other researchers for their research studies. We would do this without getting additional informed consent from you (or your legally authorized representative). Sharing of data with other researchers will only be done in such a manner that you will not be identified.

Confidentiality
The data captured for the research does not include any personally identifiable information about you except your IP address and Mechanical Turk worker ID.
By participating in this research, you understand and agree that XXX may be required to disclose your consent form, data and other personally identifiable information as required by law, regulation, subpoena or court order. Otherwise, your confidentiality will be maintained in the following manner:

Table A.5: Consent Form attached to each of our HITs. We anonymize the document for the conference review process.

Consent From (contd.)

Confidentiality (contd.)
Your data and consent form will be kept separate. Your consent form will be stored in a secure location on XXX property and will not be disclosed to third parties. By participating, you understand and agree that the data and information gathered during this study may be used by XXX and published and/or disclosed by XXX to others outside of XXX. However, your name, address, contact information and other direct personal identifiers will not be mentioned in any such publication or dissemination of the research data and/or results by XXX. Note that per regulation all research data must be kept for a minimum of 3 years.
The Federal government offices that oversee the protection of human subjects in research will have access to research records to ensure protection of research subjects.

Right to Ask Questions & Contact Information
If you have any questions about this study, you should feel free to ask them by contacting the Principal Investigator now at XXX, XXX, or by phone at XXX, or via email at XXX. If you have questions later, desire additional information, or wish to withdraw your participation please contact the Principal Investigator by mail, phone or e-mail in accordance with the contact information listed above.
If you have questions pertaining to your rights as a research participant; or to report concerns to this study, you should contact the XXX at XXX. Email: XXX. Phone: XXX or XXX.

Voluntary Participation
Your participation in this research is voluntary. You may discontinue participation at any time during the research activity. You may print a copy of this consent form for your records.

I am age 18 or older. □ Yes □ No

I have read and understand the information above. □ Yes □ No

I want to participate in this research and continue with the task. □ Yes □ No

Table A.6: Consent Form attached to each of our HITs. We anonymize the document for the conference review process. (contd)

#	Text	Answer	Type
1	A 500lb bomb packed in the Cavalier is detonated with a remote trigger. The explosion tears through Market Street.	yes	Synonym
2	The death toll of the Omagh bomb blast in Northern Ireland has risen to 29 following the death of a man in hospital.	no	Member
3	Ahmed al-Mughassil was arrested in Beirut and transferred to Riyadh, the Saudi capital, according to the Saudi newspaper Asharq Alawsat. The Saudi Interior Ministry and Lebanese authorities had no immediate comment on the capture.	yes	Synonym
4	The blast didn't cause the destruction its planners intended. But it opened up a multi-story crater in the building, injured more than 1,000 people and ultimately killed six.	no	Member
5	March 4, 1998 - Four defendants, Salameh, Ayyad, Abouhalima, and Ajaj, are convicted. They are sentenced to prison terms of 240 years each. In 1998, the sentences were vacated. In 1999, the men were re-sentenced to terms of more than 100 years.	no	Unrelated
6	Perhaps the only early clues to emerge on an early quiet second day of the Boston Marathon bombing investigation - from the ATF and the FBI and the Boston police, from anonymous law enforcement officials and doctors pulling ball bearings out of victims limbs - concern the Boston bombs themselves. A similar scene played out in the Boston suburb of Newton, where a bomb used a robot to investigate a suspicious object that turned out to be a circuit board.	no	Member
7	As of Tuesday morning, jurors began reviewing evidence and witness testimony, which will play a role in helping them divide Dzhokhar Tsarnaev's guilt on each of the 30 charges he faces. A key issue for jurors - both in the guilt phase and later the penalty phase if Tsarnaev is convicted - will be whether the jurors see Tsarnaev as an equal partner with his old brother, Tamerian Tsarnaev, in the Boston Marathon bombing and the violent events that followed.	yes	Synonym
8	Though building the bomb was relatively easy, the experts say, it was not by any means free of danger. The bulkiest part of the bomb, they say, was extremely stable and could only have been touched off with a tremendous kick, like that provided by nitroglycerine. Making the nitroglycerin, blending some of the chemicals, was the trickiest part of the process.	yes	Synonym
9	An ongoing Somali offensive, backed by the U.S. and an African Union peacekeeping force has recaptured territory from al Shabaab in south-central Somalia, but has not eliminated al Shabaab's ability to conduct VBIED attacks. U.S.-backed Somali ground operations along with improved counter-VBIED capabilities among Somali forces may have slightly decreased VBIED attacks between November 2017 and January 2018.	yes	Synonym
10	According to the United Nations, more than 2.3 million Venezuelans have left their country in recent years. Increasingly they are leaving with no money and are traveling on foot across South American countries like Colombia, Ecuador and Peru, in dangerous journeys that can take several weeks.	no	Member

Table A.7: Examples used with the qualification test on Mechanical Turk. For each paragraph with two highlighted events, we ask the question, "In the above paragraph, are the highlighted events the same?". The crowd worker has to select one of the "Yes" or "No" options.

#	Text	Answer	Type
11	Spain's King Juan Coarlos and Queen Sofia traveled to their summer residence in Majorca Saturday just two days after a bombing blamed on Basuqe separatists killed two policemen on the resort island.	no	Member
12	Yahoo Inc. is preparing to lay off between 600 and 700 workers in the latest shakeup triggered by the Internet company lackluster growth. Employees could be notified of the job cuts as early as Tuesday, according to a person familiar with Yahoo's plans.	yes	Synonym
13	A man shot and killed by police officers during a burglary here early Monday was identified by law enforcement authorities as the suspect in a string of five shooting deaths in South Carolina over the last 10 days. Sheriff Bill Blanton of Cherokee Country, S.C., where the killings took place, confirmed Monday evening that the authorities had been seeking the man killed in the burglary, Patrick T. Burris, a felon with a long record who had served seven years in prison and was paroled in April.	yes	Synonym
14	Staff Sgt. Robert Bales offered a tearful apology Thursday for gunning down 16 unarmed Afghan civilians inside their homes but said he still could not explain why he had carried out one of the worst U.S. war crimes in years. The unsworn statement from Bales, 40, came on the third day of hearing to determine whether he should ever be eligible for parole in the March 2012 Massacre.	yes	Synonym
15	In January two men were hanged after being convicted of involvement in protests, and in May, four Iranian Kurds and another man accused of terrorism were executed.	no	Unrelated
16	The Dow Corning Corporation filed for bankruptcy protection in a Federal court in Bay City, Michigan. Dow Corning said that seeking the protection of the bankruptcy court was the only way it could devise an enforceable plan to deal with the claims against it.	no	Realis
17	The UN report accused both Israel and Palestinian armed groups of commiting war crimes during the three-week war in Gaza that erupted on December 27, killing some 1,400 Palestinians and 13 Israelis.	no	Realis
18	A judge has ordered the surviving children of the Rev. Martin Luther King Jr. and Coretta Scott King to hold a shareholder's meeting to discuss their father's estate. The three siblings are the sole shareholders, directors and officers of a company that manages their father's intellectual property, but they have not met for an annual shareholder's meeting since 2004.	no	Realis
19	The first attack was a failure, but if the report is accurate, then it signals a dangerous new terror threat. The report showed pictures of the remains of a homemade attack drone.	no	Realis
20	A key issue for jurors - both in the guilt phase and later in the penalty phase if Tsarnaev is convicted - will be whether the jurors see Tsarnaev as an equal partner with his older brother, Tamerlan Tsarnaev, in the Boston Marathon bombing and the violent events that followed. Taken as a whole, the evidence suggests that the plan to bomb the Boston Marathon took shape over three months.	yes	Realis

Table A.8: Examples used with the qualification test on Mechanical Turk. For each paragraph with two highlighted events, we ask the question, "In the above paragraph, are the highlighted events the same?". The crowd worker has to select one of the "Yes" or "No" options. (contd)

Screening Test

In this test, we ask you to identify whether two events (**highlighted** in each paragraph) indicate the same thing or not. Read each paragraph carefully and answer the question by selecting the appropriate option, *Yes* or *No*.

In total, you are presented with 8 questions and the time limit for this test is 20 minutes.

Note: It is important you do this test on your own because our HITs are similar to the questions presented in this test. For your reference, we provide five examples below,

He **died** of injuries from the accident. His friends were all saddened to hear his **death**.
Question: In the above paragraph, are the highlighted events the same?
Answer: Yes (both words, **died** and **death** indicate the person's death)

The suspect was **shot** and killed in the **raid** by the armed officers.
Question: In the above paragraph, are the highlighted events the same?
Answer: No (**shot** happened during the **raid**)

The couple had been planning to **go** to Paris for a long time. They finally **went** there last month.
Question: In the above paragraph, are the highlighted events the same?
Answer: Yes (The two events do not have to take place at the same time. Here, **go** would happen in the future, and **went** did occur.)

John **gave** a gift to Mary. Mary **received** a gift from John.
Question: In the above paragraph, are the highlighted events the same?
Answer: Yes (Same events described from different perspectives.)

Following the trial of Mahammed Alameh, the first suspect in the **bombing**, investigators discovered a jumble of chemicals, chemistry implements and detonating materials. The **explosion** killed at least five people.
Question: In the above paragraph, are the highlighted events the same?
Answer: No (One event is part of the other larger event. **bombing** refers to the entire process which starts with making a bomb and ends with destructions, damages and injuries, while **explosion** is a smaller event that occurs in that processes.)

Q1. ....
☐ Yes ☐ No
Q2. ....
☐ Yes ☐ No
...

Table A.9: The template used in the qualification test to screen annotators. In addition to instructions and examples, we present eight yes/no questions.

Annotating Event Coreference in News Articles

In this HIT, you will be using our tool to perform the task. For a short tutorial on using our interface, see this 1 minute video: XXX. This HIT contains the following two steps,

- Visit the URL provided below to perform the task.
- At the end of the task, you will be provided a secret code. To submit this HIT, copy the secret code and paste it into the box provided below. Note that the secret code is unique for each task.

Link to the task: XXX

Fill in the secret code

Paste the secret code provided at the end of the task into the text box (*required)

Table A.10: The template used for each Human Intelligence Task (HIT) on Mechanical Turk.

Place: Do you think the two events happen at the same place?

☐ Exactly the same ☐ The places overlap ☐ Not at all ☐ Cannot determine

Time: Do you think the two events happen at the same time?

☐ Exactly the same ☐ They overlap in time ☐ Not at all ☐ Cannot determine

Participants: Do you think the two events have the same participants?

☐ Exactly the same ☐ They share some participants ☐ Not at all ☐ Cannot determine

Inclusion: Do you think one of the events is part of the other?

☐ Yes, the left event is part of right one ☐ Yes, the right event is part of left one
☐ No, they are exactly the same ☐ Cannot determine

Table A.11: Follow-up questions used for each annotated coreference link.

Appendix B

Appendix for Chapter 8

B.1 LLM360 Details

The recent emergence of accessible, highly capable LLMs such as LLaMA [196, 197], Falcon [162], and Mistral [107] has allowed researchers to easily obtain, customize, and deploy LLMs for diverse applications. However, many open-source LLMs restrict visibility and access to their training processes, limiting the broader AI research community's ability to study and innovate upon these models. To address this, we develop the LLM360 project, emphasizing open research by releasing intermediate checkpoints, training details, and artifacts.

Model Configurations. In this thesis, we used a 7B parameter model named AMBER trained on approximately 1.26 trillion tokens (see Table B.2 for the dataset breakdown). The model shares architectural similarities with LLaMA1 7B, including model dimensions, use of RMSNorm [221], and rotary positional embeddings (RoPE) at each layer of the network [191], a summary of model specification is in Table B.2.

Model Checkpoints. LLM360 models are released with all intermediate checkpoints saved during training, including model weights and optimizer states. These checkpoints enable continued training from various starting points and facilitate post-training research.

Pre-training Infrastructure. AMBER is trained on a GPU cluster of 56 DGX A100 nodes, each equipped with $4\times$ 80GB A100 GPUs. Each GPU is connected with 4 links NVLink. Cross node connection setting is 2 port 200 Gb/sec ($4\times$ HDR) InfiniBand. The throughput we achieved with our distributed training framework is

AMBER	
Subset	Tokens
Arxiv	30B
Book	29B
C4	198B
Refined-Web	665B
StarCoder	292B
StackExchange	22B
Wikipedia	24B
Total	1.26T

Table B.1: Data mixture in AMBER.

Number Parameters	$6.7B$	Activation	SwiGLU
Hidden Size	4096	Sequence Length	2048
Intermediate Size (in MLPs)	11008	Vocabulary Size	32000
Number of Attention Heads	32	Position Embedding	Rotary
Number of Hidden Layers	32	QK Dot Product Scaling	$\frac{QK^T}{\sqrt{d}}$
LR Schedule	Cosine Decay	Warmup Steps	2000
Normalization	RMSNorm	Batch Size	2240

Table B.2: Model architecture for AMBER

around $582.4k$ tokens per second. Our pretraining framework is lit-llama[1] developed based on PyTorch Lightning. We used mixed-precision during pre-training with BF16 for activations and gradients and FP32 for model weights [140].

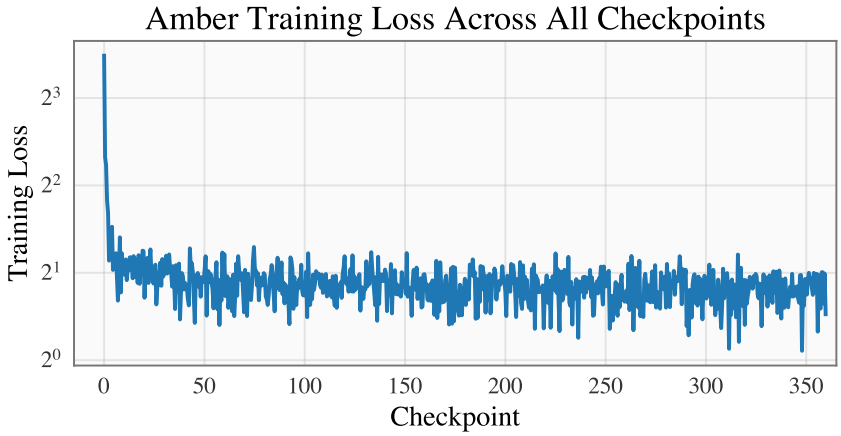

Figure B.1: The training loss curves of AMBER

Metrics. The LLM360 project also provides access to detailed logs and intermediate metrics, including system statistics, training logs, and evaluation metrics.

B.2 More Circuit Graphs

[1] https://github.com/Lightning-AI/lit-llama

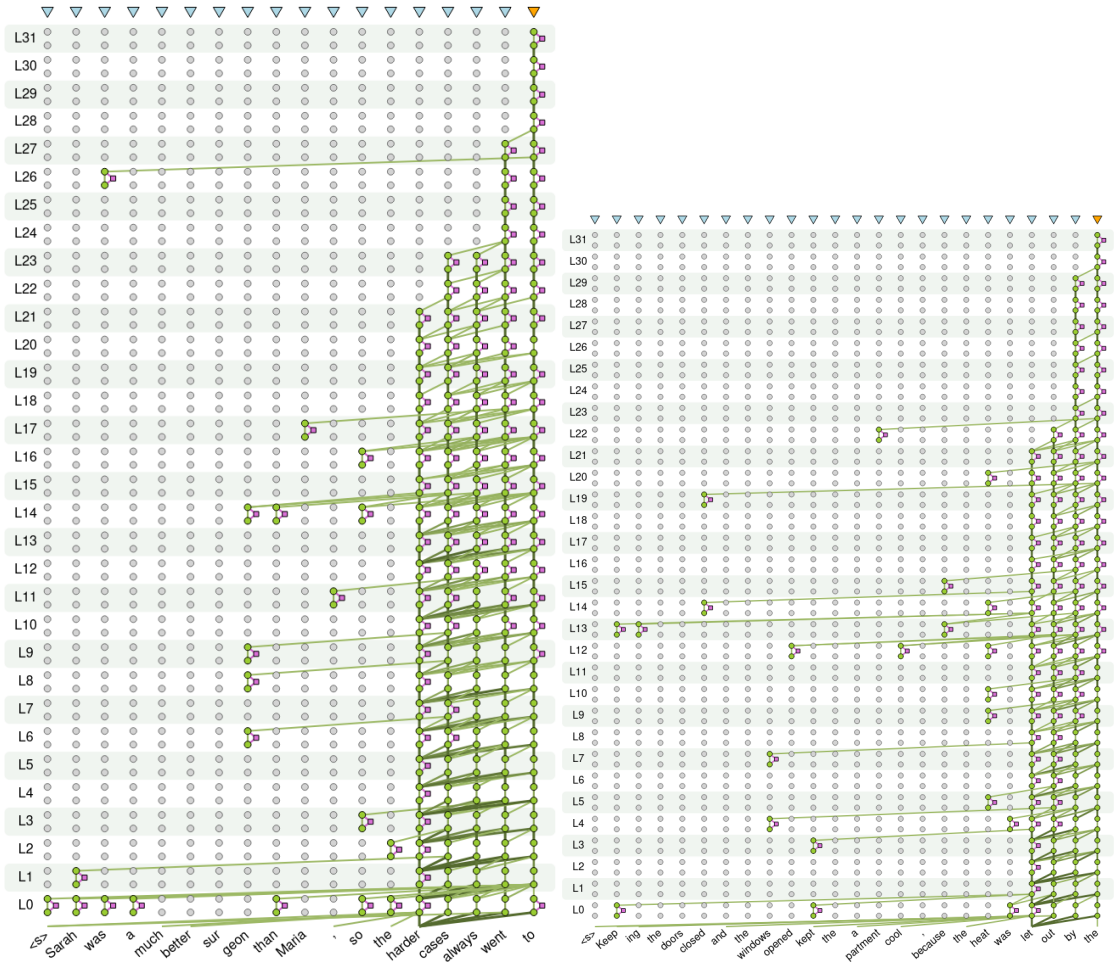

Figure B.2: Contrastive Circuit Graphs for two Winograd pairs. Left: Sarah was a much better surgeon than Maria, so the (harder/easier) cases always went to (Sarah/Maria). Right: Keeping the doors closed and the windows opened kept the apartment cool, because the heat was (kept/let out) by the (door/window)

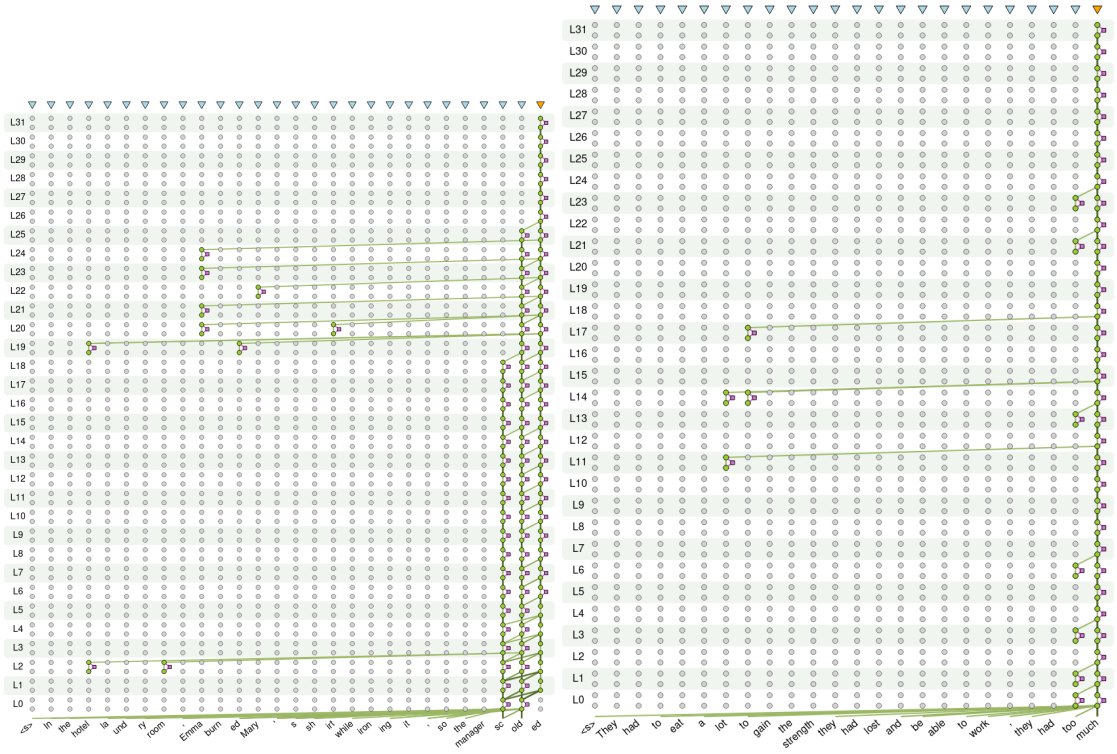

Figure B.3: Contrastive Circuit Graphs for two Winograd pairs. Left: In the hotel laundry room, Emma burned Mary's shirt while ironing it, so the manager (scolded/refunded) (Emma/Mary). Right: They had to eat a lot to gain the strength they had lost and be able to work, they had too (much/little) (work/strength)

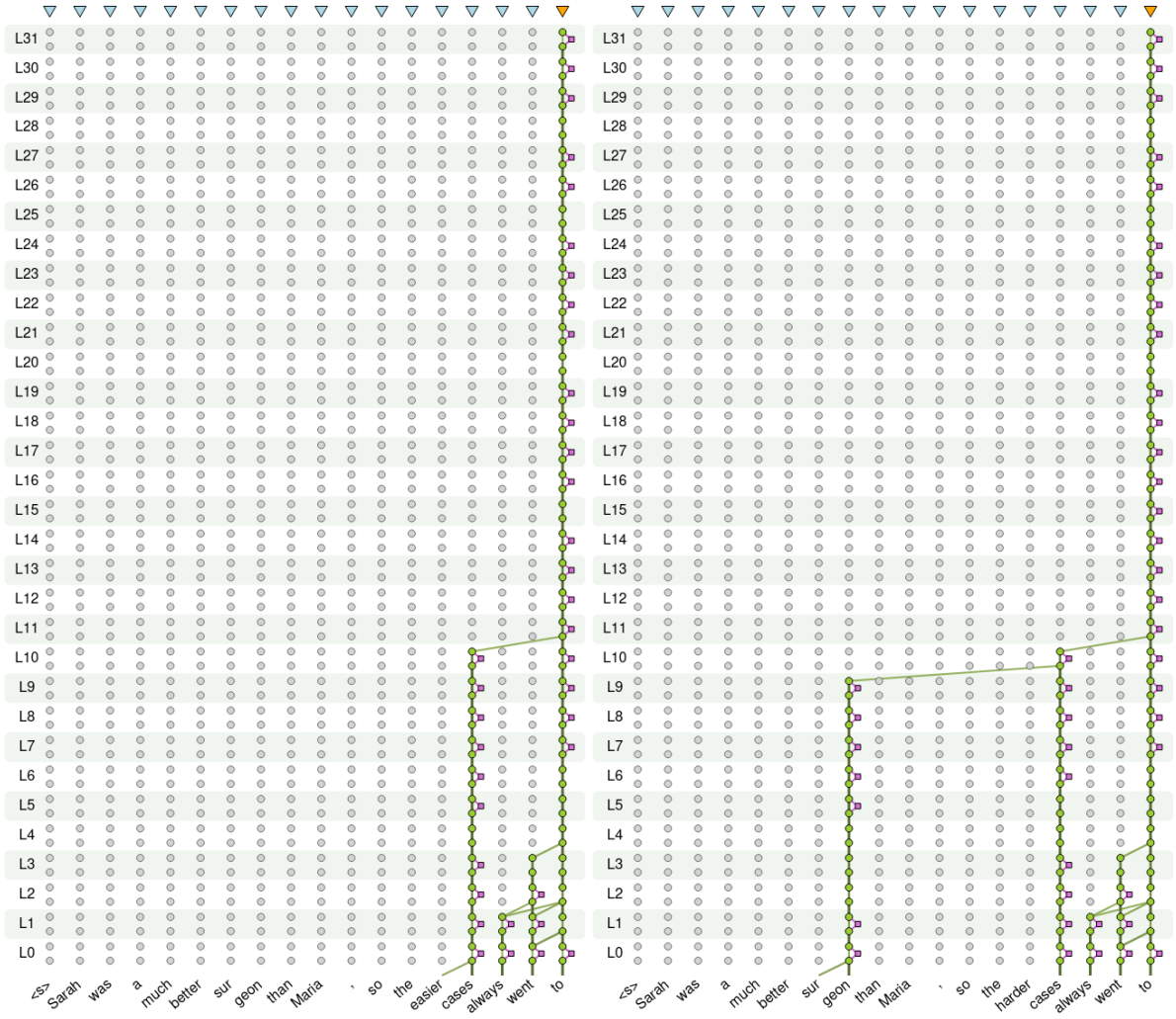

Figure B.4: Base IFR Circuit for the Winograd pair: Sarah was a much better surgeon than Maria, so the (harder/easier) cases always went to (Sarah/Maria).

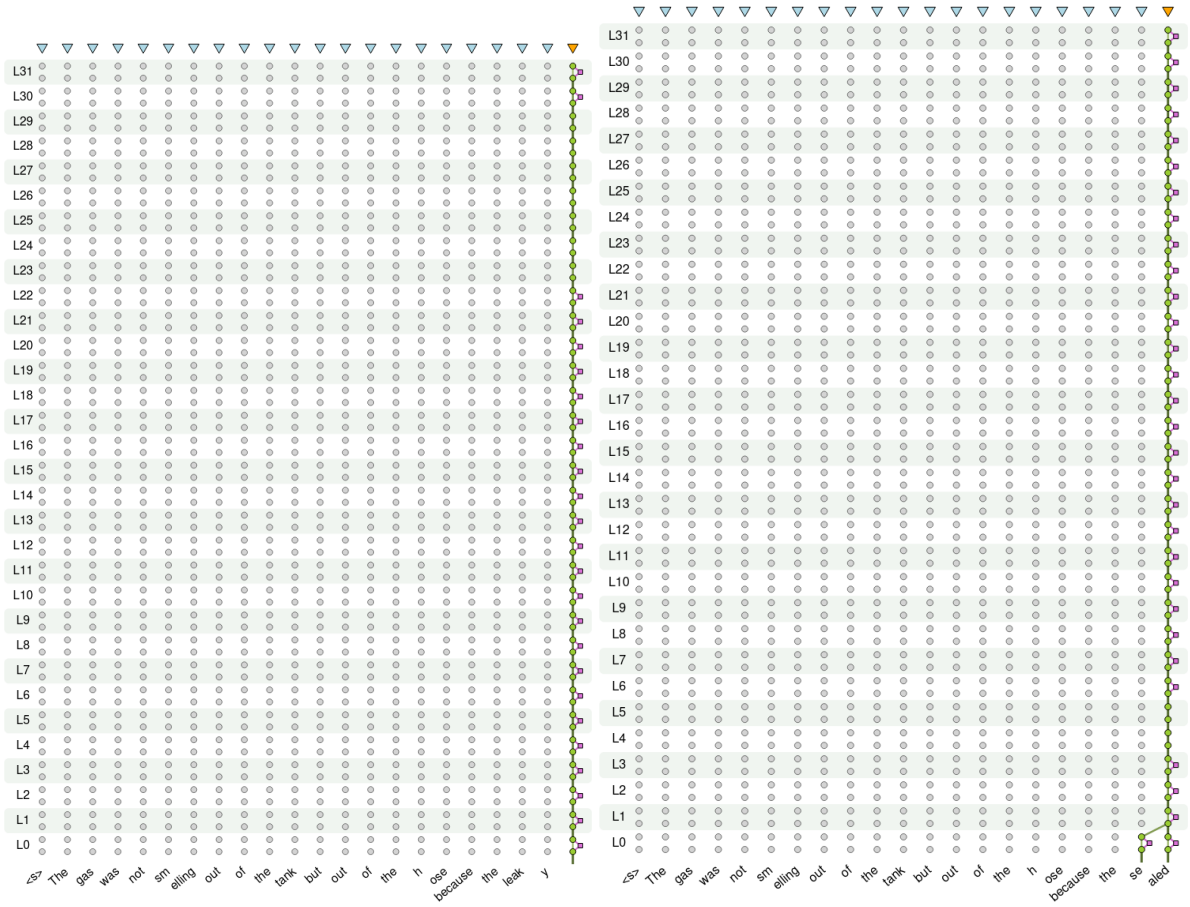

Figure B.5: Base IFR Circuit for the Winograd pair: The gas was not smelling out of the tank but out of the hose because the (leaky/sealed) (tank/hose).

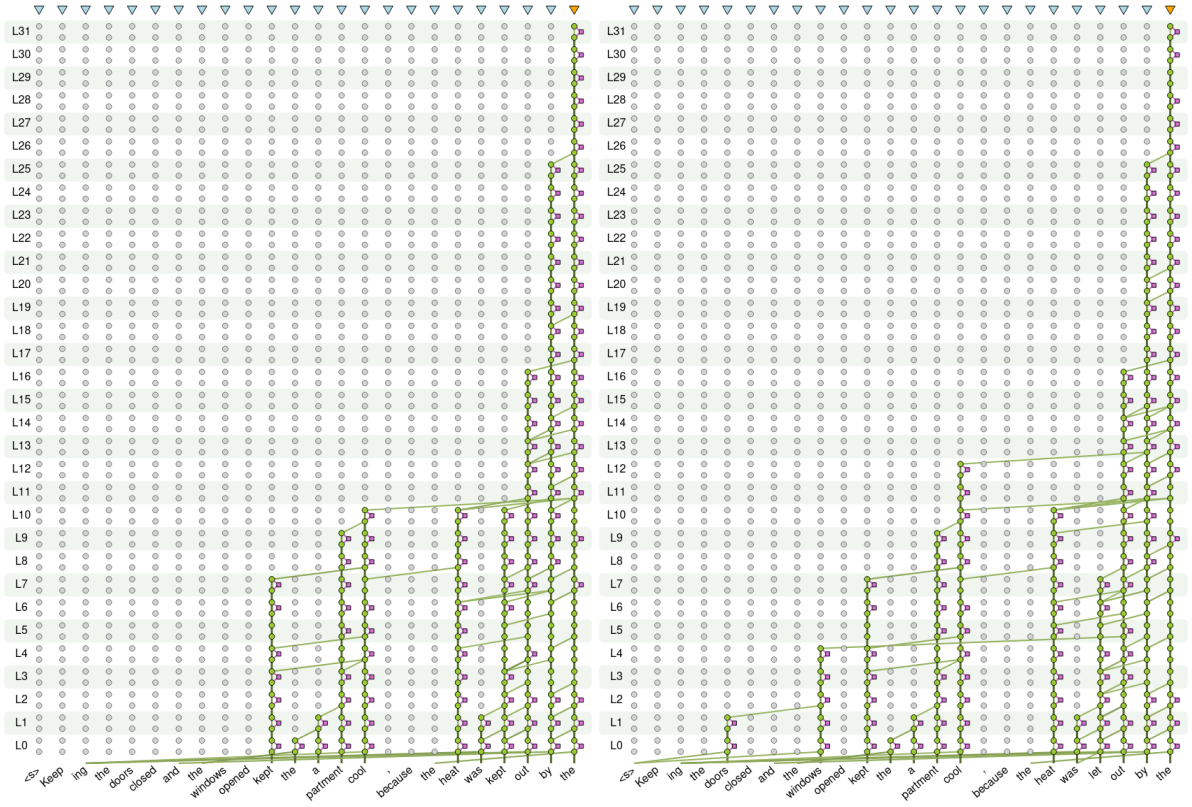

Figure B.6: Base IFR Circuit for the Winograd pair: Keeping the doors closed and the windows opened kept the apartment cool, because the heat was (let out/kept) by the (doors/windows).

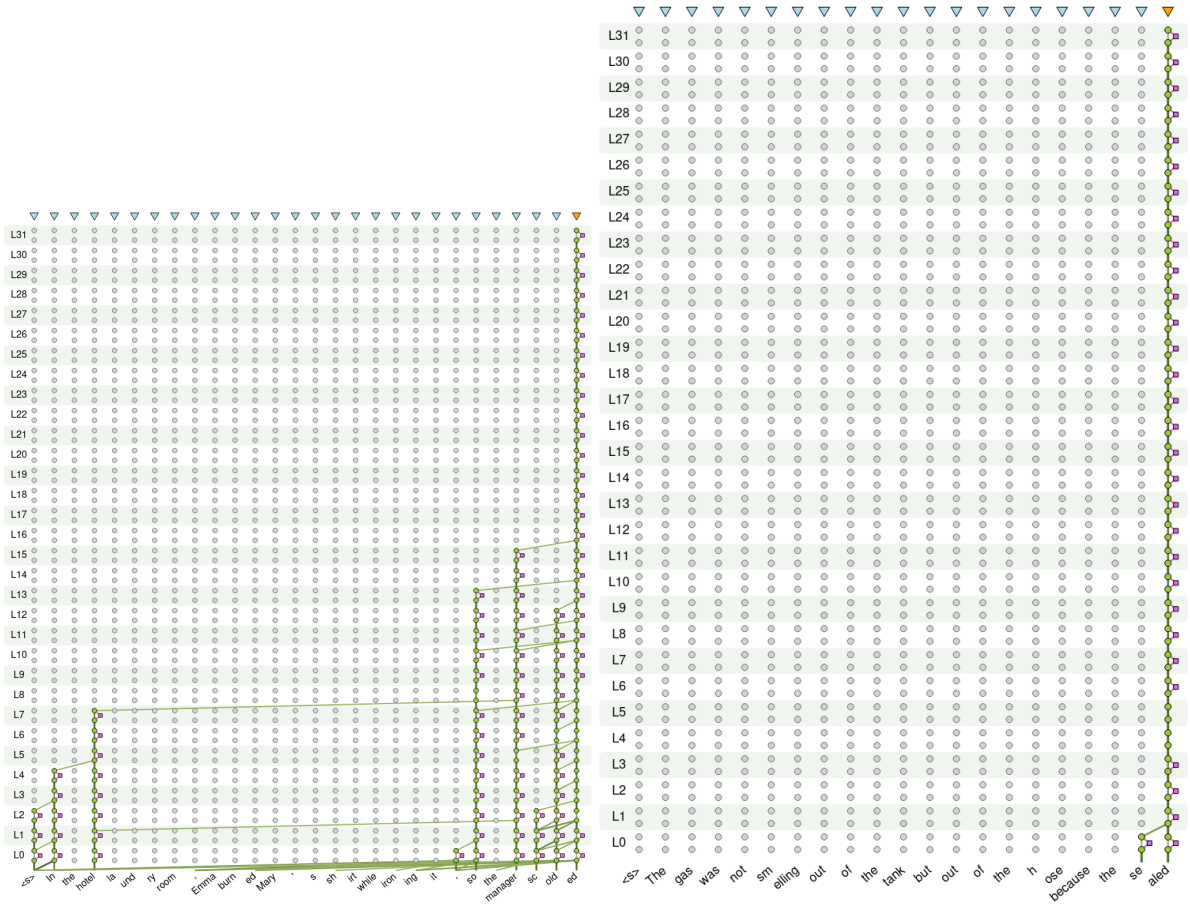

Figure B.7: Base IFR Circuit for the Winograd pair: In the hotel laundry room, Emma burned Mary's shirt while ironing it, so the manager (refunded/scolded) (Emma/Mary).

Figure B.8: Base IFR Circuit for the Winograd pair: They had to eat a lot to gain the strength they had lost and be able to work, they had too (much/little) (work/strength)

Bibliography

[1] 2016. The 4th Workshop on Events : Definition , Detection , Coreference , and Representation Proceedings of the Workshop. In *NAACL 2016*.

[2] David Ahn. 2006. The stages of event extraction. *ARTE '06 Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8. 21, 22, 24, 27

[3] AI@Meta. 2024. Llama 3 model card. 8

[4] Jun Araki. 2018. *Extraction of Event Structures from Text*. Ph.D. thesis, Carnegie Mellon University. 11

[5] Jun Araki, Zhengzhong Liu, Eduard Hovy, and Teruko Mitamura. 2014. Detecting Subevent Structure for Event Coreference Resolution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4553–4558, Reykjavik, Iceland. European Language Resources Association (ELRA). 33, 73

[6] Jun Araki and Teruko Mitamura. 2018. Open-domain event detection using distant supervision. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 878–891, Santa Fe, New Mexico, USA. Association for Computational Linguistics. 85, 89

[7] Ron Artstein and Massimo Poesio. 2008. Survey article: Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596. 92

[8] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: a nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, page 722–735. Springer-Verlag, Berlin, Heidelberg. 26, 29

[9] Emmon Bach. 1986. The algebra of events. *Linguistics and Philosophy*, 9(1):5–16. 11

[10] A. Bagga and B. Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 79–85. Association for Computational Linguistics. 26, 39

[11] Amit Bagga and Bagga Baldwin. 1999. Cross - Document Event Coreference: Annotations, Experiments, and Observations. In *ACL-99 Workshop on Coreference and Its Applications*. 22, 23

[12] CF Baker, CJ Fillmore, and JB Lowe. 1998. The berkeley framenet project. *Proceeding*

ACL '98 Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, pages 86–90. 72, 73

[13] Niranjan Balasubramanian, Stephen Soderland, and OE Mausam. 2013. Generating Coherent Event Schemas at Scale. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 73

[14] Breck Baldwin and Thomas S Morton. 1998. Dynamic Coreference-Based Summarization. *Proceedings of the Third Conference on Empirical Methods in Natural Language Processing (EMNLP-3)*, pages 1–6. 73

[15] Regina Barzilay and Mirella Lapata. 2008. Modeling Local Coherence: An Entity-Based Approach. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 1–34. 3, 71, 75, 78

[16] Cosmin Adrian Bejan and Sanda Harabagiu. 2008. A linguistic resource for discovering event structures and resolving event coreference. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'10)*, pages 2881–2887. 21, 87

[17] Cosmin Adrian Bejan and Sanda Harabagiu. 2010. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for the Association for Computational Linguistics*, July, pages 1412–1422. 21, 22, 23, 24, 27

[18] Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raf. 2023. Emergent and predictable memorization in large language models. *arXiv preprint arXiv:2304.11158*. 101

[19] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR. 103

[20] Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. 2023. Language models can explain neurons in language models. 101, 109

[21] Anders Björkelund and Jonas Kuhn. 2014. Learning Structured Perceptrons for Coreference Resolution with Latent Antecedents and Non-local Features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 47–57. 32, 33, 36

[22] Johan Bos and Jennifer Spenader. 2011. An annotated corpus for the analysis of VP ellipsis. *Language Resources and Evaluation*, 45(4):463–494. 47, 48, 49, 53

[23] L Breiman. 2001. Random forests. *Machine learning*, pages 5–32. 25

[24] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan,

and Chris Olah. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. 101

[25] Susan Windisch Brown, Claire Bonial, Leo Obrst, and Martha Palmer. 2017. The Rich Event Ontology. In *Proceedings of the Events and Stories in the News Workshop*, pages 87–97. 73

[26] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. 7

[27] Michael Bugert, Nils Reimers, Shany Barhom, Ido Dagan, and Iryna Gurevych. 2020. Breaking the subtopic barrier in cross-document event coreference resolution. In *Proceedings of Text2Story - Third Workshop on Narrative Extraction From Texts co-located with 42nd European Conference on Information Retrieval, Text2Story@ECIR 2020, Lisbon, Portugal, April 14th, 2020 [online only]*, volume 2593 of *CEUR Workshop Proceedings*, pages 23–29. CEUR-WS.org. 87

[28] Taylor Cassidy, Bill McDowell, Nathanael Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 501–506, Baltimore, Maryland. Association for Computational Linguistics. 85

[29] Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan. 2021. Realistic evaluation principles for cross-document coreference resolution. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 143–151, Online. Association for Computational Linguistics. 95

[30] Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense Event Ordering with a Multi-Pass Architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284. 33, 38, 73

[31] Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised Learning of Narrative Event Chains. In *ACL '08 Meeting of the Association for Computational Linguistics*, pages 789–797. 33, 41, 42, 73, 78

[32] Nathanael Chambers and Dan Jurafsky. 2010. A Database of Narrative Schemas. In *LREC 2010, Seventh International Conference on Language Resources and Evaluation*. 37

[33] Jonathan Chang, Sean Gerrish, Chong Wang, and David M Blei. 2009. Reading Tea Leaves: How Humans Interpret Topic Models. In *Advances in Neural Information Processing Systems 22*, pages 288—-296. 79

[34] Tyler A. Chang and Benjamin K. Bergen. 2021. Word acquisition in neural language models. 101

[35] Angelica Chen, Ravid Shwartz-Ziv, Kyunghyun Cho, Matthew L Leavitt, and Naomi Saphra. 2024. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in MLMs. In *The Twelfth International Conference on Learning Representations*. 101

[36] Chen Chen and Vincent Ng. 2013. Chinese Event Coreference Resolution: Understanding

the State of the Art. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, October, pages 822–828. 33

[37] Chen Chen and Vincent Ng. 2015. Chinese Event Coreference Resolution : An Unsupervised Probabilistic Model Rivaling Supervised Resolvers. In *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*, pages 1097–1107. 33

[38] Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176, Beijing, China. Association for Computational Linguistics. 48

[39] Zheng Chen and H Ji. 2009. Graph-based event coreference resolution. *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, pages 54–57. 22, 24, 33

[40] Zheng Chen, H Ji, and Robert Haralick. 2009. A pairwise event coreference model, feature impact and evaluation for event coreference resolution. In *Proceedings of the Workshop on Events in Emerging Text Types*, 3, pages 17–22. 24, 25, 33

[41] Pengxiang Cheng and Katrin Erk. 2018. Implicit Argument Prediction with Event Knowledge. In *NAACL 2018*, 2012. 3, 4, 71

[42] JC Kit Cheung, H Poon, and Lucy Vanderwende. 2013. Probabilistic Frame Induction. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL/HLT 2013)*. 33, 42, 82

[43] Nancy Chinchor. 1992. MUC-5 EVALUATION METRIC. In *Proceedings of the 5th Conference on Message Understanding*, pages 69–78. 26, 39

[44] Timothy Chklovski and Patrick Pantel. 2004. VERB OCEAN : Mining the Web for Fine-Grained Semantic Verb Relations. In *In Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*, pages 33–40. 29

[45] Prafulla Kumar Choubey, Kaushik Raju, and Ruihong Huang. 2018. Identifying the Most Dominant Event in a News Article by Mining Event Coreference Relations. In *NAACL 2018*. 71, 73, 82

[46] Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46. 75

[47] Michael Collins. 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proceedings of the Conference on Empirical Methods in NLP (EMNLP 2002)*, July, pages 1–8. 14, 36

[48] Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (almost) from Scratch. *Journal of Machine Learning Research*. 29

[49] Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià

Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. 8, 101

[50] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online Passive-Aggressive Algorithms. *Journal of Machine Learning Research*, 7:551–585. 35

[51] Agata Cybulska and Piek Vossen. 2012. Using Semantic Relations to Solve Event Coreference in Text. In *SemRel2012 in conjunction with LREC2012*, pages 60–67. 21, 22, 24, 27

[52] Agata Cybulska and Piek Vossen. 2014. Using a sledgehammer to crack a nut? Lexical diversity and event coreference resolution. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4545–4552. 87

[53] Mary Dalrymple, Stuart M. Shieber, and Fernando C N Pereira. 1991. Ellipsis and higher-order unification. *Linguistics and Philosophy*, 14(4):399–452. 47

[54] Laurence Danlos. 2003. Event coreference between two sentences. *Computing Meaning*, 77:271–288. 23

[55] D Das, Nathan Schneider, Desai Chen, and NA Smith. 2010. Probabilistic frame-semantic parsing. In *In Proceedings of the North American Chapter of the Association for Computational Linguistics Human Language Technologies Conference (NAACL 2010)*, volume 3, Los Angeles. 29

[56] Dipanjan Das and NA Smith. 2011. Semi-Supervised Frame-Semantic Parsing for Unknown Predicates. In *HLT '11 Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, volume 1, pages 1435–1444. xiii, 15, 37, 74

[57] Ofer Dekel, C Manning, and Yoram Singer. 2004. Log-linear models for label ranking. *Advances in neural information processing systems*, 16:497–504. 49

[58] Quang Xuan Do, Yee Seng Chan, and Dan Roth. 2011. Minimally supervised event causality identification. In *EMNLP '11 Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 294–303. 73, 78

[59] Quang Xuan Do, Wei Lu, and Dan Roth. 2012. Joint Inference for Event Timeline Construction. *EMNLP-CoNLL '12 Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 677–687. 33, 73

[60] M. Dojchinovski, D. Reddy, T. Kliegr, T. Vitvar, and H. Sack. 2016. Crowdsourced corpus with entity salience annotations. In *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC*, pages 3307–3311. 73

[61] Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations*. 60

[62] Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia.

Association for Computational Linguistics. 60

[63] Jesse Dunietz and Daniel Gillick. 2014. A New Entity Salience Task with Millions of Training Examples. In *Proceedings of the European Association for Computational Linguistics*, pages 205–209. 8, 71, 72, 73, 75

[64] Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1971–1982. 51

[65] Greg Durrett and Dan Klein. 2014. A Joint Model for Entity Analysis: Coreference, Typing, and Linking. *Proceedings of the Transactions of the Association for Computational Linguistics*. 32

[66] Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. xiii, 47, 60, 61, 62, 63

[67] Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8057–8077, Online. Association for Computational Linguistics. 96

[68] Alon Eirew, Arie Cattan, and Ido Dagan. 2021. WEC: Deriving a large-scale cross-document event coreference dataset from Wikipedia. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2498–2510, Online. Association for Computational Linguistics. 87

[69] Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html. 100, 108

[70] Aymen Elkhlifi and Rim Faiz. 2009. Automatic annotation approach of events in news articles. *International Journal of Computing & Information Sciences*, 7(1):40–50. 22, 23

[71] Günes Erkan and Dragomir R Radev. 2004. LexRank : Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*, 22:457–479. 73

[72] Parvin Sadat Feizabadi and Sebastian Padó. 2014. Crowdsourcing annotation of non-local semantic roles. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 226–230, Gothenburg, Sweden. Association for Computational Linguistics. 48

[73] Christiane Fellbaum. 1998. WordNet. 15

[74] Eraldo Rezende Fernandes, Cícero Nogueira dos Santos, and Ruy Luiz Milidiú. 2012.

Latent structure perceptron with feature induction for unrestricted coreference resolution. *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 41–48. 33

[75] Paolo Ferragina and Ugo Scaiella. 2010. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *CIKM 2010*. 79

[76] Javier Ferrando, Gerard I. Gállego, and Marta R. Costa-jussà. 2022. Measuring the mixing of contextual information in the transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8698–8714, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. 103, 104

[77] Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R. Costa-jussà. 2024. A primer on the inner workings of transformer-based language models. 8

[78] Javier Ferrando and Elena Voita. 2024. Information flow routes: Automatically interpreting language models at scale. x, 8, 100, 101, 103, 104, 110

[79] Francis Ferraro and Benjamin Van Durme. 2016. A Unified Bayesian Model of Scripts , Frames and Language. *Proceedings of the 30th Conference on Artificial Intelligence (AAAI 2016)*, pages 2601–2607. 33, 42

[80] JR Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*. 26

[81] Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. 2021. Causal analysis of syntactic agreement mechanisms in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1828–1843, Online. Association for Computational Linguistics. 101

[82] Joseph Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76. 24

[83] Radu Florian, John F Pitrelli, Salim Roukos, and Imed Zitouni. 2010. Improving mention detection robustness to noisy input. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 335–345. 29

[84] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics. 89

[85] Matthew Gerber and Joyce Chai. 2010. Beyond NomBank: A study of implicit arguments for nominal predicates. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1583–1592, Uppsala, Sweden. Association for Computational Linguistics. 48

[86] Matthew Gerber and Joyce Y. Chai. 2012. Semantic role labeling of implicit arguments for nominal predicates. *Computational Linguistics*, 38(4):755–798. 48

[87] Rhys Gould, Euan Ong, George Ogden, and Arthur Conmy. 2024. Successor heads: Recurring, interpretable attention heads in the wild. In *The Twelfth International Conference on Learning Representations*. 101

[88] Kartik Goyal, Sujay Kumar Jauhar, Mrinmaya Sachan, Shashank Srivastava, Huiying Li, and Eduard Hovy. 2013. A Structured Distributional Semantic Model for Event Coreference. In *Proceedings of The 51st Annual Meeting of the Association for Computational Linguistics*, Bulgaria. 24, 29

[89] Joseph E. Grimes. 1975. *The Thread of Discourse*. De Gruyter Mouton, Berlin, Boston. 71, 76

[90] Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. Finding neurons in a haystack: Case studies with sparse probing. *Transactions on Machine Learning Research*. 101

[91] Mark Hall, Geoffrey Holmes, Bernhard Pfahringer, Ian Witten, Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The WEKA data mining software : An update The WEKA Data Mining Software : An Update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18. 25

[92] Michael Halliday and Ruqaiya Hasan. 1976. *Cohesion in English*. Routledge. 73, 76

[93] Michael Hanna, Ollie Liu, and Alexandre Variengien. 2024. How does gpt-2 compute greater-than? interpreting mathematical abilities in a pre-trained language model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc. 101

[94] Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. 2024. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. 8, 101, 104

[95] Daniel Hardt. 1992. An algorithm for VP ellipsis. In *Proceedings of the 30th annual meeting on Association for Computational Linguistics*, January, pages 9–14. 47, 49

[96] Daniel Hardt. 1997. An empirical approach to VP ellipsis. *Computational Linguistics*, 23(4):525–541. 47

[97] Daniel Hardt. 1998. Improving Ellipsis Resolution with Transformation-Based Learning. *AAAI Fall Symposium*, pages 41–43. 47

[98] Laura Hasler and Constantin Orasan. 2009. Do coreferential arguments make event mentions coreferential? In *Proceedings of DAARC*, pages 151–163. 41

[99] Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 364–369, Melbourne, Australia. Association for Computational Linguistics. 59

[100] Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483, Vancouver, Canada. Association for Computational Linguistics. 59

[101] Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel.

2006. OntoNotes: the 90% solution. In *Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, June, pages 57–60. 87

[102] Eduard Hovy, T Mitamura, F Verdejo, J Araki, and Andrew Philpot. 2013. Events are Not Simple: Identity, Non-Identity, and Quasi-Identity. In *The 1st Workshop on EVENTS: Definition, Detection, Coreference and Representation, NAACL-HLT 2013 Workshop*, pages 21–28, Atlanta. 3, 5, 21, 23, 24, 85, 86, 88, 90, 92, 93

[103] Lifu Huang, Taylor Cassidy, Xiaocheng Feng, Heng Ji, Clare R Voss, Jiawei Han, and Avirup Sil. 2016. Liberal Event Extraction and Event Schema Induction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 258–268. 11

[104] Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*. 101

[105] Kevin Humphreys, Robert Gaizauskas, and Saliha Azzam. 1997. Event coreference for information extraction. In *Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts, 35 th Annual Meeting of Assoc. for Computational Linguistics*, pages 75–81, Madrid. 33

[106] Silja Huttunen, Roman Yangarber, and Ralph Grishman. 2002. Complexity of event structure in IE scenarios. *COLING 2002*, pages 1–7. 21

[107] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*. 133

[108] Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR 2015*. 79

[109] LDC. 2005. ACE (automatic content extraction) english annotation guidelines for events version 5.4.3. *Linguistic Data Consortium*. 47, 48

[110] LDC. 2015. Deft Rich ERE annotation guidelines: Events version 3.0. *Linguistic Data Consortium*. 47, 48

[111] Heeyoung Lee, Yves Peirsman, and Angel Chang. 2011. Stanford's Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task. In *Proc. of the 15th Conference on Computational Natural Language Learning: Shared Task*, June, pages 28–34. xiii, 29

[112] Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint Entity and Event Coreference Resolution across Documents. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing and Natural Language Learning*. 22, 23, 25, 27

[113] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end Neural Coreference Resolution. In *EMNLP 2017*. 32

[114] Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of*

Knowledge Representation and Reasoning, KR'12, page 552–561. AAAI Press. 3, 5, 99, 102

[115] Peifeng Li, Guodong Zhou, Qiaoming Zhu, and Libin Hou. 2012. Employing compositional semantics and discourse consistency in Chinese event extraction. In *EMNLP-CoNLL '12 Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, July, pages 1006–1016. 16

[116] Peifeng Li, Qiaoming Zhu, and Xiaoxu Zhu. 2011. A Clustering and Ranking Based Approach for Multi-document Event Fusion. *12th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pages 159–165. 22, 24

[117] Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82, Sofia, Bulgaria. Association for Computational Linguistics. 48

[118] Qi Li, Heng Ji, and Liang Huang. 2013. Joint Event Extraction via Structured Prediction with Global Features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL2013)*. 73

[119] Tom Lieberum, Matthew Rahtz, János Kramár, Neel Nanda, Geoffrey Irving, Rohin Shah, and Vladimir Mikulik. 2023. Does circuit analysis interpretability scale? evidence from multiple choice capabilities in chinchilla. 101, 102

[120] Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331. 76

[121] Zhengzhong Liu, Jun Araki, Eduard Hovy, and Teruko Mitamura. 2014. Supervised Within-Document Event Coreference using Information Propagation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4539–4544, Reykjavik, Iceland. European Language Resources Association (ELRA). 73

[122] Zhengzhong Liu, Jun Araki, Teruko Mitamura, and Eduard Hovy. 2016. CMU-LTI at KBP 2016 Event Nugget Track. In *TAC 2016*. 17

[123] Zhengzhong Liu, Guanxiong Ding, Avinash Bukkittu, Mansi Gupta, Pengzhi Gao, Atif Ahmed, Shikun Zhang, Xin Gao, Swapnil Singhavi, Linwei Li, Wei Wei, Zecong Hu, Haoran Shi, Xiaodan Liang, Teruko Mitamura, Eric Xing, and Zhiting Hu. 2020. A data-centric framework for composable NLP workflows. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 197–204, Online. Association for Computational Linguistics. 89, 91

[124] Zhengzhong Liu, Teruko Mitamura, and Eduard Hovy. 2015. Evaluation Algorithms for Event Nugget Detection : A Pilot Study. In *Proceedings of the 3rd Workshop on EVENTS at the NAACL-HLT*, pages 53–57. 17

[125] Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, Richard Fan, Yi Gu, Victor Miller, Yonghao Zhuang, Guowei He, Haonan Li, Fajri Koto, Liping Tang, Nikhil Ranjan, Zhiqiang

Shen, Xuguang Ren, Roberto Iriondo, Cun Mu, Zhiting Hu, Mark Schulze, Preslav Nakov, Tim Baldwin, and Eric P. Xing. 2023. Llm360: Towards fully transparent open-source llms. x, 8, 100, 110

[126] Zhengzhong Liu, Bowen Tan, Hongyi Wang, Willie Neiswanger, Tianhua Tao, Haonan Li, Fajri Koto, Yuqi Wang, Suqi Sun, Omkar Pangarkar, Richard Fan, Yi Gu, Victor Miller, Liqun Ma, Liping Tang, Nikhil Ranjan, Yonghao Zhuang, Guowei He, Renxi Wang, Mingkai Deng, Robin Algayres, Yuanzhi Li, Zhiqiang Shen, Preslav Nakov, and Eric Xing. 2024. Llm360 k2-65b: Scaling up fully transparent open-source llms. 108

[127] Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*. 96

[128] Jing Lu and Vincent Ng. 2017. Joint Learning for Event Coreference Resolution. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 90–101. 33, 73

[129] Jing Lu, Deepak Venugopal, Vibhav Gogate, and Vincent Ng. 2016. Joint Inference for Event Reference Resolution. In *Proceedings of the 26th International Conference on Computational Linguistics*. 33

[130] Xiaoqiang Luo. 2005. On coreference resolution performance metrics. *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 25–32. 26, 39

[131] Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics. 91

[132] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. *Proceedings of 52nd Annual Meeting of the ACL: System Demonstrations*, pages 55–60. xiii, 15, 37, 75

[133] Daniel Marcu. 1999. Discourse Trees are Good Indicators of Importance in Text. *Advances in Automatic Text Summarization*, pages 123–136. 73

[134] Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. 2024. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *Computing Research Repository*, arXiv:2403.19647. 101

[135] Katie McConky, Rakesh Nagi, Moises Sudit, and William Hughes. 2012. Improving event co-reference by context extraction and dynamic feature weighting. *2012 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support*, pages 38–43. 22, 24

[136] Callum Stuart McDougall, Arthur Conmy, Cody Rushing, Thomas McGrath, and Neel Nanda. 2024. Copy suppression: Comprehensively understanding an attention head. 101

[137] PN Mendes, Max Jakob, Andres Garcia-silva, and Christian Bizer. 2011. DBpedia spotlight: shedding light on the web of documents. *Proceedings of the 7th Conference on Semantic*

Systems, pages 1–8. 26

[138] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 35. 101

[139] Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2024. Circuit component reuse across tasks in transformer language models. 101

[140] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. 2017. Mixed precision training. *arXiv preprint arXiv:1710.03740*. 134

[141] Tomas Mikolov, I Sutskever, and Kai Chen. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119. 76, 79

[142] Anne-Lyse Minard, Manuela Speranza, Ruben Urizar, Begoña Altuna, Marieke van Erp, Anneleen Schoen, and Chantal van Son. 2016. MEANTIME, the NewsReader multilingual event and time corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4417–4422, Portorož, Slovenia. European Language Resources Association (ELRA). 87

[143] Teruko Mitamura, Zhengzhong Liu, and Eduard Hovy. 2015. Overview of TAC KBP 2015 Event Nugget Track. In *TAC KBP 2015*, pages 1–31. 3, 11, 13, 17, 33, 38, 74

[144] Teruko Mitamura, Zhengzhong Liu, and Eduard Hovy. 2017. Overview of TAC-KBP 2016 Event Nugget Track. In *TAC 2016*. 3, 13, 17, 18

[145] Teruko Mitamura, Zhengzhong Liu, and Eduard Hovy. 2018. Events Detection, Coreference and Sequencing: What's next? Overview of the TAC KBP 2017 Event Track. In *TAC 2017*, pages 1–42. 3, 13, 31, 87

[146] Jane Morris and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17:21–48. 73

[147] Neel Nanda. 2023. Attribution patching: Activation patching at industrial scale. 101

[148] Martina Naughton. 2009. *Sentence Level Event Detection and Coreference Resolution*. Ph.D. thesis, National University of Ireland, Dublin. 22, 23

[149] Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, July, pages 104–111, Philadelphia. 26, 36

[150] Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309, San Diego, California. Association for Computational Linguistics. 48

[151] Thien Huu Nguyen and Ralph Grishman. 2015. Event Detection and Domain Adaptation with Convolutional Neural Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 365–371. 73

[152] Leif Arda Nielsen. 2003. Using Machine Learning techniques for VPE detection. In *Proceedings of Recent Advances in Natural Language Processing*, pages 339–346. 47

[153] Leif Arda Nielsen. 2004. Robust VPE detection using automatically parsed text. In *Proceedings of the Student Workshop, ACL*, pages 31–36. 47

[154] Leif Arda Nielsen. 2004. Using Automatically Parsed Text for Robust VPE detection. In *Proceedings of the fifth Discourse Anaphor and Anaphora Reolution conference (DAARC)*. 47

[155] Leif Arda Nielsen. 2005. *A corpus-based study of Verb Phrase Ellipsis Identification and Resolution*. Doctor of philosopy, King's College London. 47, 48, 49, 53, 54

[156] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*. Https://distill.pub/2020/circuits/zoom-in. 101

[157] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. 101, 108

[158] Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. 2018. A span selection model for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1630–1642, Brussels, Belgium. Association for Computational Linguistics. 59, 60

[159] Feng Pan, Rutu Mulkar, and Jerry R. Hobbs. 2006. Extending TimeML with typical durations of events. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, July, pages 38–45. 11

[160] Feng Pan, Rutu Mulkar, and Jerry R. Hobbs. 2006. Learning event durations from event descriptions. In *ACL-44 Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, July, pages 393–400. 11

[161] Ted Pedersen and Jason Michelizzi. 2004. WordNet :: Similarity - Measuring the Relatedness of Concepts. *HLT-NAACL–Demonstrations '04 Demonstration Papers at HLT-NAACL 2004*, pages 38–41. 29

[162] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*. 133

[163] Haoruo Peng, Daniel Khashabi, and Dan Roth. 2015. Solving Hard Coreference Problems. In *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*, pages 809–819, Denver, Colorado. 43

[164] Haoruo Peng, Yangqi Song, and Dan Roth. 2016. Event Detection and Co-reference with

Minimal Supervision. In *EMNLP 2016*. 33, 73

[165] Karl Pichotta. 2016. Learning Statistical Scripts With LSTM Recurrent Neural Networks. In *In Proceedings of the 30th AAAI Conference on Artificial Intelligence*, February. 73

[166] Karl Pichotta and Raymond J. Mooney. 2016. Using Sentence-Level LSTM Language Models for Script Inference. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 279–289. 33, 42

[167] Marco Ponza, Paolo Ferragina, and Francesco Piccinno. 2019. Swat: A system for detecting salient wikipedia entities in texts. 73

[168] Sameer Pradhan, Alessandro Moschitti, and Olga Uryupina. 2012. CoNLL-2012 Shared Task : Modeling Multilingual Unrestricted Coreference in OntoNotes. In *Conll*, June, pages 1–40. 26

[169] Sameer S. Pradhan, Lance Ramshaw, Ralph Weischedel, Jessica MacBride, and Linnea Micciulla. 2007. Unrestricted Coreference: Identifying Entities and Events in OntoNotes. *International Conference on Semantic Computing (ICSC 2007)*, pages 446–453. 22, 23

[170] James Pustejovsky, Patrick Hanks, Roser Saurí, Andrew See, Robert Gaizauskas, Andrea Setzer, Beth Sundheim, David Day, Lisa Ferro, and Dragomir. 2002. The TIMEBANK Corpus. *Natural Language Processing and Information Systems*, 4592:647–656. 6, 11, 33, 73

[171] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. x, 110

[172] Altaf Rahman and Vincent Ng. 2009. Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, August, pages 968–977. 51

[173] Marta Recasens and Eduard Hovy. 2011. Blanc: Implementing the rand index for coreference evaluation. *Natural Language Engineering*, 1(1). 26, 39, 95

[174] Marta Recasens, Eduard Hovy, and M. Antònia Martí. 2010. A Typology of Near-Identity Relations for Coreference (NIDENT). *7th International Conference on Language Resources and Evaluation (LREC-2010)*, pages 149–156. 81

[175] Marta Recasens, Eduard Hovy, and M. Antonia Marti. 2011. Identity, non-identity, and near-identity: Addressing the complexity of coreference. *Lingua*, 121(6):1138–1152. 23, 85, 88

[176] Marta Recasens, Mc De Marneffe, and Christopher Potts. 2013. The Life and Death of Discourse Entities: Identifying Singleton Mentions. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, June, pages 627–633. 74

[177] Marta Recasens, M Antònia Martí, and Constantin Orasan. 2012. Annotating Near-Identity from Coreference Disagreements. In *Proceedings of The Eight International Conference on Language Resources and Evaluation (LREC 2012)*, pages 165–172. 7, 86, 88, 90

[178] Rachel Rudinger, Pushpendre Rastogi, Francis Ferraro, and Benjamin Van Durme. 2015. Script Induction as Language Modeling. In *Proceedings of the 2015 Conference on*

Empirical Methods in Natural Language Processing, pages 1681–1686. 33, 42, 73

[179] Josef Ruppenhofer, Caroline Sporleder, Roser Morante, Collin Baker, and Martha Palmer. 2009. SemEval-2010 task 10: Linking events and their participants in discourse. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pages 106–111, Boulder, Colorado. Association for Computational Linguistics. 48

[180] Josef Ruppenhofer, Caroline Sporleder, Roser Morante, Collin Baker, and Martha Palmer. 2010. SemEval-2010 task 10: Linking events and their participants in discourse. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 45–50, Uppsala, Sweden. Association for Computational Linguistics. 48

[181] Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. 2021. Neuron-level interpretation of deep nlp models: A survey. *Transactions of the Association for Computational Linguistics*, 10:1285–1303. 101

[182] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. 8

[183] Evan Sandhaus. 2008. The New York Times Annotated Corpus. 8, 73

[184] S Sangeetha and Michael Arock. 2012. Event Coreference Resolution using Mincut based Graph Clustering. *International Journal of Computing & Information Sciences*, pages 253–260. 22, 24, 33

[185] Roger C Schank and Robert P Abelson. 1977. *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum Associates. 3, 6, 31, 72

[186] Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*. 59, 62

[187] Stuart M. Shieber, Fernando C. N. Pereira, and Mary Dalrymple. 1996. Interactions of scope and ellipsis. *Linguistics and Philosophy*, 19(5):527–552. 47

[188] E Skorochod'ko. 1971. Adaptive Method of Automatic Abstracting and Indexing. In *Proceedings of the IFIP Congress 71*. 73

[189] Zhiyi Song, Ann Bies, Justin Mott, Xuansong Li, Stephanie Strassel, and Christopher Caruso. 2018. Cross-document, cross-language event coreference annotation using event hoppers. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA). 5, 87, 91

[190] Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895, New Orleans, Louisiana. Association for Computational Linguistics. 89

[191] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*. 103, 133

[192] A Sun, R Grishman, and S Sekine. 2011. Semi-supervised relation extraction with large-scale word clustering. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 521–529. 15

[193] Aaquib Syed, Can Rager, and Arthur Conmy. 2023. Attribution patching outperforms automated circuit discovery. In *NeurIPS Workshop on Attributing Model Behavior at Scale*. 101

[194] Curt Tigges, Michael Hanna, Qinan Yu, and Stella Biderman. 2024. Llm circuit analyses are consistent across training and scale. 101

[195] Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. 2023. Linear representations of sentiment in large language models. 101

[196] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*. 133

[197] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*. 133

[198] Stephen Tratz and Eduard Hovy. 2011. A fast, accurate, non-projective, semantically-enriched parser. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2010, pages 1257–1268. 15, 26, 38

[199] Shyam Upadhyay, Nitish Gupta, Christos Christodoulopoulos, and Dan Roth. 2016. Revisiting the evaluation for cross document event coreference. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1949–1958, Osaka, Japan. The COLING 2016 Organizing Committee. 95

[200] Naushad UzZaman. 2012. *Interpreting the Temporal Aspects of Language*. Ph.D. thesis, University of Rochester. 39

[201] Naushad Uzzaman and James F Allen. 2010. TRIPS and TRIOS System for TempEval-2: Extracting Temporal Information from Text. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, July, pages 276–283. 33

[202] Naushad UzZaman, Hector Llorens, Leon Derczynski, Marc Verhagen, James Allen, and James Pustejovsky. 2013. Semeval-2013 task 1: {T}empeval-3: Evaluating time expressions, events, and temporal relations. In *Second joint conference on lexical and computational semantics (* SEM)*, volume 2, pages 1–9. 3, 39

[203] Vikrant Varma, Rohin Shah, Zachary Kenton, János Kramár, and Ramana Kumar. 2023. Explaining grokking through circuit efficiency. 101

[204] Zeno Vendler. 1957. Verbs and times. *The Philosophical Review*, 66(2):143–160. 74

[205] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33,

pages 12388–12401. Curran Associates, Inc. 101

[206] Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. 2023. Neurons in large language models: Dead, n-gram, positional. 101

[207] Piek Vossen and Tommaso Caselli. 2015. Storylines for structuring massive streams of news. In *Proceedings of the First Workshop on Computing News Story Lines*, pages 40–49. 3, 4, 71, 73, 75, 78

[208] Piek Vossen, Filip Ilievski, Marten Postma, and Roxane Segers. 2018. Don't annotate, but validate: a data-to-text method for capturing event data. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA). 87

[209] Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. ACE 2005 multilingual training corpus. *Linguistic Data Consortium*, 57. 47, 48, 87

[210] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. x, 8, 100, 101, 104, 110

[211] Xiaozhi Wang, Ziqi Wang, Xu Han, Zhiyuan Liu, Juanzi Li, Peng Li, Maosong Sun, Jie Zhou, and Xiang Ren. 2019. HMEAE: Hierarchical modular event argument extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5776–5782, Hong Kong, China. Association for Computational Linguistics. 48

[212] David Wiggins et al. 1967. *Identity and spatio-temporal continuity*. Blackwell Oxford. 5, 95

[213] Jennifer Williams. 2012. Extracting fine-grained durations for verbs from Twitter. In *ACL '12 Proceedings of ACL 2012 Student Research Workshop*, July, pages 49–54. 11

[214] Jennifer Williams and Graham Katz. 2012. Extracting and modeling durations for habits and events from Twitter. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics Volume 2 Short Papers*, July, pages 223–227. 11

[215] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics. 96

[216] Kristin Wright-Bettner, Martha Palmer, Guergana Savova, Piet de Groen, and Timothy Miller. 2019. Cross-document coreference: An approach to capturing coreference without context. In *Proceedings of the 10th International Workshop on Health Text Mining and Information Analysis*, pages 1–10. 5, 21, 88

[217] Kristin Wright-Bettner, Martha Palmer, Guergana Savova, Piet de Groen, and Timothy

Miller. 2019. Cross-document coreference: An approach to capturing coreference without context. In *Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis (LOUHI 2019)*, pages 1–10, Hong Kong. Association for Computational Linguistics. 90, 93

[218] Mengzhou Xia, Mikel Artetxe, Chunting Zhou, Xi Victoria Lin, Ramakanth Pasunuru, Danqi Chen, Luke Zettlemoyer, and Ves Stoyanov. 2023. Training trajectories of language models across scales. In *Association for Computational Linguistics (ACL)*. 101

[219] Chenyan Xiong, Zhengzhong Liu, Jamie Callan, and Eduard Hovy. 2017. JointSem: Combining Query Entity Linking and Entity based Document Ranking. In *Proceedings of the 26th ACM International Conference on Information and Knowledge Management (CIKM 2017)*, pages 2391–2394. 77, 79

[220] Chenyan Xiong, Zhengzhong Liu, Jamie Callan, and Tie-Yan Liu. 2018. Towards Better Text Understanding and Retrieval through Kernel Entity Salience Modeling. In *SIGIR 2018*. 3, 4, 71, 73, 78

[221] Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. 103, 133

[222] Congle Zhang, Stephen Soderland, and Daniel S. Weld. 2015. Exploiting parallel news streams for unsupervised event extraction. *Transactions of the Association for Computational Linguistics*, 3:117–129. 3, 4, 71, 73