Towards Syntax and Semantics-Driven Neural Coreference Resolution

Fan Jiang
School of Computing and Information Systems
The University of Melbourne
Supervisor: Prof. Trevor Cohn

Student Number: 1042268
Project Type: Research Project
Subject Code: COMP90070
Credits Points: 75cp

Submitted in partial fulfilment of the requirements for the degree of
Master of Science (Computer Science)

January 2022

Abstract

Teaching machines to understand the semantics of documents is one of the most difficult and long-standing challenges in the Natural Language Processing (NLP) community. As a core task in NLP, the coreference resolution task aims to group all mentions that refer to the same real-world entity. With the development of deep learning techniques, neural end-to-end coreference resolution models have become dominant and conventional wisdom is that hand-crafted features derived from both syntax and semantics are redundant, as they are believed to be automatically captured in document representations by neural models. However, this claim has not been thoroughly tested. Therefore, in this thesis, we focus on evaluating the impact of incorporating external syntax and semantics for neural coreference resolution models.

This thesis consists of two parts. In the first part, we present a heterogeneous graph-based model to incorporate syntactic and semantic structures of sentences. The proposed graph contains a syntactic sub-graph where tokens are connected based on a dependency tree and a semantic sub-graph that contains arguments and predicates as nodes and semantic role labels as edges. In the second part of this thesis, we build a graph based on the structures of constituent parse trees. We argue that although most leading systems use only dependency trees, constituent trees also encode important information, such as explicit span-boundary signals, extra linguistic labels and hierarchical structures useful for detecting anaphora. In order to fully exploit constituent parse tree structures, we further introduce edges with different orders and interpret tree structures from different views. Novel information propagation mechanisms are designed in both parts to enable information flow among different nodes in the graph.

The methodologies employed in both parts of this thesis are novel. They deliver convincing and promising results supported by our thorough and large-scale experiments on

standard benchmark datasets across different languages, and new state-of-the-art performance is achieved on a standard Chinese coreference resolution dataset.

Declaration

I certify that:

I. This thesis does not incorporate without acknowledgement any material previously submitted for degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the text.

II. This thesis is fewer than 25000 words in length (excluding text in images, table, bibliographies and appendices).

Fan Jiang
January 2022

Citations to Previously Published Work

Large portions of Chapter 3 have appeared in the following paper:

> Fan Jiang and Trevor Cohn. 2021. Incorporating syntax and semantics in coreference resolution with heterogeneous graph attention network. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1584–1591, Online. Association for Computational Linguistics.

Large portions of Chapter 4 have appeared in the following paper:

> Fan Jiang and Trevor Cohn. 2021. Incorporating Constituent Syntax for Coreference Resolution. In Proceedings of the 36th AAAI Conference on Artificial Intelligence.

Acknowledgments

First and foremost, I would like to express my greatest thanks and sincere gratitude to my supervisor Prof. Trevor Cohn. Without a doubt, I have learnt a lot from him and what I have learnt from him will benefit me for the rest of my life. I first met Trevor in December 2019 when I was looking for supervisors for my research project (it is also the only one in-person meeting with him) and I knew little about NLP. Trevor is a person who is extremely enthusiastic for research and I could always be enlightened by his new ideas and insights (although overwhelming sometimes) every time I met with him. I really appreciate that I can complete my research project under his supervision.

I am incredibly grateful to my parents. Like most Chinese students in my generation, I am the only child of my family and I maintain a very close relationship with them (more like friends). My parents made me who I am, both physically and mentally, and I could not make it this so far without their support. Great thanks also to my grandparents for their unconditional love since my childhood.

Notations

We provide a brief summary of some mathematical notations used throughout this thesis (unless stated otherwise).

i, j: both i and j are mention spans consisting of one or more tokens, where $i = [\text{start}_i, \text{end}_i]$, $j = [\text{start}_j, \text{end}_j]$ and start and end means the token indices of endpoints of mention spans.

$\mathcal{Y}_i = \{\epsilon, 1, \ldots, c-1\}$: a list of associated candidate antecedent mentions of a given mention i

$s_m(i)$: the predicted mention score of a given mention i

$c(i, j)$: the coarse pairwise coreference score between mention i and j

$s_c(i, j)$: the fine-grained pairwise coreference score between a pair of mentions i and j

$s(i, j)$: the final pairwise coreference score between mention i and j

$\mathcal{N}_i$: the set of neighbour nodes of a given node i

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Coreference Resolution, the task of grouping all text mention spans referring to the same real-world entity, has been a core task in the research field of the Natural Language Processing (NLP) community since the 1960s. An example for illustrating this task is shown in Figure 1.1. In this example, we have a dialogue between two speakers. Text spans, such as *both of you* and *I*, which consist of one or more tokens, are defined as mentions. Here we have *you* refers *I* because they represent the same person, and we say *you* is anaphoric and *I* is the antecedent of *you*. We can also have more complex clusters involving arbitrary mention spans, such as the coreference cluster {*both of you*, *us* and *we*}.

Despite being intensively investigated for about 60 years, coreference resolution task is still far from being resolved due to the difficultly derived naturally. Below is an example from the Winograd Schema Challenge dataset (Levesque et al., 2012), which can better demonstrate the difficulty of the task:

> The *city councilmen* refused the demonstrators a permit because *they* feared violence.
> The city councilmen refused *the demonstrators* a permit because *they* advocated violence.

Determining the correct reference of *they* requires understanding that the second clause after *because* serves as the explanation for the first clause, and also that *city councilmen* are more likely to fear violence. In contrast, *demonstrators* are more likely to advocate violence, which requires the model to have access to external commonsense knowledge

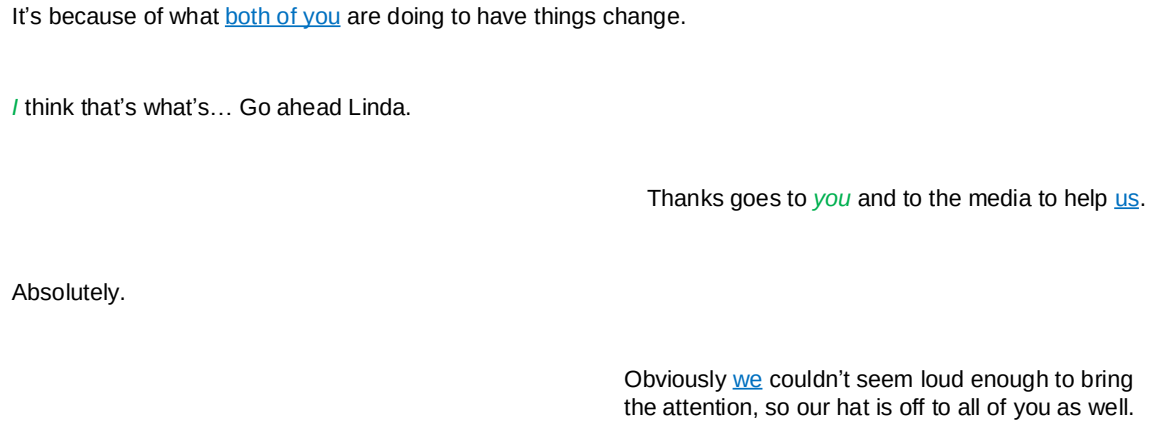

Figure 1.1: An example for coreference resolution task. Mentions in the same coreference cluster are presented with the same colour (Example from Wiseman et al. (2016)).

about the world. By contrast, human audiences typically have few problems identifying the correct antecedent of the pronoun *they* by utilising background knowledge.

From the above example, we can see that the difficulty of the coreference resolution task mainly derives from the intensive background knowledge required for reasoning, which is typically difficult for machines to acquire. Despite the difficulty, coreference resolution plays an essential role in aggregating entity-related information when interpreting document texts, which is crucial for a variety of higher-level NLP tasks including relation extraction (Luan et al., 2019; Wadden et al., 2019), machine reading comprehension (Dasigi et al., 2019), document summarisation (Xu et al., 2020) and neural machine translation (Stojanovski and Fraser, 2018).

The focus of coreference resolution research has undergone three main stages, namely heuristics and rule-based methods, machine learning approaches, and end-to-end neural models. Like many other NLP tasks, the coreference resolution task was first dealt with by designing sophisticated rules by taking inspiration from various knowledge resources and inference procedures. Computational theories of discourse, including focusing (Grosz, 1977) and centering (Grosz et al., 1983, 1995) have largely driven the development in this stage (the 1970s and 1980s). The research community gradually started focusing on statistical machine learning approaches since the 1990s, partially due to the wide applications of statistical methods in the NLP community and other Artificial Intelligence areas. Another reason is the public availability of some small and moderate-sized annotated

coreference corpus (MUC-6 (muc, 1995) and MUC-7 (muc, 1998)). Learning-based methods in this stage mainly employed hand-designed features derived from linguistics and external knowledge sources to train supervised models. Three main model frameworks have been developed, namely mention-pair (Soon et al., 2001; Ng and Cardie, 2002b), mention-ranking (Iida et al., 2003; Yang et al., 2003; Denis and Baldridge, 2008) and entity-based (Luo et al., 2004; Yang et al., 2004; Rahman and Ng, 2009). The hybrid of rule-based and learning-based methods has become popular around the 2010s with the advent of multi-pass sieve-based models (Raghunathan et al., 2010; Lee et al., 2011; LEE et al., 2017), which outperformed the learning-based counterparts on large-sized corpus (Pradhan et al., 2011, 2012). The drastic shift to neural model-based methods began in 2017 when the first end-to-end neural coreference resolution model (Lee et al., 2017) has proposed. Various improved work has been proposed (Lee et al., 2018; Kantor and Globerson, 2019; Joshi et al., 2019, 2020; Wu et al., 2020) and impressive progress has been made in recent years, with a promising improvement of 15.9% on the standard OntoNotes 5.0 English benchmark dataset over the past three years.

Traditional learning-based methods depend heavily on various hand-engineered features derived from both syntactic and semantic patterns. In contrast, neural methods aim to design appropriate model architectures to enable end-to-end training without too many human interventions, resulting in significantly less reliance on external features. However, whether conventional wisdom employed in learning-based methods can benefit strong end-to-end neural models remains unknown.

This thesis aims to investigate whether various kinds of syntax and semantics widely used by early statistical machine learning methods can show positive impacts on neural coreference resolution models trained in an end-to-end fashion. We show that leveraging external syntax and semantics with our carefully designed graph-based methods can be highly effective and show promising results.

In this chapter, we briefly discuss the motivation behind our research and propose research questions in Section 1.1. Then we finish this chapter by outlining the thesis structure in Section 1.2.

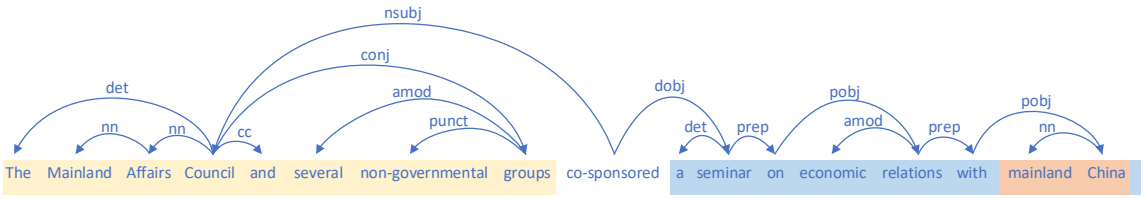

Figure 1.2: An example sentence with mention spans and annotated dependencies in the OntoNotes 5.0 English dataset

1.1 Motivation

1.1.1 Syntax

Syntactic features derived from syntactic parse trees are widely used in early learning-based methods. Ge et al. (1998) proposes Hobbs distances to encode the rank of candidate antecedents of a given pronoun based on Hobbs's syntax parse tree based pronoun resolution algorithm (Hobbs, 1978). Bergsma and Lin (2006) implements path-related features based on syntactic parse trees, where the sequence of words and dependency labels in the path between a given pronoun and its candidate antecedent is utilised. Statistical information collected from such paths is used to measuring the likelihood of being coreferent for the pronoun and antecedent. Constituent and dependency syntactic information has also been applied in the anaphoricity determination task by using tree-kernel-based methods: Kong et al. (2010) and Kong and Zhou (2011) design various kinds of path-related features such as root path between the root node and current mention. All previous work shows that carefully designed features derived from both constituent and dependency syntax reveal strong signals for resolving coreference, which motivates our research to a large extent.

Dependency syntax captures the bilexical relations between pairs of words and non-linear structures of sentences. Moreover, some words that form valid mention spans have associated complete subtree embedded in the graph from our observation. Considering the dependency tree presented in Figure 1.2. We can see that the mention span *mainland China* is a complete unit, and it should form a subtree in this example. The same applies to the other two mentions *The Mainland Affairs Council and several non-governmental groups* and *a seminar on economic relations with mainland China*. This means that the dependency tree implicitly encodes the information for identifying mention spans. We

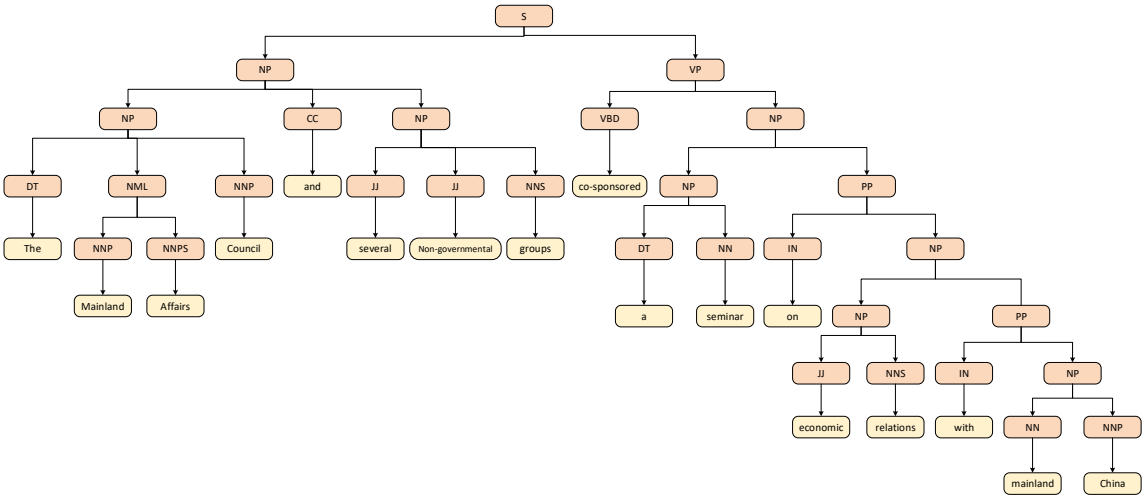

Figure 1.3: An example of constituent parse tree in the OntoNotes 5.0 English dataset

suspect that this kind of strong relationship can be helpful to the mention detection task, which in turn benefits the overall performance of coreference resolution.

Unlike the dependency tree, which models the relationships between individual words, the constituent parse tree captures the syntactic structure in the form of nested multi-word phrases, and phrase structures are labelled with linguistic tags such as part-of-speech and phrase tags. Since they model the relationships of different semantic units, we suspect that these two kinds of syntactic structures may behave quite differently in capturing different aspects of syntactic phenomena. Figure 1.3 shows an example of constituent parse tree. We can see that all those three spans mentioned above have matched constituent phrases in the parse tree. Moreover, the constituent parse tree structures also provide the boundary information for each phrase. By contrast, such information is either implicitly encoded or not revealed in the dependency tree. On the other hand, we find that gold mentions on the OntoNotes 5.0 English dataset map to a limited range of constituent phrases and POS tags. Therefore, we believe constituent tree structures capture signals for mention detection more effectively and explicitly, which we suspect is more helpful to the overall coreference resolution task.

Therefore, by drawing inspirations from previous work and our observations from both dependency and constituent parse trees, we argue that leveraging such syntactic structures can be highly effective for the coreference resolution task. The primary research challenge

is how we can make use of these structures to improve our neural coreference resolution systems. We ask the following research questions:

- We can identify valid mention spans from the subtree embedded in the dependency tree and phrases in the constituent tree. Can we impose certain constraints on the neural coreference resolution model to filter out invalid candidate mentions, thereby reducing spans that are noisy for the mention-linking stage as many as possible?
- Besides, the structures of both syntactic trees are useful and reveal signals for coreference resolution. How can we effectively encode the complete structures to enable our model to take full advantage of such structural information when resolving coreferences?

1.1.2 Semantics

Semantic features have also been intensively investigated in the literature. Selectional preference, which prefers to choose the candidate antecedent sharing the same verb and same semantic role with the pronoun to be resolved, has been widely employed in early days (Dagan and Itai, 1990; Kehler et al., 2004). WordNet and Wikipedia have also been used to derive semantic relatedness and compatibility between pairs of mentions (Kübler and Zhekova, 2016; Ponzetto and Strube, 2006a). Semantic role features were also utilised by taking inspiration from semantic parallelism heuristics (Ponzetto and Strube, 2006a,b) or employing the centering theory from the semantic perspective (Kong et al., 2009).

In particular, Semantic Role Labelling (SRL) models the semantic roles of arguments. It provides semantic relationships between arguments and predicates, revealing the semantics of *Who did what to whom*, which allows us to capture the information of document-level event descriptions. Ponzetto and Strube (2006a) shows that SRL features can be used to link two referring mentions correctly in some scenarios by employing the semantic parallelism heuristic:

> A state commission of inquiry into the sinking of the Kursk will convene in *Moscow* on Wednesday, *the Interfax news agency* reported. It said that the diving operation will be completed by the end of next week.

From above example (from Ponzetto and Strube (2006a)), we can see that *the Interfax news agency* is the argument of the predicate *reported*, and *It* is the argument of the predicate

said. By knowing *reported* and *it* conveys similar semantics, we could easily link *It* to *the Interfax news agency* rather than *Moscow*.

Therefore, the deep semantics captured by SRL can reveal strong signals for coreference reasoning in some scenarios. But unlike the syntactic trees, which have clear structures, SRL only models the relations between arguments and predicates. How to build a tree or graph based on SRL features remains a challenge. We ask the following questions:

- How can we design appropriate methods to organize SRL features into graph or tree structures so that different arguments and predicates can interact with each other to improve coreference reasoning?
- Once the SRL feature-based structures are well designed, how can we model such structures to incorporate useful information effectively for coreference resolution?

1.2 Thesis Outline

The structure of this thesis is organized as follows:

In Chapter 2, we first give a brief description of different types of coreference, widely-used coreference datasets and standard evaluation metrics. Next, a thorough overview of the history and recent development in the field of coreference resolution are presented, where we give priority to the neural model-based methods. The challenges, strengths and weaknesses of different methods are well discussed and compared.

In Chapter 3, we explore the utility of dependency syntax and SRL semantics for neural entity coreference resolution task, where a heterogeneous graph consisting of semantic units with different granularity is constructed and a novel information propagation mechanism is designed to effectively capture coreference-related syntactic and semantic information. Our thorough experiments on two standard benchmark datasets demonstrate the proposed method is highly effective and the leveraged external syntax and semantics are consistently helpful even in the era of deep learning.

In Chapter 4, we further show that leveraging syntax in the form of constituent parsing trees with edges of longer ranges and dual graphs can bring significant benefits. A novel way of representing constituent non-terminal nodes is proposed, and effective

information propagation mechanisms are employed to benefit coreference reasoning. Our large-scale experiments on datasets with two languages confirm the effectiveness of our proposed model across different languages, and we establish new state-of-the-art performance on the Chinese portion of OntoNotes 5.0 benchmark.

In Chapter 5, we summarise the findings of this thesis and discuss possible avenues for future work.

Chapter 2

Literature Review

To have a comprehensive survey of current literature, we provide a detailed summarisation of the development of coreference resolution task, with the focus on neural models, in this chapter. Besides, we also present the brief description of different types of coreference, widely-used coreference resolution datasets and evaluation metrics.

2.1 Coreference Types

Hirst (1981) and Lappin and Leass (1994) have classified coreference into a variety of types. In this section, we present a brief outline of a few types of coreference based on their work.

Zero Anaphora This type of anaphora uses the gap between phrases or clauses to refer back to the antecedent. This is a special case in pro-dropped languages, such as Chinese and Japanese, where the anaphoric expressions are eliminated.

An example is shown in Figure 2.1. We use ϕ to represent the zero pronouns. In this example, we can assign the mention *the government* that appears in earlier contexts to be the antecedent of ϕ_2 while there are no such mentions for ϕ_1.

Pronominal Anaphora Pronominal anaphora is the most common type of coreference, which is realised by using pronouns referring back to a mention in preceding contexts.[1]

[1]The term *preceding contexts* means document texts appearing before a given mention.

This times earthquake has some rooms collapse of this inside if has
这 次 地震 ϕ_1 有 一些 房屋 塌 的，这 里面 如果 有
house construction of quality issues be must investigate duty of
建房 的 质量 问题， ϕ_2 是 要 追究 责任 的。

In this earthquake ϕ_1 some rooms collapsed, if there exist some room quality issues, ϕ_2 will need to call to account.

Figure 2.1: An example for zero pronoun. The interlinear gloss text and its English translation are presented.

Furthermore, the referent antecedent can be classified into three types: definite, indefinite and adjective. Definite means the pronoun refers to a specific and unique entity (e.g., the tower), while indefinite means that the referred entity is not described with a determinant (e.g., tourists). Adjective indicates the referred mention is described with an adjective (e.g., beautiful city skylines).

Non-Anaphoric Mentions or Singletons In some datasets, the non-anaphoric mentions or singletons, which are not coreferent with any previously mentioned entity (antecedent), are annotated. The most common one is the pleonastic pronoun *it* (e.g., It is rainy today.). For other general mention spans, singleton means a certain entity only appears once and is never mentioned in the subsequent document texts.[2]

Split Anaphora This type of coreference means that an anaphoric mention can refer back to more than one antecedent.

> John likes *green*, Mary likes *blue*, but Tom likes both colours.

In this example, we can see that the mention both colours refers backs to two antecedents *green* and *blue*. Such cases are not annotated on the OntoNotes dataset.

Generics Generics means a specific mention and its antecedent does not necessarily refer to the same real-world entity. The given mention or its antecedent also may not correspond to a specific entity.

[2]The term *subsequent document texts* means document texts appearing after a given mention

> *Meetings* are most productive when *they* are held in the morning. *Those meetings*, however, generally have the worst attendance.

In this example (borrowed from Pradhan et al. (2012)), the mention *Meetings* does not refer to a specific meeting but is treated as a generic term. In other words, it does not correspond to a certain real-world entity. Also, *Those meetings* and *Meetings* do not strictly refer to the same real-world entity (*Those meetings* is actually the subset of *Meetings*).

2.2 Datasets

CoNLL-2012 The CoNLL-2012 shared task (Pradhan et al., 2012) contains a standard document-level coreference resolution dataset based on the OntoNotes Release 5.0.[3][4] The documents are from telephone conversations, newswire, newsgroups, broadcast news, weblogs, broadcast conversation, and religious texts. This dataset annotates coreference links of noun phrases (e.g., named or definite nominal mentions), verbs, pronouns, generic mentions, proper pre-modifiers, copular verbs, small clauses, temporal expressions and appositives. Three metrics: MUC (Vilain et al., 1995), B-CUBED (B^3) (Bagga and Baldwin, 1998) and Entity-based CEAF ($CEAF_{\phi 4}$) (Luo, 2005) (see §2.3 for details) and their average F_1 score are commonly used for model performance evaluation.

GAP GAP (Webster et al., 2018) is designed for identifying ambiguous gender pronouns.[5] Each instance is a short document with two candidates. Each candidate is associated with a boolean value representing whether it is the correct reference of a given gendered pronoun. Models are evaluated by overall F_1 score, F_1 scores by gender (Masculine: F_1^M and Feminine: F_1^F) and gender bias (the ratio of F_1^M to F_1^F).

ACE 2005 (Walker and Consortium, 2005) dataset is a multilingual dataset consist of three languages, namely English, Chinese and Arabic. It contains annotations for Name Entity Recognition (NER), Relation Extraction (RE), and Entity and Event Coreference Resolution. It has been extensively evaluated by various neural NER and RE models and

[3] https://catalog.ldc.upenn.edu/LDC2013T19
[4] We use CoNLL-2012 shared task and OntoNotes 5.0 dataset interchangeably throughout this thesis.
[5] https://github.com/google-research-datasets/gap-coreference

serves as a standard benchmark for these two tasks. In contrast, it has been forgotten by many modern neural coreference resolution models after the release of the OntoNotes 5.0 benchmark. Like the CoNLL-2012 dataset, the documents are sampled from broadcast conversation, broadcast news, conversational telephone speech, newswire, discussion forums, and weblogs. Coreference links among proper names, nominals and pronouns are annotated. Unlike the CoNLL-2012 dataset, coreferent mentions should be with the same entity type. Besides, singleton mentions are annotated, and speaker information is not available. For evaluation, it uses the same metrics as CoNLL-2012, but singleton mentions are included in the evaluation phase.

PreCo (Chen et al., 2018) is a large-scale English dataset for coreference resolution, containing 38K documents and 12.5M words mostly from English-speaking preschoolers' vocabulary.[6] The documents are mainly collected from reading comprehension tests for Chinese High-school students. Compared to OntoNotes 5.0 and ACE05, the vocabulary of PreCo is much simpler, and models can generally achieve higher average F_1 score. Similar to ACE05, singleton mentions are also included. Although Preco contains more than ten times of documents than OntoNotes 5.0, it has been less evaluated by recently proposed models.

Datasets for Evaluation In this thesis, we mainly evaluate our proposed methods on the OntoNotes 5.0 English shared task and ACE 2005 English dataset. In order to test the generality of our methods across different languages, the OntoNotes 5.0 Chinese dataset is also used in evaluation.

2.3 Evaluation Metrics

Three popularly used evaluation metrics (MUC, B-CUBED, CEAF) of the general coreference resolution task are discussed in this section, including their advantages and disadvantages.

A coreference cluster is defined as C, and $|C|$ is the number of mentions in the cluster. The term key entities (clusters) refer to gold coreference clusters, while response entities

[6] https://preschool-lab.github.io/PreCo/

(clusters) refer to predicted entities. $K(d)$ and $S(d)$ are the set of gold and predicted coreference clusters, respectively. $|K(d)|$ and $|S(d)|$ represent the number of gold and predicted clusters in a single document, respectively.[7]

2.3.1 MUC

MUC is a coreference link based metric. It computes the minimum number of insertions and deletes required to make predicted coreference clusters identical to gold ones. The intersection between gold and predicted clusters is defined as:

$$P\left(S_j\right) = \left\{C_j^i : i = 1, 2, \ldots, |K(d)|\right\} \tag{2.1}$$

where C_j^i is $S_j \cap K_i$. Then the common links between gold and predicted clusters are defined as:

$$c(K(d), S(d)) = \sum_{j=1}^{|S(d)|} \sum_{i=1}^{K(d)|} w_c\left(C_j^i\right) \tag{2.2}$$

$$w_c(C_j^i) = \begin{cases} 0 & |C_j^i| = 0 \\ |C_j^i| - 1 & |C_j^i| > 0 \end{cases} \tag{2.3}$$

$$s(S(d)) = \sum_{i=1}^{|S(d)|} (|S_i| - 1) \tag{2.4}$$

$$k(K(d)) = \sum_{i=1}^{K(d)|} (|K_i| - 1) \tag{2.5}$$

where $w_c(C_j^i)$ represents the number of coreference links in the intersection set of key clusters K_i and response clusters S_j. $k(K(d))$ and $s(S(d))$ are the number of links in the gold and predicted coreference clusters, respectively.

Therefore, the precision and recall can be computed as:

$$\text{Precision} = \frac{c(K(d), S(d))}{s(S(d))} \tag{2.6}$$

[7] Most notations used in this section are based on Stylianou and Vlahavas (2021).

$$\text{Recall} = \frac{c(K(d), S(d))}{k(K(d))} \tag{2.7}$$

It is evident that the MUC metric ignores clusters that only contain one mention, and it can not be trusted on a dataset where singleton entities exist.

2.3.2 B-CUBED

B^3 is a mention-based metric. The final precision/recall is calculated based on the precision/recall of individual mentions. For each mention m in the key coreference cluster, B^3 computes how many correct mentions are included in the mention clusters of response cluster:

$$\text{Precision(m)} = \frac{w_c(C_j^i)}{w_s(S_j)} \tag{2.8}$$

$$\text{Recall(m)} = \frac{w_c(C_j^i)}{w_k(K_i)} \tag{2.9}$$

where $w_c(C_j^i) = |C_j^i|$, $w_k(K_i) = |K_i|$ and $w_s(S_j) = |S_j|$. The final precision/recall are the weighted sum of all mentions' individual precision/recall.

Since it is calculated based on mentions rather than coreference links, B^3 has a significant flaw. That is, if a gold mention exists in the response cluster, it will be considered correct no matter whether it is included in the correct coreference cluster of response clusters.

2.3.3 CEAF

Constrained Entity Alignment F-measure (CEAF) (Luo, 2005) assumes that each key coreference cluster should only refer to one response coreference cluster. It uses the Kuhn-Munkres algorithm to find the best mapping from the key clusters to the response clusters (g^*) based on their similarity score computed by a similarity function (Φ):

$$\Phi(g) = \sum_{K_i \in K_{min}(D)} \Phi(K_i, g(K_i)) \tag{2.10}$$

where $g(K_i) = S_j$ represents the mapping from K_i to S_j. Φ is the measure implying the similarity between coreference cluster K_i and S_j. For example, $\Phi(K, S)$ represents the number of shared mentions between K and S and $\Phi(K, K)$ is the number of mentions in coreference cluster K.

Then the precision/recall equations can be defined as:

$$\text{Precision} = \frac{\Phi(g^*)}{\sum_{j=1}^{|S(d)|} \Phi(S_j, S_j)} \tag{2.11}$$

$$\text{Recall} = \frac{\Phi(g^*)}{\sum_{i=1}^{|K(d)|} \Phi(K_i, K_i)} \tag{2.12}$$

where g^* is the collection of key clusters in the optimal mapping.

There are many different similarity functions proposed by Luo (2005):

$$\Phi_1(K, S) = \begin{cases} 1, & K = S \\ 0, & \text{otherwise} \end{cases} \tag{2.13}$$

$$\Phi_2(K, S) = \begin{cases} 1, & K \cap S \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \tag{2.14}$$

$$\Phi_3(K, S) = |K \cap S| \tag{2.15}$$

$$\Phi_4(K, S) = \frac{2|K \cap S|}{|K| + |S|} \tag{2.16}$$

Equation (2.13) considers that two coreference clusters are identical if all mentions are the same, while (2.14) insists that two clusters are the same if their intersection is not null. However, these two similarity measurements are not discriminative since they are not able to measure how similar two coreference clusters are. In contrast, the last two equations are clearer to measure the degree to which two coreference clusters are similar. (2.15) considers the size of the intersection set of two coreference clusters, whereas (2.16) is the mention F-measure between predicted and gold coreference clusters, which is used in the implementation of the official evaluation scripts of the CoNLL-2012 shared task.

Considering the cons and pros of these three evaluation metrics, the best practice is to use the average of their F1 scores as the final model performance. This practice also allows fair comparison among different coreference resolution models on standard benchmark

datasets. [8]

2.4 Rule-based Coreference System

Like many methods in NLP literature, coreference systems of the early days heavily relied on hand-crafted rules derived from heuristics or syntactic and semantic features extracted from document texts. In this section, we specifically discuss several classic and important rule-based models.

2.4.1 Hobb's Algorithm

Hobb's naive algorithm (Hobbs, 1978) can arguably be one of the first algorithms proposed to deal with coreference resolution. It designed a bunch of complicated rules to traverse the constituent parse tree of sentences to resolve the referent of a given mention span in a left-to-right and breadth-first manner. Another heuristic is the use of selectional preference to eliminate impossible candidates. The antecedent search space is pruned using the selectional constraints, and the algorithm stops when the space converges to a single antecedent. Unlike models proposed nowadays, which are automatically evaluated using large datasets and well-designed evaluation metrics, it was manually evaluated on different datasets like news and magazines.

2.4.2 Centering Theory

Unlike Hobb's algorithm, which utilises syntactic features, centering theory (Grosz et al., 1995) attempts to exploit the discourse properties of coreferences. The centering theory defines the center as an entity that links several utterances or sentences. The forward-looking center refers to a list of entities mentioned in a single utterance. In contrast, the backward-looking center is defined as the intersection of forward-looking centering of the current and preceding sentence, generally as the highest-ranking entity of the preceding sentence realised in the current sentence.

[8] Please note that for the evaluation on datasets with singleton mentions annotated, B^3 and CEAF can handle singletons by simply treating singletons as coreference clusters containing only one mention.

The algorithm first constructs all possible pairs of forward-looking and backward-looking centers, which are denoted as anchors. Then it uses several defined rules to prune impossible anchors. There are three main phases in center identification: center continuation means that two neighbouring utterances share the same center; center retaining indicates a possible shift between centers; center shifting implies the change from one center to another center in the next utterance. Meanwhile, another two rules are also applied: Rule1 states that the center entity has the highest probability of being pronominalized, and Rule2 requires that center continuation is preferred than center retaining is preferred than center shifting.

2.4.3 Multi-Pass Sieve System

Raghunathan et al. (2010) claims that a function that utilises a set of hand-engineered features can lead to inferior performance, for the reason that high-precision features can be overwhelmed by low-precision features. The multi-pass sieve model organises deterministic rules as a pyramid, where rules are first ordered descendingly by their corresponding precision scores and then applied in that order. Each sieve's input is the cluster output from the preceding sieve. Important features are guaranteed to have higher priority. It comprises two phases: the first phase is responsible for extracting, sorting and pruning mentions using myriad constraints based on parse structures, whereas the second phase is in charge of filtering impossible candidates using multi passes such as head matching, string matching and gender, animacy and number agreement.

2.4.4 Discussion

Coreference models in the early days explored this task broadly by exploiting many different kinds of features, including discourse structures and syntactic and semantic features, providing numerous advice for the development of statistical and deep learning models. However, these rule-based systems were typically evaluated on different datasets and evaluation metrics, making the comparison of their performance inconsistent. Moreover, the reliance on hand-designed rules hinders their generalization ability, resulting in the phenomenon that an algorithm performs well in one domain but fails in others.

2.5 Statistical and Machine Learning Coreference System

With the availability of tagged coreference datasets, learning-based and statistical models have become prevalent and they also outperformed their rule-based counterparts. Learning-based models are normally categorised into three categories, namely mention-pair models, mention-ranking models and entity-mention models. We briefly discuss each of these model types in this section.

2.5.1 Mention-Pair Model

Mention-pair models are binary classifiers determining whether a pair of mentions are coreferent without depending on other pairs. The mention pair model consists of three independent phases, and the improvement on one phase does not necessarily indicate the improvement of the subsequent phase.

The first phase is to create training instances. One strategy is to create a positive instance by taking a mention A1 and its nearest correct antecedent A2. In contrast, the negative instance is constructed by randomly choosing a mention A3 in the document contexts between A1 and A2 (Soon et al., 2001). Another constraint (Ng and Cardie, 2002b) is that if A1 is a pronoun, A2 cannot be chosen unless it is non-pronominal.[9]

The second phase is model training. Many popular learning algorithms such as decision trees and random forests (Aone and William, 1995; McCarthy and Lehnert, 1995) were widely used as a binary classifier.

The final phase is the generation of entity clusters. Many clustering methods have been proposed such as best-first (Ng and Cardie, 2002b) and closest-first (Soon et al., 2001) methods. The closest-first clustering considers that all previous mentions of each mention are processed in a right-to-left manner. If a candidate is classified as true, it is regarded as correct and others are discarded. In other words, it will choose the closest candidate that is predicted as true. By contrast, the best-first method chooses the mention classified as true but with the largest predicted score.

However, decision making is limited in two compared mentions and can fail when a mention is ambiguous. For example, consider a document which contains three mentions:

[9]Deciding whether a pronoun is the antecedent of another pronoun can be comparatively harder since they carry limited semantic information. Such examples sometimes can be even too difficult for humans.

Obama, *Mr. Obama* and *she*. A mention-pair model may links *Obama* with *Mr. Obama* due to string matching and determines that *Obama* is coreferent with *she* because of proximity. Under this case, this model cannot cluster these three mentions for violating gender agreement. But transitivity implies that *Mr. Obama* and *she* will end up being grouped into the same cluster. This kind of error is mainly caused by the assumption that coreference decisions are independent: earlier decisions about pairs of mentions cannot inform later ones.

Another line of research in this stage focuses on whether determining anaphoricity explicitly is necessary (Ng and Cardie, 2002a). Anaphoricity determination is the task of determining whether a given mention is anaphoric or not.[10] The motivation behind this idea is that compared to coreference resolution, which aims to find the correct antecedent for a given mention, determining whether a mention is anaphoric or not is much easier. Besides, anaphoricity determination can vastly simplify the coreference resolution task, as only mentions that are predicted to be anaphoric should be resolved further. The common practice is to independently train an anaphoricity prediction model and use it as a component for the coreference model in a pipeline manner. Meanwhile, many studies also confirm that syntax both in the form of dependency graph and constituent trees brings benefits in determining anaphoricity (Kong et al., 2010; Kong and Zhou, 2011). The potential risk of such a method is that inaccurate anaphoricity determination can hurt coreference resolution performance because of the cascaded errors from the anaphoricity component to the coreference component. Therefore, although anaphoricity determination could simplify coreference resolution, early approaches do not explicitly model anaphoricity. They instead consider a mention as non-anaphoric if no antecedent mentions are selected for it when building coreference chains, which is also widely used by neural models nowadays.

2.5.2 Mention-Ranking Model

The Mention-ranking model explicitly ranks all antecedents of a mention and chooses the one with the highest score as reference. This kind of model is first proposed by Yang et al. (2003) and Iida et al. (2003) to do pairwise ranking. This method was later improved

[10] Anaphoric means a mention is coreferent with a preceding mention (preceding mentions means mentions that appear in preceding contexts of a given mention), while non-anaphoric means a mention is not coreferent with any preceding mentions and starts an entirely new coreference cluster.

by Denis and Baldridge (2008) to use the softmax function to rank all previous mention candidates, which almost every modern neural mention-ranking model adopts. Ng (2005) proposed to do mention detection and antecedent linking jointly in an end-to-end manner. A dummy token is used to represent antecedents for non-anaphoric mention spans. That is, for a given mention i, a list of associated random variables $\mathcal{Y}_i = \{\epsilon, 1, \ldots, c-1\}$ is used to represent all its possible antecedents, and the final decision is based on the ranking over all variables by choosing the one with the highest score.

2.5.3 Entity-Based Model

The motivation of the entity-mention model is the incapability of the mention-pair model in modelling global features (only local features derived from two compared mentions are employed). Sometimes the limited information extracted from two mentions can not provide sufficient signals for coreference reasoning, especially when the compared candidate antecedent is a pronoun or lack of discriminative information such as gender. Therefore, various machine learning entity-based models have been proposed over the past decades.

The entity-mention model is the entity-based version of the mention-pair model, which generally focuses on determining whether a mention is coreferent with previously-formed mention clusters (Rahman and Ng, 2009). It improves over mention-pair models by utilising past coreference decisions to inform future ones. It focuses on whether it should assign a mention span to a previously-formed entity cluster or create a new entity cluster. Traditional feature-based entity models normally extracted features over clusters. Features such as cluster size and shapes, which are the types of mentions within the cluster (Björkelund and Kuhn, 2014) were used. Entity-mention model can also build upon mention-pair models by aggregating the mention-pair probabilities over all mention-pairs in two clusters (Clark and Manning, 2015).

The cluster-ranking model (Rahman and Ng, 2009) extends the mention-ranking model by ranking existing clusters instead of candidate mentions, and they have been confirmed to outperforms their entity-mention counterparts. Agglomerative clustering method (Culotta et al., 2007; Stoyanov and Eisner, 2012) is another line of research for entity-based models. In this framework, each mention is initialized as a single cluster, and the learned model chooses to merge two best clusters in each iteration by exploiting cluster-level features.

However, although entity-based models enable us to utilise information on cluster level,

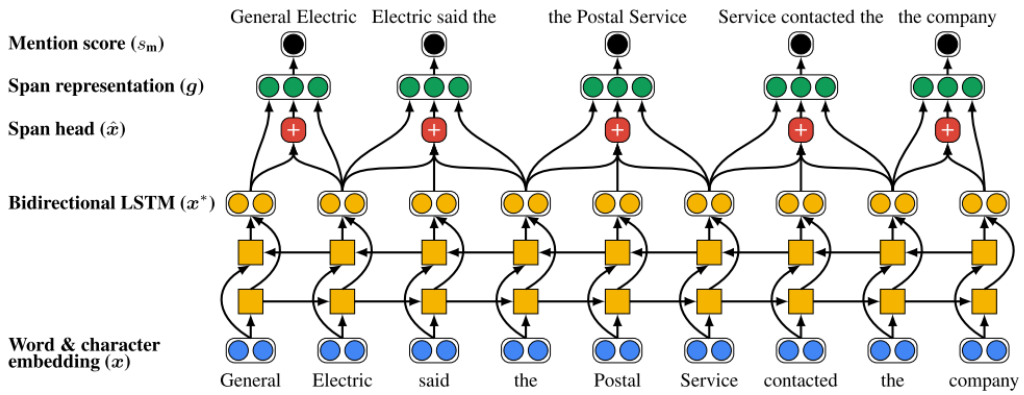

Figure 2.2: The first stage of the end-to-end coreference model generates mention span representations and computes scores for each mention (Figure from Lee et al. (2017)).

it does not lead to significant performance gains than mention-ranking models. The incremental processing way is also less efficient than mention-ranking models and cannot be easily parallelized.

2.6 Neural Coreference System

The classification of neural coreference models is almost the same as traditional learning-based models. But the dependence on hand-designed features has been significantly reduced. In this section, we discuss some recently proposed neural mention-ranking models.

2.6.1 End-to-End Neural Coreference Resolution

In this section, we describe the neural mention-ranking model proposed by Lee et al. (2017). It is the first end-to-end neural mention ranking model without a separate mention detection component. Instead, it enumerates all possible text spans up to a certain length limit as candidate mentions. Moreover, its dependence on hand-engineered features has been significantly reduced and only a few hand-crafted features such as span width are included, which can be easily obtained. We refer to this as E2E-COREF in this thesis, and it also forms the basis of our proposed methods and is a key baseline in evaluation.

Task Formulation The task aims to assign an antecedent y_i to each span i, choosing from a sequence of candidate variables $\mathcal{Y}_i = \{\epsilon, 1, \ldots, c-1\}$, where ϵ is a dummy symbol indicating that mention i is non-anaphoric or it is the first appearance of an entity.

For each pair of span i and j, the model assigns a score $s_c(i, j)$ to measure the likelihood of mention j being the antecedent of mention i. Therefore, for each mention i, the system selects the mention j with the highest score from $\mathcal{Y}_i$.

Model Architecture As shown in Figure 2.2, the input text is first transformed into a sequence of continuous vectors $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_t\}$ by combining pretrained word embeddings (Mikolov et al., 2013; Pennington et al., 2014) and CNN-based or BiLSTM-based character embeddings (Turian et al., 2010). Next it is processed by a one-layer BiLSTM (Hochreiter and Schmidhuber, 1997) to form contextualized features:

$$\overleftarrow{\mathbf{h}}_t = \mathbf{LSTM}^{\text{forward}}(\overleftarrow{\mathbf{h}}_{t-1}, \mathbf{w}_t) \tag{2.17}$$

$$\overrightarrow{\mathbf{h}}_t = \mathbf{LSTM}^{\text{backward}}(\overrightarrow{\mathbf{h}}_{t+1}, \mathbf{w}_t) \tag{2.18}$$

$$\mathbf{h}_t = [\overleftarrow{\mathbf{h}}_t; \overrightarrow{\mathbf{h}}_t] \tag{2.19}$$

Then, each span's representation is a concatenation of start and end tokens of the span, the head words obtained by using attention mechanism among all words within the span:

$$\alpha_t = \mathbf{FFNN}(\mathbf{h}_t) \tag{2.20}$$

$$\beta_{i,t} = \frac{\exp(\alpha_t)}{\sum_{k=\text{Start}(i)}^{\text{End}(i)} \exp(\alpha_k)} \tag{2.21}$$

$$\mathbf{h}_{\text{Head}(i)} = \sum_{t=\text{Start}(i)}^{\text{End}(i)} \beta_{i,t} \cdot \mathbf{h}_t \tag{2.22}$$

where **FFNN** represents a two-layer feedforward neural network with the **ReLU** activation function inside.

Therefore, each span i is represented as a vector $\mathbf{g}_i$ defined as:

$$\mathbf{g}_i = [\mathbf{h}_{\text{Start}(i)}, \mathbf{h}_{\text{End}(i)}, \mathbf{h}_{\text{Head}(i)}, \phi(i)] \tag{2.23}$$

where $\phi(i)$ is the embedded span width.

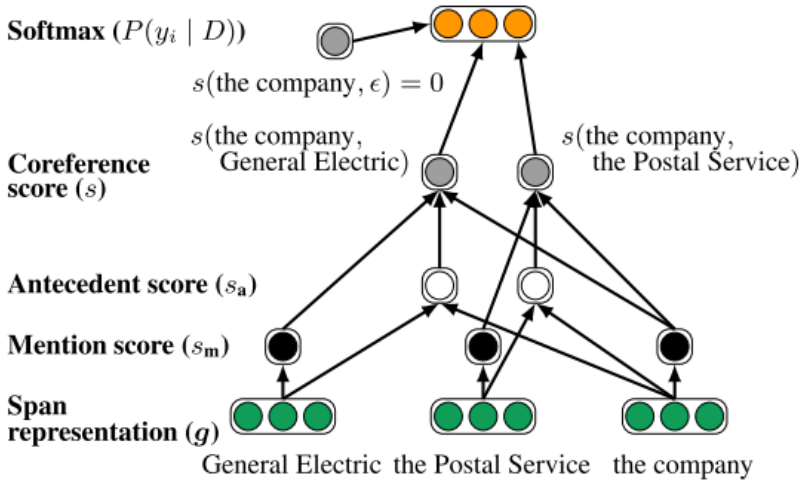

Figure 2.3: In the second stage, mention score is used for pruning unlikely candidate mentions and antecedent score is computed for each pair of mentions. Final coreference scores are obtained by summing the mention and antecedent scores of each mention pair (Figure from Lee et al. (2017)).

By comparing each span with other possible antecedents, the model chooses the one with the highest score. As shown in Figure 2.3, the final score is factored into two parts: mention score $s_m(i)$ and coreference score $s_c(i, j)$, which measure how likely span i and j comprise valid mentions and corefer to one another, respectively:

$$s_m(i) = \textbf{FFNN}_m(g_i) \tag{2.24}$$

$$s_c(i, j) = \textbf{FFNN}_c([\mathbf{g}_i, \mathbf{g}_j, \mathbf{g}_i \odot \mathbf{g}_j, \phi(i, j)] \tag{2.25}$$

$$s(i, j) = s_m(i) + s_m(j) + s_c(i, j) \tag{2.26}$$

where $\odot$ is element-wise multiplication, whereas, $\phi(i, j)$ is a feature vector that encodes useful features such as document genre, speaker identities and span distances. Both $\textbf{FFNN}_m$ and $\textbf{FFNN}_s$ are two-layer feedforward networks with **ReLU** function.

To reduce memory consumption and sidestep quardic complexity, the model only keeps a few best antecedent candidates as a function of $0.4T$ (T is the document length) according to their mention scores. Moreover, each mention's maximum number of antecedents is set to 250 to reduce inference time further. Finally, the model can create final clusters for

each entity by choosing the antecedent with the highest score for each mention i and doing transitive closure on the antecedents. Despite the aggressive pruning strategy, it achieved new state-of-the-art performance on the CoNLL-2012 English shared task at that time.

Discussion As the first end-to-end neural coreference resolution model, E2E-COREF reduces the need for a separate mention detection component. Thus, mention detection and coreference resolution are jointly trained in a fully end-to-end manner to reduce cascading errors. Besides, it uses an attention mechanism rather than a syntax parser to find head words. Experimental analysis shows that this method can find head words more accurately than traditional ones, showing its contribution to a better final performance.

However, E2E-COREF has several limitations. Firstly, considering inference efficiency, it aggressively uses distance pruning strategy to limit potential antecedents to the nearest 250 ones, while for some documents, especially in biomedical areas, the distance between two coreferent mentions can be much further. Secondly, due to long documents in the dataset, the model uses independent LSTMs for each sentence. Although it avoids RNN's locality biasing problem, it ignores relations between neighbouring sentences, thereby losing inter-sentence information (Luo and Glass, 2018). Thirdly, at the beginning stage of training, mention pruning is completely random since no mention-detection supervision is provided, resulting in longer training time (Zhang et al., 2018). Lastly, as a mention-ranking model, it does not build global entity-level mention representation, which treats each mention independently and limits the information it can use when resolving coreference.

In this thesis, we propose two novel methods based on the E2E-COREF model. They effectively incorporate dependency and constituent syntax to capture long-range dependencies between words and semantic role labels to have a deep understanding of semantics encoded in document texts (Chapter 3 and 4). Furthermore, we also utilise large pretrained language models (Joshi et al., 2020) to encode documents to obtain contextualized representations and inter-sentence information.

2.6.2 Higher-Order Span Refinement with Coarse-to-Fine Pruning

Lee et al. (2018) improved the E2E-COREF model by building approximated entity-level representation conditioning on higher-order structures and *coarse-to-fine* pruning strategy.

This model is denoted as C2F-COREF in this thesis.

C2F-COREF is improved by including inference with N iterations of refining span representations, denoted as $\mathbf{g}_i^n$ for the representation of span i at iteration n. At each iteration n, the refined span representation is the combination of previous representation $\mathbf{g}_i^{n-1}$ and the corresponding expected antecedent vector by using a gating mechanism, while the expected antecedent vector is defined as the weighted sum among all possible antecedents using the antecedent distribution:

$$P_n(y_i) = \frac{\exp(\mathbf{g}_i^n, \mathbf{g}_{y_i}^n)}{\sum_{y_i \in \mathcal{Y}_i} \exp(\mathbf{g}_i^n, \mathbf{g}_y^n)} \tag{2.27}$$

where $P_n(y_i)$ is the likelihood of mention y_i being the correct antecedent of mention i at iteration n. Then we can use this antecedent distribution P_n to compute the expected antecedent vector $\mathbf{a}_i^n$ for each mention:

$$\mathbf{a}_i^n = \sum_{y_i \in \mathcal{Y}_i} P_n(y_i) \cdot \mathbf{g}_{y_i}^n \tag{2.28}$$

Then the span representation $\mathbf{g}_i^n$ is obtained as the interpolation between $\mathbf{a}_i^n$ and $\mathbf{g}_i^{n-1}$:

$$\mathbf{f}_i^n = \sigma(\mathbf{W}_f[\mathbf{g}_i^{n-1}; \mathbf{a}_i^n)] \tag{2.29}$$

$$\mathbf{g}_i^n = \mathbf{f}_i^n \odot \mathbf{g}_i^{n-1} + (1 - \mathbf{f}_i^n) \odot \mathbf{a}_i^n \tag{2.30}$$

Where σ is the logistic sigmoid function and $\mathbf{f}_i^n$ represents a vector whose value is in $[0, 1]$, controlling to which extent the new information from its attended antecedent should be integrated. The above procedure can be repeated for several times to gradually obtain refined span representation for each mention.

Another improvement is the *coarse-to-fine* inference which uses a bilinear score function to prune impossible antecedents:

$$c(i, j) = \mathbf{g}_i^T \mathbf{W}_c \mathbf{g}_j \tag{2.31}$$

where $\mathbf{W}_c$ is a trainable weight matrix and $c(i, j)$ is the roughly approximated antecedent score measuring the probability of mention i and j being coreferent. Thus, the overall score function is now factored into three parts: mention score, *coarse-to-fine* score and

coreference score:

$$s(i, j) = s_m(i) + s_m(j) + c(i, j) + s_c(i, j) \tag{2.32}$$

In summary, the procedure of C2F-COREF consists of three stages:

1. Keep the best M mentions based on mention score $s_m(i)$ of each mention span i.
2. Keep the best K antecedent of remaining spans based on $s_m(i) + s_m(j) + c(i, j)$
3. Use above described overall score $s(i, j)$ to obtain the best likely antecedent for each mention i.

By using this strategy, C2F-COREF can prune impossible candidates more progressively (decrease from 250 to 50) while improving performance. Apart from these two improvements, it alternatively uses deep contextualized word embeddings (Peters et al., 2018) as input to achieve further improvements. This also implies that information from other sentences and longer-range contexts are important in resolving coreference. Furthermore, the C2F-COREF model has been extended to integrate more powerful pretrained models including BERT (Devlin et al., 2019) and SpanBERT (Joshi et al., 2020) to achieve state-of-art performance (Jiang and Cohn, 2021).

2.6.3 Knowledge Enhanced Variants

Some coreference resolution tasks may require external knowledge. Linguistic features and heuristics can help models utilise some patterns to make decisions easier, while external knowledge enables the model to achieve better performance in interpreting texts of professional areas.

Zhang et al. (2019a) proposed a novel neural model incorporating external linguistic features and knowledge derived from heuristics based on the framework of E2E-COREF. It processes the input document in the same way as C2F-COREF to obtain span representation and calculate the coreference score for each mention pair. Then based on the current mention pair's context information, it selectively extracts relevant information by using a knowledge attention layer to get knowledge scores. Due to a large number of antecedent candidates, the model uses the softmax pruning strategy to eliminate candidates with lower coreference scores, and the final score is factored into the combination of knowledge and coreference scores.

Based on Zhang et al. (2019a), Zhang et al. (2019b) represents external knowledge as knowledge graphs to extract the most relevant knowledge for each mention based on its contexts and knowledge embeddings using an attention mechanism. Apart from linguistic and heuristic features, knowledge bases including commonsense and medical knowledge are used to enable the model to depend on complex knowledge when interpreting texts in professional areas.

Both papers show that external knowledge helps to make some difficult coreference decisions. However, good features always require careful design by experts, and how to properly integrate these features should be well considered. Moreover, they only validated the effectiveness of external knowledge on pronoun resolution. Whether such external knowledge is still helpful for resolving common coreference resolution including proper names has not been explored yet.

2.6.4 Coreference Resolution as Machine Reading Comprehension

Very recently, Wu et al. (2020) proposes to deal with coreference resolution by using machine reading comprehension (MRC) framework (*CorefQA*): For each mention span, a query is generated by using the sentence it resides in, and the corresponding antecedent mention is obtained using the generated query through a span-prediction module.

More specifically, it uses SpanBERT (Joshi et al., 2020) as the backbone to obtain input representations, and the process of long documents are dealt with sliding window. For mention proposal, it uses a feedforward network to compute the likelihood of each text span being mentions and prune those unlikely spans. Then, for each proposed mention candidate, it uses the MRC mechanism to extract relevant antecedents by injecting the generated query and documents into SpanBERT, and the predicted answer is the most likely antecedent.

CorefQA achieves new state-of-the-art performance on the GAP and CoNLL-2012 English datasets, demonstrating the effectiveness of MRC mechanism in coreference resolution. By using this framework, the query for each mention can depend on longer contexts, while on previous models, contexts are limited in two compared mentions. Moreover, the use of MRC enables the model to use transfer learning techniques to pretrain on many other large-sized question answering datasets, which can further improve its generalization ability.

However, this model is limited in some aspects. For each document, the number of detected mentions is still significant after being pruned, and many queries will be generated. Therefore, for each generated query, the whole document context will be fed into the document encoder independently to find their corresponding antecedents. The consequence is that it is memory-intensive and inefficient during both training and inference time since the above procedure should be repeated many times. Moreover, the slow inference speed hinders its utility in real-world settings and real-time scenarios. Besides, the generation of queries should be well designed (e.g., how to generate informative questions), and how to provide useful texts rather than the whole documents should be well evaluated.

2.7 Sequential Neural Coreference Resolution Model

One drawback of the mention ranking models described in the previous section is that they usually require a large amount of memory at both the training and inference stage, hindering their utility in real-world settings. This phenomenon has become even more severe when utilising pretrained language models as document encoders. Recently, there are a variety of models which imitate the reading behaviour of human beings, processing documents linearly and incrementally resolving coreferences using constant memory.

2.7.1 Recurrent Entity Network for Pronoun Resolution

Recently, Liu et al. (2019) proposes a novel architecture based on recurrent entity network (*RefReader*). Unlike mention-ranking models, it processes text incrementally and resolves pronouns on the fly. It uses a fixed-length memory cell to represent each entity. Each cell is a triplet of key, value and salience. It consists of two components: a memory unit responsible for storing and tracking entity states and a recurrent unit controlling memory updates via a set of gates.

Specifically, the recurrent unit controls the update of the current hidden state by combining the previous hidden state and current input via a gating mechanism. The memory unit uses **overwrite**, **update** and **replicate** gates to control memory updates. **Overwrite** represents the probability of writing new entities and removing old entities, **update** controls information updates on stored entities and **replicate** deprecates the salience of old entities when no mention appears. Gate copies at each time step are maintained to build

coreference links for the final decision. Furthermore, the author uses language modelling objective to pretrain the model, and experimental results show that even without explicit golden coreference labels, it implicitly learns latent coreference structures to some extent, which is also found in the analysis of learned attentions of large pretrained language models (Clark et al., 2019).

RefReader sets new state-of-the-art performance on the GAP dataset, and it is the first neural model which resolves pronouns incrementally. However, it does not examine its feasibility in the CoNLL-2012 shared task, where texts are full documents and much more entities should be resolved. Its experimental analysis also shows model performance degrades, and extra memory cells are required to maintain comparable performance when text length increases.

Further attempts have been made to improve the *RefReader* model. Toshniwal et al. (2020) proposed a similar model *PeTra* based on the *RefReader* with simplified memory modules. Similarly, it contains **overwrite** and **coref** gates to track entities stored in memory cells, with much-simplified memory contents. The key-value vectors have been replaced with a simple content vector to represent a memory cell. Furthermore, some restrictions have been applied to improve the *RefReader* model, such as **coref** gates should not open for a memory cell unless it has been updated before. The improvements that *PeTra* achieved against the *RefReader* model verify its effectiveness with much simpler architecture. However, the deficiency of processing long documents still remains a difficult problem for such sequential models.

2.7.2 Coreference Resolution with Constant Memory

Following similar ideas to resolve coreference incrementally, Xia et al. (2020) has successfully recast a memory-intensive mention ranking model (Joshi et al., 2019) into a sequential model, consuming constant memory with respect to document length. Similar to the C2F-COREF model, it first proposes a set of candidate mention spans. Then for each mention, it will compute the similarity scores against the embeddings of existing clusters. These spans are then used to either update the cluster embeddings (if refer to one of the existing entity clusters) or create a new entity cluster. Furthermore, only a set of salient entities are maintained in the memory, and expired entities are removed from memory and will never be revisited. The salience of entities is mainly decided by the cluster size and the distance

between the current context and the last appearance of spans in each cluster. Their model has been successfully converted from a fully-trained high-performing model with a slight performance drop, reducing memory usage with constant space complexity in regards to document length. However, such sequential models still rely on mention-ranking models that have been fully trained. Our experiments also show that training such sequential models from scratch on the same dataset could lead to significant performance degradation. Nevertheless, the success of converting a memory-intensive mention-ranking model to a sequential model with constant memory may give us new research directions: trying to improve mention-ranking models as much as possible and reducing it to a sequential model using similar ways to increase its utility in real-world settings and real-time scenarios.

Chapter 3

Incorporating Syntax and Semantic Roles

3.1 Introduction

In recent years, impressive progress has been made since the introduction of the first end-to-end neural coreference resolution model (Lee et al., 2017) by utilising contextualized embeddings from large pretrained language models such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019), with an improvement of 15.9% over recent three years (Joshi et al., 2019, 2020; Kantor and Globerson, 2019; Xu and Choi, 2020; Wu et al., 2020). Rich language knowledge encoded in these pretrained models has largely alleviated the need for syntactic and semantic features. However, such information has been shown to benefit a series of BERT-based models on other tasks (Nie et al., 2020a; Wang et al., 2020; Pouran Ben Veyseh et al., 2020). Therefore, we believe such information could also benefit the coreference resolution task.

In this chapter, we propose a neural coreference resolution model based on Joshi et al. (2019) (§2.6.1 and §2.6.2), which we extend by incorporating external syntactic and semantic information. For syntactic information, we use dependency trees to capture the long-term dependency that exists among mentions. Kong and Jian (2019) has successfully incorporated structural information into neural models, but their model still requires the design of complex hand-engineered features. In contrast, our model is more flexible, using

a graph neural network to encode syntax in the form of dependency trees. For semantic information, we adopt semantic role labelling (SRL) structures. SRL labels capture *who did what to whom*, and it effectively provides document-level event description information, which allows us to identify the relationship between mentions better. Previous statistical coreference systems have successfully integrated such information (Ponzetto and Strube, 2006a; Kong et al., 2009), but their effectiveness has not been examined in neural models.

Moreover, by drawing inspiration from recent progress made in document-level relation extraction task (Christopoulou et al., 2019), we encode both syntactic and semantic information in a heterogeneous graph. Nodes of different granularity are connected based on the feature structures. Node representations are updated iteratively through our defined information propagation mechanism and incorporated into contextualized embeddings using an attentive integration module and gating mechanism. We conduct experiments on the OntoNotes 5.0 (Pradhan et al., 2012) benchmark and ACE 2005 dataset (Walker and Consortium, 2005), where the results show that our proposed model significantly outperforms a strong baseline.[1]

This chapter is organized as follows. We first discuss related work in Section 3.2, and then we briefly review the baseline model where our proposed model was based (§3.3). Next, we introduce our proposed model in Section 3.4, including the construction of syntactic and semantic graph and the design of the information propagation mechanism. We then describe the learning objective of our model in Section 3.5. Experiments and results are presented in Section 3.6. Besides, we also analyse the behaviour of our proposed model in detail in Section 3.7 and 3.8. Finally, we conclude and discuss future work in Section 3.9.

3.2 Related Work

Coreference Resolution Coreference resolution is a core task in NLP, which aims to identify all *mentions* that refer to the same entity. With the introduction of the first end-to-end coreference resolution model (Lee et al., 2017), the coreference resolution task has been dominated by neural end-to-end based models. Moreover, with the help of large pre-trained language models, neural coreference resolution models have achieved impressive

[1]Code is available at: https://github.com/Fantabulous-J/coref-HGAT

improvements on the standard evaluation benchmark dataset (Pradhan et al., 2012). Recently, Wu et al. (2020) (CorefQA) chose to formulate the coreference resolution task as a span-prediction problem by utilising machine reading comprehension framework. With the help of transfer learning, they successfully pretrain their model on large QA datasets such as SQuAD (Rajpurkar et al., 2016), achieving current state-of-the-art performance. However, training the CorefQA model requires intensive GPU memory and the inference speed is not ideal because of the repeated QA procedure to find correct antecedents for each proposed mention candidate. In this chapter, we do not adopt the CorefQA as a start baseline for hardware concerns but choose another lightweight baseline (Joshi et al., 2019) to evaluate the utility of external syntax and semantics for the coreference resolution task.

Incorporating External Features using Graph Neural Network Graph Neural Networks (GNN) have long been used for integrating external features of graph structures into a range of Natural Language Understanding and Generation tasks, including semantic role labelling (Marcheggiani and Titov, 2017) and machine translation (Bastings et al., 2017). However, the utilisation of GNN on coreference resolution task is less explored. Xu and Yang (2019) adopted dependency syntax to improve gendered pronoun resolution. However, they did not evaluate their model on larger datasets such as OntoNotes and identify whether syntax features are still helpful for common coreference resolution such as noun phrases. In this chapter, we utilise not only syntax features but also semantic features. We show that both of them contribute to significant improvement over a strong baseline on a large standard dataset and a dataset of smaller size.

There are a variety of GNN variants. Graph Convolutional Network (GCN) (Kipf and Welling, 2017) is the most widely-used one and has been shown to benefit a number of NLP tasks in integrating external features. However, it lacks the ability to model different edge labels, including directions and edge types. Although Relational Graph Convolutional Network (RGCN) (Schlichtkrull et al., 2017) was proposed to tackle this problem, the way of representing edge information as label-wise parameters makes it suffer from the over-parameterisation problem even for small-sized label vocabularies. In this work, we use a graph encoder improved based on Graph Attention Network (GAT) (Veličković et al., 2018) to capture structural syntax and semantics better, as GAT performs better in handling sparse graphs and can model different types of edges with few parameters.

BERT for Coreference Resolution Large pretrained language models (Peters et al., 2018; Devlin et al., 2019; Joshi et al., 2020) have largely driven the advances in coreference resolution and other NLP tasks in recent two years. Joshi et al. (2019) improves the C2F-COREF model by replacing the BiLSTM-based contextualized document encoder with BERT to effectively model long-range dependencies. The performance on OntoNotes has been further improved by utilising SpanBERT (Joshi et al., 2020), which introduces the random span masking and span boundary detection pretraining tasks to obtain better span representations. In this chapter, we use SpanBERT as the document encoder since it has demonstrated its superior effectiveness on span-based tasks.

3.3 Brief Overview of Baseline Model

Our model is based on the C2F-COREF model (Lee et al., 2018) , which is described in detail in §2.6.2. Here we have a quick recap of its component. It enumerates all text spans as potential mentions and prunes unlikely spans aggressively. For each mention i, the model will learn a distribution over its possible antecedents $\mathcal{Y}(i)$:

$$P(y) = \frac{e^{s(i,y)}}{\sum_{y' \in \mathcal{Y}(i)} e^{s(i,y')}} \tag{3.1}$$

where the scoring function $s(i, j)$ measures how likely span i and j comprise valid mentions and corefer to one another:

$$s(i, j) = s_m(i) + s_m(j) + s_c(i, j) \tag{3.2}$$

$$s_m(i) = \mathbf{FFNN}_m(\mathbf{g_i}) \tag{3.3}$$

$$s_c(i, j) = \mathbf{FFNN}_c(\mathbf{g_i}, \mathbf{g_j}, \phi(i, j)) \tag{3.4}$$

where $\mathbf{g_i}$ and $\mathbf{g_j}$ are span representations formed by the concatenation of contextualized embeddings of span endpoints and head vector using attention mechanism. $\mathbf{FFNN}$ represents the feedforward layer, $\phi(i, j)$ are meta features including span distance and speaker identities, and s_m and s_c are the mention score and pairwise coreference score. We do not use the higher-order inference module to get the refined span representation using antecedent distribution as an attention mechanism since a recent study (Xu and Choi, 2020)

shows that it fails to boost performance.

3.4 Proposed Model

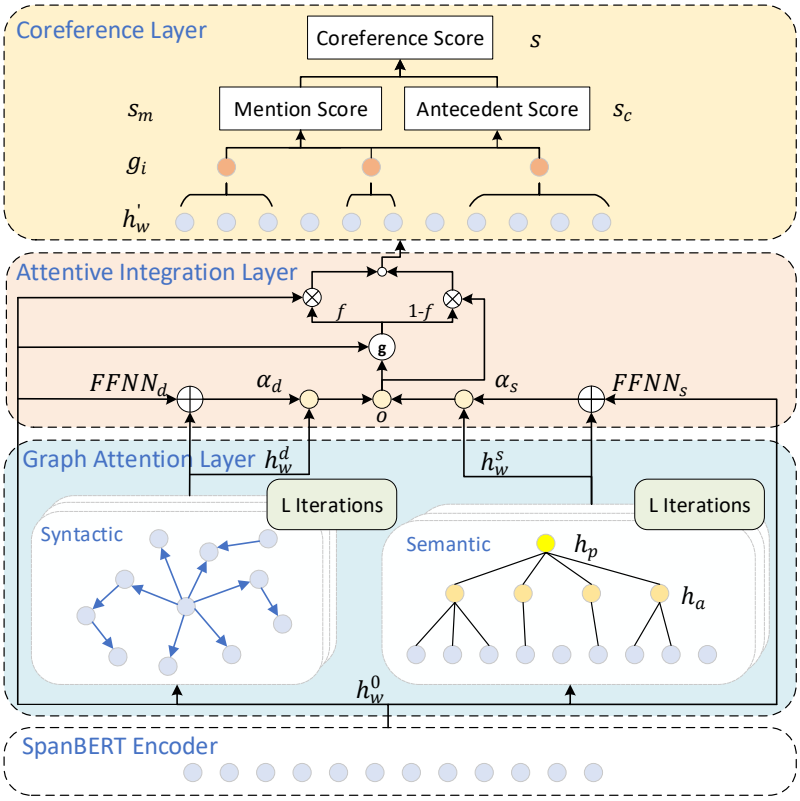

Figure 3.1: The overall architecture of our proposed model. Firstly, a document is encoded by SpanBERT to get initial token representations, which are then enhanced by the syntactic and semantic graph to learn rich global information, respectively. Next, an attentive integration layer is employed to infuse enhanced token representation dynamically. Finally, enhanced token representations are utilised to form span embeddings and compute pairwise coreference scores.

Figure 3.1 shows the architecture of our proposed model, where the key components are presented in blue and orange backgrounds. Other parts follow Lee et al. (2018) (see §3.3) except that we use SpanBERT (Joshi et al., 2020) as the document encoder.

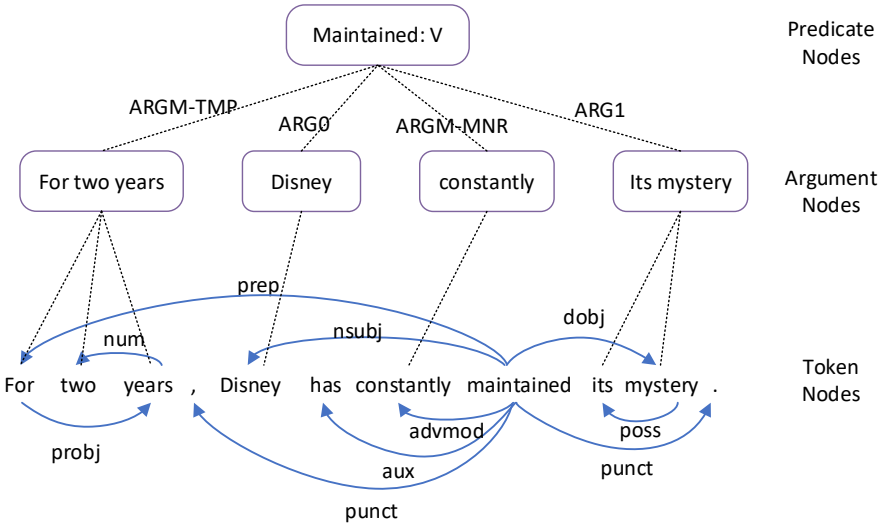

Figure 3.2: An example of our proposed Syntax and Semantics-based Heterogeneous Graph.

3.4.1 Node Construction

There are three types of nodes in our heterogeneous graph: token nodes (T), argument nodes (A) and predicate nodes (P). The representation of token nodes and predicate nodes is the contextualized embeddings from the SpanBERT encoder, denoted as $\mathbf{h}_w$ and $\mathbf{h}_p$ respectively. The representation of an argument node is formed by averaging the embeddings of tokens it contains, denoted as $\mathbf{h}_a$.

3.4.2 Edge Construction

Graph edges are constructed based on feature structures. An example is shown in Figure 3.2 to illustrate the structure of our heterogeneous graph.

Token-Token (TT) Edges are constructed according to dependency tree structures. Specifically, there will be a directed edge between two token nodes starting from head to dependent if connected, with edges being the corresponding dependency labels. A self-loop edge with *cyclic* label is also added to each node in the graph. Besides, we also link the root

nodes of two adjacent sentences to allow cross-sentence interaction.

However, having SpanBERT work with word-level dependency syntax is one challenge because it tokenizes documents into wordpiece units. To resolve this inconsistent issue, we use a simple alignment procedure to map word-level syntax sequence into wordpiece sequence in SpanBERT: if there is an edge between word i and j, then we assign the same edge between any subtoken in word i and any subtoken in word j.

Token-Argument (TA) Argument nodes are linked to token nodes they contain. The edge is unlabelled but bidirectional to allow token-level information to augment the averaged representation of arguments and propagate semantic information back to tokens.

Predicate-Argument (PA) Argument nodes are connected to predicate nodes they belong to with edges being the corresponding SRL labels. The edge is made bidirectional to allow mutual information propagation. Predicates can be regarded as intermediate nodes to allow each argument to aggregate information from other arguments with the same predicate.

3.4.3 Graph Attention Layer

We use a Graph Attention Network (Veličković et al., 2018) to propagate syntactic and semantic information to basic token nodes. For a node i, the attention mechanism allows it to selectively incorporate information from its neighbour nodes $\mathcal{N}_i$:

$$\alpha_{ij} = \text{softmax}(\sigma(\mathbf{a}^T[\mathbf{W}\mathbf{h}_i; \mathbf{W}\mathbf{h}_j; \mathbf{e}_{ij}])) \tag{3.5}$$

$$\mathbf{h}'_i = \|_{k=1}^{K} \text{ReLU}(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \mathbf{h}_j) \tag{3.6}$$

where $\mathbf{h}_i$ and $\mathbf{h}_j$ are embeddings of node i and j, $\mathbf{a}^T$, $\mathbf{W}$ and $\mathbf{W}^k$ are trainable parameters. $\mathbf{e}_{ij}$ is edge label embedding between node i and j, σ is the **LeakyReLU** activation function. $\|$ and $[;]$ represent the concatenation operation. Eqs. 3.5 and 3.6 are designated as an operation, which will be used in next section for simplicity:

$$\mathbf{h}'_i = \text{GAT}(\mathbf{h}_i, \mathbf{h}_j) \tag{3.7}$$

where $\mathbf{h}_i$ and $\mathbf{h}_j$ are the embeddings of target and neighbour node and $\mathbf{h}'_i$ is the updated embedding of target node.

3.4.4 Information Propagation

To make each node embedding more informative, we update all nodes in the graph multiple times through our designed information propagation path. First, we update token nodes using its neighbour token nodes connected through dependency syntactic edges:

$$\mathbf{h}_w^l = \mathrm{GAT}(\mathbf{h}_w^{l-1}, \mathbf{h}_w^{l-1}) \tag{3.8}$$

where $\mathbf{h}_w^{l-1}$ is the token representation in previous layer $l-1$, $\mathbf{h}_w^l$ is the updated representation in current layer l and $\mathbf{h}_w^0$ is the SpanBERT encoding.

In parallel, we update the argument using the token representation; then the updated argument is used to update the predicate features; after that, the updated predicate nodes propagate information back to their connected argument nodes; finally, the updated argument nodes distribute the representation to all connected basic token nodes:

$$\mathbf{h}_a^l = \mathrm{GAT}(\mathbf{h}_a^{l-1}, \mathbf{h}_w^{l-1}) \tag{3.9}$$

$$\mathbf{h}_p^l = \mathrm{GAT}(\mathbf{h}_p^{l-1}, \mathbf{h}_a^l) \tag{3.10}$$

$$\mathbf{h}_a^l = \mathrm{GAT}(\mathbf{h}_a^l, \mathbf{h}_p^l) \tag{3.11}$$

$$\mathbf{h}_w^l = \mathrm{GAT}(\mathbf{h}_w^{l-1}, \mathbf{h}_a^l) \tag{3.12}$$

After L iterations, we can get the final syntax and semantic-enhanced token representation, which can be denoted as $\mathbf{h}_w^d$ and $\mathbf{h}_w^s$, respectively.

3.4.5 Attentive Integration Layer

Since attention mechanisms are effective in choosing the most relevant information (Nie et al., 2020a,b), we use an attentive integration layer to incorporate the syntactic and semantic information selectively. For each type of information $\mathbf{h}_w^c \in \{\mathbf{h}_w^d, \mathbf{h}_w^s\}$, we concatenate it with initial token representation $\mathbf{h}_w^0$ and use the concatenation to compute the importance

score of $\mathbf{h}_w^c$ to $\mathbf{h}_w^0$:

$$\alpha_c = \text{softmax}(\mathbf{FFNN}_c([\mathbf{h}_w^0; \mathbf{h}_w^c])) \tag{3.13}$$

where $\mathbf{FFNN}_c$ is a one-layer feedforward network with sigmoid activation function for information type c (either Dep or SRL). After obtaining the valid attention weights using the softmax function, we could compute the weighted average sum of both syntactic and semantic information:

$$\mathbf{o} = \sum_{c \in \{d,s\}} \alpha_c \mathbf{h}_w^c \tag{3.14}$$

Since the extra syntactic and semantic information is not always useful, we use a gate to leverage such information dynamically[2]:

$$\mathbf{f} = \sigma(\mathbf{W}_g \cdot [\mathbf{h}_w^0; \mathbf{o}] + \mathbf{b}_g) \tag{3.15}$$

$$\mathbf{h}'_w = \mathbf{f} \odot \mathbf{h}_w^0 + (1 - \mathbf{f}) \odot \mathbf{o} \tag{3.16}$$

where $\mathbf{W}_g$ and $\mathbf{b}_g$ are trainable parameters, $\odot$ means element-wise multiplication and σ is the logistic sigmoid function.

Finally, the augmented token representation $\mathbf{h}'_w$ can be used to form span representation and compute pairwise coreference score as in §3.3.

3.5 Model Learning

The objective function of our model consists of two parts: the cluster loss, $\mathcal{L}_{cluster}$, and the mention detection loss, $\mathcal{L}_{mention}$:

$$\mathcal{L} = \mathcal{L}_{cluster} + \lambda \mathcal{L}_{mention} \tag{3.17}$$

where λ is the weight of mention detection loss.

[2]We also tried directly using the augmented information encoded by the graph attention network, but the result is even worse than the baseline model. We believe some critical information such as position embeddings may be lost after graph layers. Another explanation is that the gate mechanism helps preserve gradients from the document encoder after stacks of graph layers, playing a role similar to Highway Networks (Srivastava et al., 2015).

	OntoNotes 5.0			ACE 2005		
	Train	Dev	Test	Train	Dev	Test
#docs	2802	343	348	365	117	117
#mentions	155558	19155	19764	34340	11074	9167
#clusters	35142	4545	4532	11846	3760	3013

Table 3.1: The statistics of OntoNotes 5.0 and ACE 2005 datasets, including the number of documents, mentions and entity clusters.

Mention Detection Loss For mention detection loss, we use binary cross-entropy to optimize this objective:

$$\mathcal{L}_{mention} = \sum_{i=1}^{n} y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \tag{3.18}$$

where $\hat{y}_i = \text{sigmoid}(s_m(i))$, $y_i = 1$ if span i is a gold mention, otherwise $y_i = 0$.

Cluster Loss Follow Lee et al. (2017), we optimize the marginal log-likelihood of the intersection of candidate antecedent sets and gold coreference clusters:

$$\log \prod_{i=1}^{N} \sum_{\hat{y} \in \mathcal{Y}(i) \cap \text{GOLD}(i)} P(\hat{y}) \tag{3.19}$$

where $\text{GOLD}(i)$ is the set of correct antecedent spans in the gold cluster containing span i and $\mathcal{Y}(i)$ is the set of candidate antecedent spans of span i. $P(\hat{y})$ is the predicted antecedent distribution described in §3.3. For example, for a given mention C, we assume its candidate antecedent set is $\{A, B, D\}$ and its gold cluster is $\{A, B, C\}$. Then the intersection of candidate antecedent set and gold cluster is $\{A, B\}$, which means both A and B are correct antecedents of C. The cluster loss objective will thus maximize the coreference scores of $A - C$ and $B - C$.

3.6 Experiments

3.6.1 Experiment Setup

Dataset: To verify the effectiveness of our model, we conduct experiments on two benchmark datasets. The ACE 2005 dataset (Walker and Consortium, 2005) is the last version of the ACE dataset series, which contains annotations for Name Entity Recognition (NER), Relation Extraction(RE) and both Entity and Event Coreference Resolution. It has been extensively evaluated by various neural NER and RE models but forgotten by many modern neural coreference resolution models after the release of the OntoNotes 5.0 benchmark. Here we only evaluate our model on entity coreference resolution task. The ACE 2005 organizer has not officially released the test set (only the training set is available). Previous work used different train/dev/test splits over the training set, making the comparison among different systems inconsistent. We follow the same train/dev/test split after Lu and Ng (2020), resulting in 365, 117 and 117 documents in the training, development and test datasets, respectively.

The English OnotoNotes 5.0 benchmark (Pradhan et al., 2012) has a standard split, which consists of 2802, 343 and 348 documents in the training, development and test data sets, and it is also the most widely-used dataset for entity coreference resolution task.

Unlike the ACE 2005 dataset, singleton mentions are not annotated in OnoteNotes 5.0. Another key difference is that coreference links are only annotated between mentions that belong to the same entity type in ACE, while OntoNotes does not have such restrictions, allowing a mention to refer to other mentions that do not share the same entity type. The statistics of these two datasets are shown in Table 3.1.[3]

Implementation Details: We reimplement the C2F-COREF+*SpanBERT*[4] baseline using PyTorch and use the Independent setup for long documents. As discussed in §3.3, we

[3]The statistics in terms of the number of mentions and clusters in the ACE dataset is slightly inconsistent with what has been reported in Lu and Ng (2020). The reason is that after preprocessing the ACE dataset, we could get repeated mentions and coreference clusters that share the same mention. Such repeated mentions are removed and clusters with the same mention are merged in our experiment. We confirmed with the author of Lu and Ng (2020) that they have not done so.

[4]https://github.com/mandarjoshi90/coref

Hyparameters	OntoNotes 5.0		ACE 2005	
	SpanBert-B	SpanBert-L	SpanBert-B	SpanBert-L
Max span width	30	30	30	30
Max top antecedents	50	50	50	50
Max training segments	3	3	3	3
Top span ratio	0.4	0.4	0.35	0.4
Max segment length	384	512	384	512
SpanBERT learning rate	2×10^{-5}	1×10^{-5}	2×10^{-5}	1×10^{-5}
Task learning rate	3×10^{-4}	5×10^{-4}	3×10^{-4}	3×10^{-4}
Epochs	40	40	60	60
#GAT layers	2	2	2	2
#GAT heads	4	8	4	4
Edge label embedding size	300	300	300	300
Mention loss weight	0.5	0.5	100	100

Table 3.2: The best hyperparamters used in this experiment.

removed the high-order span representation refinement mechanism. The GAT implementation is based on Deep Graph Library (Wang et al., 2019a). Besides, since singleton mentions are annotated in ACE, we keep those mentions predicted to have no antecedents but have positive mention scores and construct singleton clusters for them during the postprocessing phase.

Training Details and Hyperparameter Setting: Most hyperparameters are adopted from previous work (Joshi et al., 2019; Lu and Ng, 2020) and newly introduced hyperparameters are determined through grid search. The ratio of proposing top mentions with high recall is set to 0.4 and 0.35 for OntoNotes and ACE datasets, respectively. We enumerate spans with a maximum length of up to 30 tokens. Documents are split into independent segments with a length of at most 384 for SpanBERT-Base and 512 for SpanBERT-Large. The maximum number of segments is set to 3 for both SpanBERT-Base and SpanBERT-Large models during training. The model is finetuned for 40 epochs on OntoNotes and 60 epochs on ACE datasets with a batch size of 1 (single document). The learning rates of finetuning SpanBERT-base and large model are 2×10^{-5} and 1×10^{-5}. The learning rates of task-specific parameters are 3×10^{-4} and 5×10^{-4} for OntoNotes, and 3×10^{-4} for ACE when using Base and Large model, respectively. The weight of mention loss λ is

empirically set to 0.5 and 100 for the OntoNotes and ACE05 dataset, respectively.[5] Both SpanBERT parameters and task parameters are trained using Adam optimizer (Kingma and Ba, 2015), with a warmup learning scheduler for the first 10% of training steps and linear decay scheduler decreasing to 0, respectively. The number of heads for syntactic and semantic sub-graphs is set to 4 for the ACE dataset, while 4 and 8 on OntoNotes for base and large models. The size of dependency and SRL tag embeddings is 300. The number of layers is set to 2 for both sub-graphs. The Base model is trained on a single Nvidia Tesla V100 GPU with 16G memory, while training of Large model requires 32G memory. A summary of training details and hyperparameters is shown in Table 3.2.

Feature Extraction: Gold features annotated on the OntoNotes 5.0 dataset are used in this experiment. We use Stanford CoreNLP toolkit (Manning et al., 2014) to convert the annotated constituent trees into Stanford dependency trees (de Marneffe and Manning, 2008). SRL labels are presented in the form of triples: (p, a, l), which refers to predicate, argument and label, respectively. For the ACE 2005 dataset, we use the off-the-shelf parsers from AllenNLP toolkit (Gardner et al., 2018) to obtain predicted dependency and SRL features.

3.6.2 Baselines and State-of-the-Art

We compare our proposed model with a range of end-to-end neural coreference resolution models:

- E2E-COREF (Lee et al., 2017) (§2.6.1) is the first end-to-end neural model for coreference resolution which jointly detects and groups entity mention spans.
- C2F-COREF (Lee et al., 2018) (§2.6.2) improves the E2E-COREF model (Lee et al., 2017) by introducing the *coarse-to-fine* candidate antecedent pruning strategy and the high-order span refinement mechanism. The ELMo embeddings (Peters et al., 2018) are also leveraged to boost model performance.
- EE (Kantor and Globerson, 2019) uses cluster-level information to improve coreference resolution by summing all mentions in the cluster as the approximation of cluster representations.

[5] We empirically find that for datasets with singleton mentions annotated, our model benefits from larger mention loss weight.

	MUC			B^3			$CEAF_{\phi_4}$			
	P	R	F1	P	R	F1	P	R	F1	Avg. F1
OntoNotes English Test Data										
E2E-COREF (Lee et al., 2017)	78.4	73.4	75.8	68.6	61.8	65.0	62.7	59.0	60.8	67.2
C2F-COREF (Lee et al., 2018)	81.4	79.5	80.4	72.2	69.5	70.8	68.2	67.1	67.6	73.0
EE (Kantor and Globerson, 2019)	82.6	84.1	83.4	73.3	76.2	74.7	72.4	71.1	71.8	76.6
SpanBERT-base (Joshi et al., 2020)	84.3	83.1	83.7	76.2	75.3	75.8	74.6	71.2	72.9	77.4
Our baseline + SpanBERT-base*†	83.6	83.9	83.7	75.1	76.5	75.8	74.2	71.6	72.9	77.5 ($\pm$0.1)
coref-HGAT + SpanBERT-base†	**85.1**	**84.5**	**84.8**	**77.4**	**77.2**	**77.3**	**75.5**	**73.3**	**74.4**	**78.8** ($\pm$0.1)
SpanBERT-large (Joshi et al., 2020)	85.8	84.8	85.3	78.3	77.9	78.1	76.4	74.2	75.3	79.6
Our baseline + SpanBERT-large*†	85.7	85.6	85.6	78.5	78.7	78.6	76.5	75.0	75.7	80.0 ($\pm$0.1)
coref-HGAT + SpanBERT-large†	**86.8**	**86.3**	**86.5**	**80.0**	**79.7**	**79.8**	**78.0**	**75.9**	**76.9**	**81.1** ($\pm$0.2)
CorefQA (Wu et al., 2020)	88.6	87.4	88.0	82.4	82.0	82.2	79.9	78.3	79.1	83.1
ACE 2005 English Test Data										
SpanBERT-base (Lu and Ng, 2020)	84.2	81.9	83.1	74.7	74.1	74.4	66.3	73.5	69.7	75.7
Our baseline + SpanBERT-base*†	84.8	83.1	83.9	75.4	75.9	75.6	65.4	73.9	69.4	76.3 ($\pm$0.2)
coref-HGAT + SpanBERT-base†	**86.1**	**81.9**	**84.0**	**77.9**	**74.9**	**76.4**	**66.0**	**75.7**	**70.5**	**77.0** ($\pm$0.3)
SpanBERT-large (Lu and Ng, 2020)	86.9	84.0	85.4	77.5	79.6	78.5	69.2	76.6	72.7	78.9
Our baseline + SpanBERT-large*†	87.7	83.9	85.8	80.0	77.2	78.6	66.9	77.7	71.9	78.7 ($\pm$0.3)
coref-HGAT + SpanBERT-large†	**87.5**	**84.3**	**85.9**	**80.0**	**77.8**	**78.9**	**67.8**	**77.6**	**72.4**	**79.1** ($\pm$0.2)

Table 3.3: The results on the test set of the OntoNotes English dataset and ACE 2005 dataset compared with previous systems. The main evaluation metric is the averaged F1 of MUC, B^3 and CEAFϕ_4. $*$ indicates our reimplemented baseline. † indicates average performance over 5 runs using different random seeds.

- SpanBERT (Joshi et al., 2020) is based on the C2F-COREF model by replacing the ELMo embeddings with pretrained SpanBERT embeddings to obtain better span-level representation.

- CorefQA (Wu et al., 2020) is the state-of-the-art model that deals with the coreference resolution problem using machine reading comprehension framework. The MRC framework can benefit the CorefQA model from being pretrained on existing large MRC datasets.

3.6.3 Results

The average F1 score of three metrics – MUC, B^3 and CEAFϕ_4 (§2.3) on the test set is reported by using the official CoNLL-2012 evaluation scripts.[6] Table 3.3 shows the results of coref-HGAT +SpanBERT-base and large model compared with previous work on both OntoNotes and ACE datasets. For the OntoNotes dataset, our model consistently outperforms the SpanBERT baseline (Joshi et al., 2020) on all three metrics with an improvement of 1.4% and 1.5% on Avg. F1 score respectively, as well as our reimplemented baseline (+1.3% and +1.1%), which is a substantial improvement by considering the difficulty of this task. Similarly, we can also observe that our model improves over our replicated baseline by a large margin, with an increase of 0.7% and 0.4%, respectively. Please note that, since the ACE dataset does not contain gold dependency syntax and SRL semantics, we use predicted features generated from third-party parsers. This demonstrates the effectiveness of our heterogeneous graph-based method in leveraging syntactic and semantic features, and such features are indeed useful in neural methods, even under the case when gold features are not available. Note that we also show the current state-of-the-art CorefQA model (Wu et al., 2020), which uses the span-prediction paradigm to compute pairwise coreference scores. The model is compatible with our method, i.e., adding our proposed graph attention and attentive integration layer on top of their document encoder with minor modification. The reason why we did not use it as a start baseline is due to hardware limitations since it requires 128G GPU memory for training. Moreover, we suspect that our model could gain further improvements when integrated with the CorefQA model, as recent work (Zhang et al., 2020b,c) has shown that incorporating syntax and semantics into BERT like model could achieve promising results on Machine Reading Comprehension task.

3.6.4 Analysis

Ablation Study We perform ablation study on the test set to investigate the contribution of different features in our model, with results shown in Table 3.4. We can see that both dependency features and SRL labels individually contribute to the success of our final model with a minor difference (+1.0% and 0.9%), and the gains are complementary to each other.

[6] http://conll.cemantix.org/2012/software.html

	Avg. F1	$\Delta F1$
Baseline	77.5	-
+ Dep	78.5	+1.0
+ SRL	78.4	+0.9
+ Dep & SRL	78.8	+1.3
GAT Layer = 1	78.5	-0.3
GAT Layer = 2	78.8	-
GAT Layer = 3	78.6	-0.2

Table 3.4: The Avg. F1 of coref-HGAT Base model by adding different features and stacking different number of GAT layers on the test set of OntoNotes 5.0.

Combination Strategy	Avg. F1	$\Delta F1$
Parallel	78.8	-
Sequential (Dep, SRL)	78.6	-0.2
Sequential (SRL, Dep)	78.2	-0.6

Table 3.5: The Avg. F1 of coref-HGAT Base model using different ways to integrate information from syntactic and semantic sub-graphs.

Effect of #Graph Layers From Table 3.4, we can see that both using one layer and three layers hurt model performance. This indicates that first-order information is not effective in capturing long-range dependencies while third-order information may cause overfitting due to too much model capacity.

Effect of Combination Strategy We also experiment with different strategies to combine information flow from two sub-graphs and present results on Table 3.5. The *Parallel* strategy means that we apply the information propagation mechanism on different graphs separately and combine the information using the attentive integration layer in Section 3.4.5. *Sequential* means that we sequentially perform information propagation with a specific order (from syntactic graph to semantic graph, or vice versa). We observe that the parallel strategy gives us the best result, and the sequential one does not introduce extra gains compared to models with a single graph, especially for the SRL-Dep setting, which hurts the semantic graph. This demonstrates that the parallel combination strategy could preserve information from both sub-graphs, but the sequential way can make information

	Dep	SRL	F1	$+\Delta F1$
Baseline	-	-	77.5	-
	Stanford	CoNLL05-SRL	78.1	+0.6
	Stanford	CoNLL12-SRL	78.2	+0.7
	Stanford	Gold	78.4	+0.9
	Biaffine	CoNLL05-SRL	78.2	+0.7
	Biaffine	CoNLL12-SRL	78.4	+0.9
	Biaffine	Gold	78.6	+1.1

Table 3.6: Averaged F1 score of coref-HGAT+Base model with predicted features against the baseline on the test set of OntoNotes 5.0.

from one graph overwhelmed by the other one.

Effect of Feature Quality To evaluate how the quality of features will affect the performance, we use the biaffine dependency parser (Dozat and Manning, 2017) and SRL parser (Shi and Lin, 2019) (denoted as *CoNLL12-SRL*) implemented by AllenNLP (Gardner et al., 2018) as well as the Stanford Parser (Chen and Manning, 2014) to extract features. The biaffine parser has roughly 3% LAS improvements compared to the Stanford CoreNLP parser on Penn Treebank. In addition to this, in order to evaluate the impact of different semantic role labelling parsers, we also implemented the same parser from Shi and Lin (2019) but trained on the CoNLL 2005 dataset (Carreras and Màrquez, 2004), which achieves an F1 of 81.9% on the out-of-domain setting. This parser is denoted as *CoNLL05-SRL* in our experiment. Table 3.6 shows the performance of our model when using different dependency parsers as well as predicted and gold SRL features. From the table, we can observe that better parsers and parsers trained in closer domains result in higher Avg. F1 score, with improvements of at most 0.9% when using both predicted features. Meanwhile, although our model suffers a performance drop from imperfect features, it can still achieve robust performance, outperforming the baseline with at least 0.6% improvement. Overall, high-quality features are important to a good performance of the proposed model, and further improvements are expected to see with the advances of both dependency and SRL parsers.

Doc length	#Docs	Baseline	Ours	$+\Delta F1$
0 – 128	57	82.9	85.4	+2.5
129 – 256	73	81.8	83.1	+1.3
257 – 512	78	82.2	83.2	+1.0
512 – 768	71	77.7	78.2	+0.5
769 – 1152	52	76.8	78.6	+1.8
1153+	12	67.5	70.3	+2.8
All	343	77.8	79.2	+1.4

Table 3.7: The Avg. F1 on the development set of the SpanBERT-base model and our core-HGAT+Base model, broken down by document length following Xia et al. (2020).

Document Length In Table 3.7, we show the performance of our model against the baseline on the development set as a function of document lengths. As expected, our model consistently outperforms the baseline model on all document sizes, especially for documents with lengths larger than 765 tokens. This demonstrates that the incorporated external syntax and semantics are beneficial for modelling longer dependencies. However, our model has a similar pattern as the baseline model, performing distinctly worse as document length increases. This shows that the sentence-level syntax and semantics used in this work are not sufficient to tackle the deficiency of modelling long-range dependency. One possible solution is to leverage document-level features such as hierarchical discourse structures.

Mention Detection As a subtask for coreference resolution, the performance on detecting mentions has a direct impact on the following coreference linking task, and recent progress made on coreference resolution task has largely benefited from better mention detectors (Lu and Ng, 2020). Therefore, to further understand our model, we analyse its performance on the mention detection subtask. Figure 3.3 shows the mention detection accuracy according to different mention span width ranges. We can see that our model consistently performs better than the baseline model. The advantage becomes clearer and clearer with the increase in span lengths, especially for longer spans which consist of more than five tokens. This is another strong evidence showing that external syntax and semantics make a difference in capturing long-range dependency and the resulted better mention detection performance contributes to significant improvement on the final coreference resolution task.

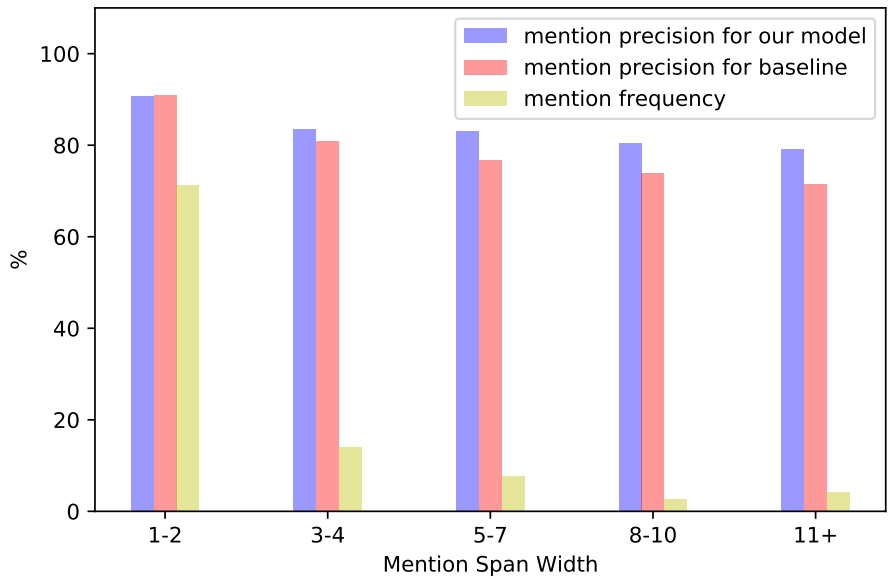

Figure 3.3: The performance of mention detection subtask on the development set of OntoNotes 5.0, broken down by mention span width.

3.7 Error Analysis

In this section, we classify errors made by our proposed model into different types by following previous work (Kummerfeld and Klein, 2013; Liu et al., 2020). Specifically, we compare our model with the baseline on the following seven types of errors: (1) Span Error: the predicted mention shares some content words with the gold one; (2) Conflated Entities: two clusters belong to different entities are merged into the same cluster; (3) Extra Mention: an extra mention is included in an entity cluster; (4) Extra Entity: an entity should not exist; (5) Divided Entity: two clusters should have been merged into a larger cluster; (6) Missing Mention: a mention should be introduced and included in a cluster; (7) Missing Entity: an entire entity is missing.

With the compared results shown in Table 3.8, we can see that our proposed model consistently makes fewer mistakes compared to the baseline on most error types, indicating the effectiveness of leveraging external syntax and semantics. Specifically, our model performs much better in capturing global entity-level information (114 fewer conflated entity errors) and introducing fewer extra mentions and entities. Besides, it can also better identify valid

Error	Baseline	Ours
Span Error	269	**195** (-74)
Conflated Entities	912	**798** (-114)
Extra Mention	493	**445** (-48)
Extra Entity	566	**522** (-44)
Divided Entity	884	**852** (-32)
Missing Mention	**583**	600 (+17)
Missing Entity	**565**	590 (+25)

Table 3.8: The number of each type of error made by our proposed model compared to the baseline in all documents from the development set of OntoNotes 5.0.

mentions with the help of syntax and semantics. By contrast, our model achieves inferior performance in terms of missing mentions and entities. This shows that our model is more rigorous and careful in selecting entity mentions and creating clusters. But overall, by making much fewer errors on most error types, our model achieves significant improvements over the baseline on both precision and recall scores.

To further understand the errors made by our proposed model, we choose conflated entity as the only error source and classify it into different sub-categories. Following previous work (Joshi et al., 2019; Wu and Gardner, 2020), we randomly choose 100 conflated entity errors made by our model from the development set of OntoNote 5.0 and show associated examples and the number of errors in each sub-type in Table 3.9. From the table, we can see that our model makes most mistakes on pronouns and tends to be confused by various kinds of text string matches, which are consistent with our intuition.

For the pronouns, our model is most likely to link to wrong common noun phrases or non-entity mentions. Sometimes when clusters only include pronoun mentions, our model also tends to include irrelevant pronouns, which shows the necessity of introducing high-order features to enrich pronoun representation. Besides, a similar bad performance is also witnessed in first and second pronouns when pronouns contain the speaker information and the entity representing the speakers does not explicitly appear in the context, thus misleading the model. This phenomenon becomes much worse when speaker switching happens frequently and many speakers participate in the conversation, which indicates the necessity of improvements in dialogue modelling. Other errors such as exact match also happen often. An example is shown in the third row of Table 3.9. The first *Disney* refers

Category	Example	#
Pronoun	... a cross-sea bridge connecting **Hong Kong, Zhuhai, and Macao** ... after **their** return, Macao, and Hong Kong, the two special ... regions ...	70
Exact Match	... heads toward **Disney**, trying to experience this mysterious park from close by. The most important thing about **Disney** is that it is a global brand.	12
Head Match	Off the coast there was a dramatic rescue by a **cruise ship** ship. **The ship** was sinking by the second, and as the waves pounded against them ...	7
Otherwise Match	Nixon concluded five days of private talks with Chinese leaders in **Beijing** ... **Beijing's** rulers complained ... U.S. interference in China's domestic affairs.	2
Semantic Proximity	... **a picture that people have long been looking forward to** started ... ... **these well-known cartoon images** once again caused Hong Kong ...	3
Others	... crossing from bases in **neighboring Angola**, violating U.N. ... ... Pretoria was attempting to sabotage next week's elections in **Namibia**.	6

Table 3.9: Qualitative Analysis: examples of classifying the conflated entity error type into different categories. We present two snippets for each category with bold mentions referring to two incorrectly linked entities. # indicates the number of mistakes made in each sub-class of 100 conflated entities randomly chosen from the development set of OntoNotes 5.0.

to the *Hong Kong Disneyland*, which is associated with the *Facility* named entity type; while the second *Disney* means *Disney the corporation*. This information can actually be inferred from its surrounding contexts *park* and *brand*. Therefore, more semantic and better contextual information which helps distinguish mentions with exact text strings and similar semantics should be leveraged for further improvement.

3.8 Resolution Classes

To further understand the behaviour of our proposed models, we compare its performance with the baseline model on different types of entity mentions. Following Stoyanov et al. (2009) and Lu and Ng (2020), we classify gold mentions into different resolution classes as discussed below.

Proper Names Gold mentions associated with named entity types are assigned to *proper names* on the OntoNotes dataset, while for ACE05 dataset, mentions whose head words

are annotated with *NAM* are classified into this class. Moreover, four sub-classes are defined. (1) *e*: a proper name belongs to the *exact string match* class if there exists at least one preceding mention in its gold coreference cluster that exactly has the same string; (2) *p*: a proper name belongs to the *partial string match* class if there exists at least one preceding mention in its gold coreference cluster that shares some words; (3) *n*: a proper name belongs to the *no string match* class if there exists no preceding mention in its gold coreference cluster that shares some words; and (4) *na*: a proper name belongs to the *non-anaphoric* class if it does not refer to any preceding mention. For the OntoNotes dataset, the first mention in each gold coreference cluster is regarded as non-anaphoric, while for the ACE dataset, the *non-anaphoric* class also includes singleton mentions.

Common NPs Gold mentions without named entity types belong to *Common NPs* on the OntoNotes dataset, and mentions with nominal head words are assigned to this class on the ACE05 dataset. Similarly, we define four sub-classes: (5) *e*; (6) *p*; (7) *n*; and (8) *na*.

Pronouns Five pronoun sub-classes are defined. (9) *1/2*: 1st and 2nd person pronouns (e.g., you); (10) *G3*: gendered 3rd person pronouns (e.g., she); (11) *U3*: ungendered 3rd person pronouns (e.g., they); (12) *oa*: any anaphoric pronouns that do not belong to (9), (10), and (11) (e.g., demonstrative pronouns); and (13) *na*: non-anaphoric pronouns (e.g., pleonastic pronouns).

Results For performance measurements, we follow Lu and Ng (2020) to use mention detection recall (MD) and resolution accuracy (RA). For MD, we count the percentage of gold mentions that are correctly detected in each resolution class; while for RA, we compute the percentage of correctly detected mentions that are correctly resolved.[7]

Table 3.10 shows the performance of the baseline and our proposed model on each resolution class. Firstly, we can see that both models perform the best on proper names, followed by common nouns and pronouns. Secondly, by analysing the fine-grained classes, the *exact match* class in proper names and common nouns are easier than the *partial match* one, which in turn is easier than the *no string match* class. For pronouns, the 3rd person

[7] A gold anaphoric mention is regarded as correctly resolved if its predicted antecedent is in the corresponding gold coreference cluster. Moreover, it will be considered correctly resolved for a gold non-anaphoric mention if it is predicted to have a dummy antecedent.

	OntoNotes 5.0					ACE 2005				
		Baseline		Ours			Baseline		Ours	
Class	Size %	RA	MD	RA	MD	Size %	RA	MD	RA	MD
PN-e	15.52	**96.5**	93.9	95.4	**94.1**	15.61	95.8	**97.3**	**97.4**	**97.3**
PN-p	6.06	89.7	86.0	**91.1**	**87.7**	2.96	**81.1**	88.7	79.7	**91.5**
PN-n	6.63	86.2	**88.8**	**86.9**	88.0	2.36	**67.9**	83.5	65.0	**85.4**
PN-na	6.74	94.2	82.0	**95.7**	**83.5**	11.90	84.4	88.2	**85.4**	**89.3**
CN-e	6.17	96.3	**91.5**	**96.7**	91.1	3.86	**88.4**	86.7	88.2	**86.9**
CN-p	8.60	82.8	**81.0**	**84.8**	80.8	5.31	**64.2**	88.8	60.5	**89.1**
CN-n	3.39	**74.1**	68.9	72.4	**69.4**	3.96	**60.6**	83.8	57.4	**86.6**
CN-na	15.47	91.6	69.5	**93.3**	**70.8**	18.24	89.4	84.4	**91.3**	**87.3**
PR-1/2	11.64	93.7	**95.8**	**94.3**	95.3	13.28	87.3	**99.9**	**88.2**	99.7
PR-G3	5.99	95.8	**99.6**	**95.9**	99.3	6.84	93.4	99.7	**93.5**	**99.9**
PR-UG3	10.14	87.0	**95.1**	**88.6**	94.4	5.71	81.5	97.6	**82.5**	**98.3**
PR-oa	1.45	65.9	63.5	**66.7**	**63.9**	1.22	**69.2**	**86.7**	66.7	**86.7**
PR-na	2.20	54.2	**88.4**	**56.4**	85.8	8.73	48.6	88.7	**50.5**	**89.1**

Table 3.10: The results of resolution classes in the development set of OntoNotes 5.0 and ACE 2005. Each row contains the performance on each fine-grained resolution class. **Size** represents the percentage of mentions in a specific resolution class over all mentions. **RA** and **MD** means resolution accuracy and mention detection recall, respectively.

gendered pronoun is the easiest one, followed by the 1st/2nd person noun, while both models find it difficult to resolve other pronouns such as reflective pronouns.

Thirdly, we find that our model gains most of its improvements on non-anaphoric mentions, showing its superiority in dealing with the difficulty of anaphoricity determination, with improvements up to 2.2% RA. Moreover, the improvements on 3rd ungendered pronouns are also significant (1.6% and 1.0%). This demonstrates that the harder a resolution class is, the more significant our model's improvement is. Besides, this also shows that the leveraged syntax and semantics help resolve traditionally difficult anaphors. Overall, by maintaining comparable performance in other easier classes simultaneously, our model has achieved significantly better final results on these two datasets compared with the baseline.

3.9 Summary

In this chapter, we propose a heterogeneous-graph based model to enhance coreference resolution by effectively leveraging dependency tree structures and SRL semantic features.

Particularly, nodes of different granularity in the graph propagate and aggregate information to and from neighbour nodes to obtain both syntactic and semantic augmented representation. Moreover, an attention-based mechanism is used to dynamically aggregate such augmented information. Experiments on the OntoNotes 5.0 benchmark and ACE 2005 dataset confirm the effectiveness of our proposed model with significant improvement achieved against the strong baseline. Future work will focus on applying other features, such as constituent parsing trees and WordNet.

Chapter 4

Evaluating the Utility of Constituent Syntax

4.1 Introduction

Recent attempts show that graph-based methods for leveraging dependency syntax have benefited various neural models (Marcheggiani and Titov, 2017; Bastings et al., 2017; Wang et al., 2020). By contrast, encoding constituent syntax trees using graph-related techniques is less explored. Previous work either employed binarized trees (Wang et al., 2007) or designed TreeLSTM to encode tree structures (Tai et al., 2015). But they do not take full advantage of rich syntactic information and relationships between nodes.

In this chapter, we argue that incorporating constituent syntax is natural for coreference resolution. In constituent trees, the information encoding boundaries of non-terminal phrases is explicitly presented. Extra linguistic labels also reveal linguistic constraints for coreference resolution (Ng, 2010)(e.g., are both noun phrases definite?). By contrast, such information is either implicitly embedded or not revealed in the dependency tree (Chapter 3). Moreover, constituent syntax has long been employed in coreference resolution task. Hobbs (1978) uses a rule-based and breadth-first traversal of parse trees to resolve referents of given mentions. Luo and Zitouni (2005) design various features from constituent trees guided by the Binding theory (Chomsky, 1988), which describes the constraints on finding antecedents of English pronouns. Non-anaphoric information encoded in constituent trees is also employed to benefit the anaphoricity determination task using tree kernel-based

methods (Zhou and Kong, 2009).

In this chapter, we propose a neural coreference resolution model using constituent syntax, as an extension of Joshi et al. (2019). Trieu et al. (2019) and Kong and Jian (2019) apply constituent trees as hard constraints to filter invalid mentions. However, they fail to leverage the hierarchical constituent structures or encode such information using complex hand-engineered path features. In contrast, our method builds a graph consisting of terminal and non-terminal nodes and applies graph neural networks to encode such structures more flexibly.

The most similar work compared to ours is that of Marcheggiani and Titov (2020), which applies Graph Convolutional Networks (GCNs) (Kipf and Welling, 2017) over constituent trees for Semantic Role Labelling. This work differs from Marcheggiani and Titov (2020) both in terms of application (SRL vs Coreference), and important aspects of the algorithm: 1) we extend the plain constituent trees by adding edges with higher orders and interpret parsing trees with dual graphs to capture forward and backward views (Zhao et al., 2020; Ribeiro et al., 2019); 2) we use a novel span yield enhanced method to represent constituent nodes (§4.3.2.1) instead of initializing them with zero vectors, and we believe our method is more natural to conference resolution and consistent with the span representations. We also design a new information propagation mechanism over the underlying extended graph, where constituent node representations are updated iteratively using bidirectional graph neural networks, and explicit hierarchical syntax and span boundary information are propagated to enhance the contextualized token embeddings. We conduct experiments on the English and Chinese portions of OntoNotes 5.0 (Pradhan et al., 2012) benchmark, and show that our proposed model significantly outperforms a strong baseline and achieves new state-of-the-art performance on the Chinese dataset.

4.2 Related Work

External syntax has long been used for enhancing neural models to benefit a variety of NLP tasks. Socher et al. (2013) and Tai et al. (2015) utilise recursive neural networks for encoding constituent trees through creating the representation of constituent terminal nodes recursively. Nevertheless, such a method is less efficient than applying graph neural networks since the recursive way of encoding trees means that later steps should depend on earlier

ones. Moreover, nodes on top layers, especially the root node, encode the information of the entire tree, which will make it prone to noisy and imperfect features. By contrast, our method, which uses graph attention networks, only encodes a small part of the constituent tree around the central updated node, making it more robust when only predicted syntax is available. The utilisation of syntax for NLP tasks in later time mainly relies on graph neural networks to capture the structural information. Marcheggiani and Titov (2017); Bastings et al. (2017) use graph convolutional networks to leverage dependency syntax to improve semantic role labelling (SRL) and machine translation tasks. Wang et al. (2020) proposes to employ reshaped dependency syntax trees to improve aspect-based sentiment analysis.

Compared to dependency syntax, the constituent syntax has less been explored. Wang et al. (2019b) uses the full representation of constituency parsing trees (Gómez-Rodríguez and Vilares, 2018) as word-level features to improve a SRL model. Trieu et al. (2019) and Kong and Jian (2019) treat constituent trees as signals to filter invalid candidate mention spans for coreference resolution task. However, their methods either ignore the hierarchical structures encoded in parsing trees or create complex hand-designed path-related features. In contrast, our method is more flexible, which only utilises graph attention networks and node representations are learned automatically. Compared to ours, the most similar work is the Marcheggiani and Titov (2020)'s SpanGCN model for the SRL task, which leverages GCNs to encode the structure of constituent parsing trees and uses information propagation mechanisms to enhance word-level features with learned constituent node representations. Nevertheless, our method differs in extending plain parsing trees with higher-order edges (grandparent-grandchild edges) and dual graphs capturing the forward and backward views. More importantly, we apply graph attention networks to encode constituent syntax trees for coreference resolution rather than the SRL task. We demonstrate that the introduced constituent parse trees and our encoding methods can improve a strong coreference resolution baseline by a large margin and achieve comparable performance with the state-of-the-art model in English and establish new state-of-the-art performance in Chinese.

4.3 Proposed Model

Our model is based on Joshi et al. (2019) (§2.6.1 and §2.6.2), which we extend by integrating constituent syntactic features. This work is also different from the proposed

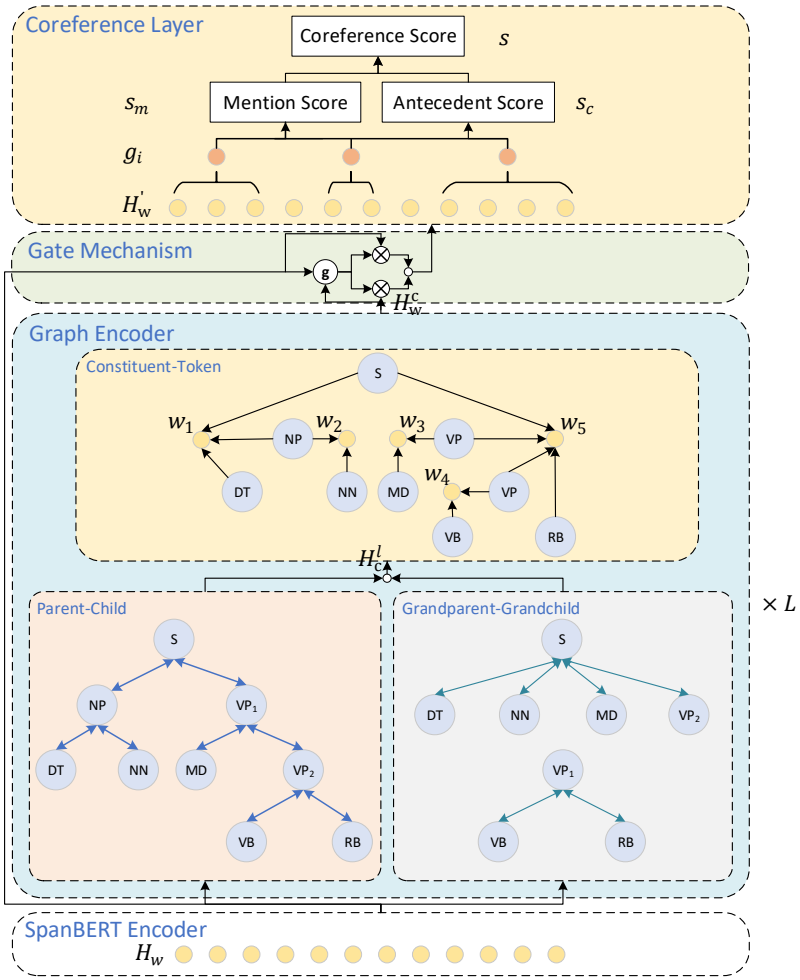

Figure 4.1: The overall architecture of our proposed model.

method in Chapter 3: we incorporate constituent parse trees rather than dependency trees and introduce extra features on top of vanilla parse trees to capture longer dependencies among nodes. It first encodes the whole document using the sliding-window approach (Section 4.3.1), then a constituent syntax based graph which consists of multiple types of edges is constructed (Section 4.3.2). An information propagation mechanism is applied to update the representation of constituent nodes and propagate enriched constituent node embeddings to enhance the contextualized basic token representations (Section 4.3.3 and 4.3.4). Finally, enhanced token representations will be employed to form mention span

embeddings and compute pairwise coreference scores. Figure 4.1 shows the overall architecture of our proposed model.

4.3.1 Document Encoder

SpanBERT (Joshi et al., 2020) is a pretrained language model which focuses on pretraining span representations through span boundary detection task. Following previous work (Joshi et al., 2019), we use SpanBERT as the document encoder and finetune the SpanBERT model for coreference resolution.

BERT-based models are pretrained to encode single sentence or sentence pairs with at most 512 word pieces. However, a document normally contains more than 512 tokens. Previous work chooses to split documents into independent segments in order to fit long documents. But one drawback of this method is that it has limited modelling capacity as tokens can only attend to other tokens within the same segment, especially for tokens at the boundary of each segment (Joshi et al., 2019). Rather than using independent segments for encoding long documents, we follow previous work (Wu et al., 2020) to create overlapped segments. Specifically, we use a sliding-window approach to create T-sized segments with a stride of $\frac{T}{2}$ tokens. Our preliminary experimental results show that the sliding-window approach performs slightly better than the independent setup (Joshi et al., 2019). Besides, speaker identities are crucial for coreference resolution, especially for pronouns in dialogues and multi-party conversations. Unlike previous methods (Lee et al., 2017, 2018; Joshi et al., 2019) which convert the speaker information of two compared mentions into binary features (1 if a pair of mentions appear in the utterances of the same speaker and 0 otherwise), we instead follow Wu et al. (2020) to directly insert the speaker's name at the beginning of the corresponding utterance. Therefore, the overlapped segments with attached speaker information are then encoded by the SpanBERT encoder to obtain contextualized representation, which can be denoted as $\mathbf{H}_w = (\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n)$, where $\mathbf{h}_i \in \mathcal{R}^d$ and n is the document length.

4.3.2 Graph Construction

For each sentence in the document, we have an associated constituent tree which consists of words (terminals) and constituents (non-terminals). Therefore, we have two types

of nodes in our graph: token nodes ($T = \{t_1, t_2, \ldots, t_n\}$)[1] and constituent nodes ($C = \{c_1, c_2, \ldots, c_m\}$), where n and m are the number of token and constituent nodes, respectively. Moreover, for a given constituent node c_i, we use START(i) and END(i) to denotes its start and end token indices.

4.3.2.1 Node Initialization

We use $\mathbf{H}_w$ to initialize the features of token nodes, while given a constituent node $c_i \in C$, the node representation $\mathbf{h}(c_i)$ is defined as:

$$\mathbf{h}(c_i) = [\mathbf{h}_{\text{START}(i)}; \mathbf{h}_{\text{END}(i)}; \mathbf{e}_{\text{type}}(s_i)] \tag{4.1}$$

where $\mathbf{h}_{\text{START}(i)}$ and $\mathbf{h}_{\text{END}(i)}$ are the contextualized embeddings of start and end tokens of constituent c_i and $\mathbf{e}_{\text{type}}(s_i)$ is the constituent type embeddings. Therefore, we could obtain a set of initialized constituent node representations: $\mathbf{H}_c = \{\mathbf{h}(c_1), \mathbf{h}(c_2), \ldots, \mathbf{h}(c_m)\}$.

4.3.2.2 Edge Construction

An example of graph structures is shown in Figure 4.2.

Constituent-Constituent We design two categories of edges in our graph, namely *parent-child* and *grandparent-grandchild*, to capture longer-range dependencies. For each edge category, we further add reciprocal edges for each edge in the graph and label them with *forward* and *backward* types, respectively. Additionally, self-loop edges are added to each node in the graph. Thus, the edges are constructed based on following rules:

- A pair of **parent-child** and **child-parent** edges between node c_i and c_j are constructed if and only if these two nodes are directly connected in the constituent tree.
- A pair of **grandparent-grandchild** and **grandchild-grandparent** edges between node c_i and c_j are constructed if and only if node c_i can reach node c_j using two hops, and vice versa.

[1] We use token and word interchangeably throughout this chapter.

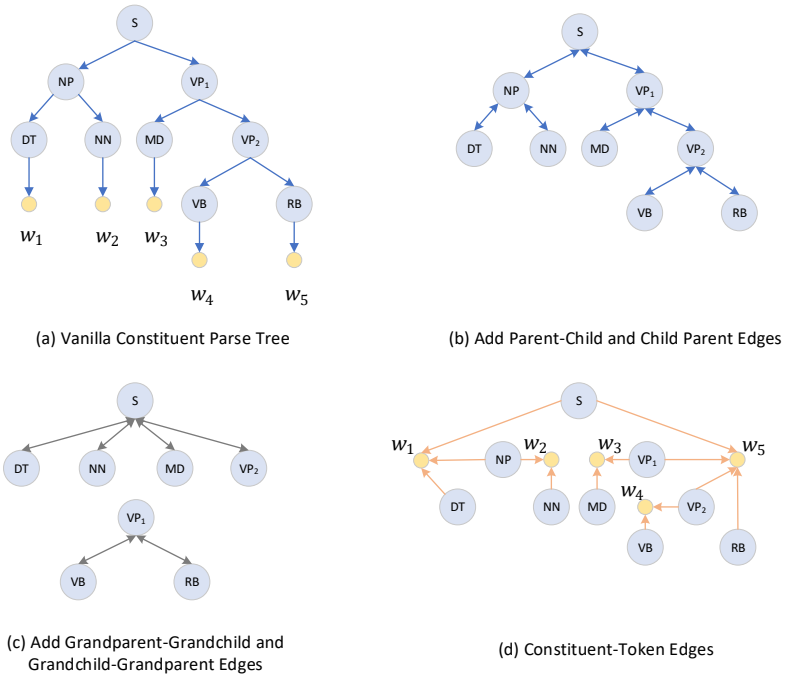

Figure 4.2: An example of our constructed graph based on constituent parse trees by adding forward and backward edges and higher-order edges.

Constituent-Token A token node w_i is linked to c_j if it is the left or rightmost token in the yield of c_j. Such edges are made unidirectional to make sure that information can only be propagated from constituent nodes to token nodes, which aims to enrich basic token representations with span boundary information and the hierarchical syntax structures.

4.3.3 Graph Encoder

We use a Graph Attention Network (GAT) (Veličković et al., 2018) to update the representation of constituent nodes and propagate syntactic information to basic token nodes. For a node i, the attention mechanism allows it to selectively incorporate information from its neighbour nodes:

$$\alpha_{ij} = \text{softmax}(\sigma(\mathbf{a}^T[\mathbf{W}\mathbf{h}_i; \mathbf{W}\mathbf{h}_j])) \tag{4.2}$$

$$\mathbf{h}'_i = \|_{k=1}^{K} \text{ReLU}(\sum_j \alpha_{ij}^k \mathbf{W}^k \mathbf{h}_j) \tag{4.3}$$

where $\mathbf{h}_i$ and $\mathbf{h}_j$ are embeddings of node i and j, $\mathbf{a}^T$, $\mathbf{W}$ and $\mathbf{W}^k$ are trainable parameters. σ is the LeakyReLU activation function (Xu et al., 2015). $\|$ and $[;]$ represent the concatenation operation. Eqs. 4.2 and 4.3 are designated as an operation:

$$\mathbf{h}'_i = \mathrm{GAT}(\mathbf{h}_i, \mathbf{h}_j | j \in \mathcal{N}_i) \tag{4.4}$$

where $\mathcal{N}_i$ is the set of neighbour nodes of the target node i, $\mathbf{h}_i$ and $\mathbf{h}_j$ are the embeddings of target and neighbour node and $\mathbf{h}'_i$ is the updated embedding of target node.

Bidirectional GAT Layer We design a bidirectional GAT layer to model the constituent-constituent edges with directions illustrated in §4.3.2.2. Specifically, for a given constituent node c_i, we could obtain its neighbour nodes with edge type k in *forward* (outgoing) and *backward* (incoming) directions: $\mathcal{N}^{kf}_{c_i}$ and $\mathcal{N}^{kb}_{c_i}$, respectively. Then we use two separate GAT encoders to derive the updated representation of node c_i in different directions:

$$\mathbf{h}^{kf}_{c_i} = \mathrm{GAT}(\mathbf{h}_{c_i}, \mathbf{h}_{c_j} | c_j \in \mathcal{N}^{kf}_{c_i}) \tag{4.5}$$

$$\mathbf{h}^{kb}_{c_i} = \mathrm{GAT}(\mathbf{h}_{c_i}, \mathbf{h}_{c_j} | c_j \in \mathcal{N}^{kb}_{c_i}) \tag{4.6}$$

Then the updated representation of constituent node c_i is obtained by the summation of the representations of two directions: $\mathbf{h}^{k}_{c_i} = \mathbf{h}^{kf}_{c_i} + \mathbf{h}^{kb}_{c_i}$.

Multi-type Integration Layer In order to aggregate updated node representations using different types of edges, we use the self-attentive mechanism (Lee et al., 2017):

$$\alpha_{c_i,k} = \mathrm{softmax}(\mathbf{FFNN}(\mathbf{h}^{k}_{c_i})) \tag{4.7}$$

$$\mathbf{h}_{c_i} = \sum_{k=1}^{K} \alpha_{c_i,k} \mathbf{h}^{k}_{ci} \tag{4.8}$$

where **FFNN** is a two-layers feedforward neural network with **ReLU** function. Furthermore, we designate an operation to summarise above procedures:

$$\mathbf{h}'_{c_i} = \text{Multi-BiGAT}(\mathbf{h}_{c_i}, \mathbf{h}_{c_j} | c_j \in \mathcal{N}_{c_i}) \tag{4.9}$$

4.3.4 Information Propagation

With the above-defined operations, we can design the information propagation mechanism to enable information flow from constituent nodes to basic token nodes through iterative updates. First, we update the constituent node representation using our defined bidirectional GAT layer with multi-type edges:

$$\mathbf{h}_{c_i}^{l} = \text{Multi-BiGAT}(\mathbf{h}_{c_i}^{l-1}, \mathbf{h}_{c_j}^{l-1} | c_j \in \mathcal{N}_{c_i}) \tag{4.10}$$

where $h_{c_i}^{l-1}$ is the constituent node representation from previous layer $l-1$ and $h_{c_i}^{0}$ is the initialized embedding.

Then the updated constituent node representation propagate information to update the token node representation through constituent-token edges:

$$\mathbf{h}_{i}^{l} = \text{GAT}(\mathbf{h}_{i}^{l-1}, \mathbf{h}_{c_j}^{l} | c_j \in \mathcal{N}_{i}) \tag{4.11}$$

where $\mathbf{h}_i^{l-1}$ is the token representation from previous layer $l-1$ and $\mathbf{h}_i^0$ is the SpanBERT encoding.

The updated token representation is utilised to reconstruct the updated constituent node embeddings using Eq. 4.1, which will be employed in the next graph encoder layer. After L iterations, we could obtain the final constituent syntax enhanced token representations, which can be denoted as $\mathbf{H}_w^c$. We then use a gate mechanism to infuse the syntax-enhanced token representation dynamically:

$$\mathbf{f} = \sigma(\mathbf{W}_g \cdot [\mathbf{H}_w; \mathbf{H}_w^c] + \mathbf{b}_g) \tag{4.12}$$

$$\mathbf{H}'_w = \mathbf{f} \odot \mathbf{H}_w + (\mathbf{1} - \mathbf{f}) \odot \mathbf{H}_w^c \tag{4.13}$$

where $\mathbf{W}_g$ and $\mathbf{b}_g$ are trainable parameters, $\odot$ means element-wise multiplication and σ is the logistic sigmoid function.

Finally, the constituent syntax augmented token representation $\mathbf{H}'_w$ can be used to form span representation and compute pairwise coreference score. More details about how to construct span emebddings and compute coreference scores have been discussed thoroughly in Section 2.6.1 and 2.6.2.

Please note that the information propagation mechanism proposed here is different from

	English			Chinese		
	Train	Dev	Test	Train	Dev	Test
#docs	2802	343	348	1810	252	218
#mentions	155558	19155	19764	102853	14183	12801
#clusters	35142	4545	4532	28256	3875	3559

Table 4.1: The statistics of the English and Chinese portions of OntoNotes 5.0 dataset in terms of the number of documents, mentions and entity clusters.

that of Chapter 3 in terms of aspects of algorithms: 1) different information propagation path among different kinds of nodes; 2) we additionally introduce edges in *forward* and *backward* directions and higher-order edges to capture longer-range dependencies, and thus newly designed graph attention layers are employed to model additionally introduced features.

4.4 Experiments

4.4.1 Experiment Setup

Dataset Our model is evaluated on the English and Chinese portions of OntoNotes 5.0 dataset (Pradhan et al., 2012). The English corpus consists of 2802, 343 and 348 documents in the training, development and test splits, respectively, while the Chinese corpus contains 1810, 252 and 218 documents for train/dev/test splits. The model is evaluated using three coreference metrics: MUC, B^3 and CEAFϕ_4 and the average F1 score (Avg. F1) of the three are reported. We use the latest version of the official evaluation scripts (version 8.01),[2] which implements the original definitions of the metrics. The statistics of these two datasets are shown in Table 4.1.

Implementation Details We reimplement the C2F-COREF+*SpanBERT*[3] baseline using PyTorch. For English model, we use SpanBERT-base and large model to encode documents;[4] while for Chinese, we use BERT-wwm-base and RoBERTa-wwm-ext-large[5] as the

[2] http://conll.cemantix.org/2012/software.html
[3] https://github.com/mandarjoshi90/coref
[4] https://github.com/facebookresearch/SpanBERT
[5] https://github.com/ymcui/Chinese-BERT-wwm

Hyparameters	English		Chinese	
	SpanBERT-B	SpanBERT-L	BERT-B	RoBERTa-L
Max span width	30	30	30	30
Max top antecedents	50	50	50	50
Max training segments	3	3	3	3
Top span ratio	0.4	0.4	0.3	0.3
Max segment length	384	512	384	512
BERT learning rate	2×10^{-5}	1×10^{-5}	2×10^{-5}	1×10^{-5}
Task learning rate	3×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}
Epochs	40	40	40	40
Num of GAT layers	2	2	2	2
Num of GAT heads	4	8	4	8
Node type embedding size	300	256	300	256

Table 4.2: The best hyperparamters used in this experiment.

document encoders. As suggested by Xu and Choi (2020), we discard the high-order span representation refinement module. Graph attention networks and the information propagation module are implemented based on Deep Graph Library (Wang et al., 2019a).[6] We follow (Lee et al., 2017) to use the same marginal log-likelihood based optimization objective. Besides, mention detection loss is also added as described in Section 3.5. Gold constituent parsing trees annotated on the datasets are used in this experiment.

Hyperparameter Settings For the English model, we follow most of the parameters in Section 3.6.1, while for the Chinese model, we have several hyperparameters to tune using grid search. Specifically, we search for: 1) max span with (maximum number of words a candidate mention contains) out of {20, 25, 30}; 2) number of training epochs out of {40, 50, 60}; 3) top span ratio (the fraction of candidate mentions with top mention scores that are kept for mention linking) out of {0.3, 0.35, 0.4}; 4) task parameters learning rate out of {1e-4, 2e-4, 3e-4, 4e-4, 5e-4}; 5) BERT parameters learning rate out of {1e-5, 2e-5, 3e-5, 4e-5, 5e-5}; 6) mention loss ratio out of 1, 10, 100; 7) number of graph layers out of {2, 3, 4} and 8) number of graph attention heads out of {4, 8, 16}. The best hyperparameter setting is shown in Table 4.2.

[6] https://github.com/dmlc/dgl

Training Details Both BERT parameters and task parameters are trained using Adam optimizer (Kingma and Ba, 2015), with a warmup learning scheduler for the first 10% of training steps and linear decay scheduler decreasing to 0, respectively. Each model is run five times using different random seeds and the averaged performance is reported.

The training of base model is conducted on a single Nvidia Telsa V100 GPU with 16G memory, while training large model requires 32G memory. Training base model takes about 22 hours while training of large model can be finished within 32 hours.

Baselines and State-of-the-Art We compare our proposed model with a variety of previous competitive models: Clark and Manning (2016) is a neural network based model which incorporates entity-level information. E2E-COREF (Lee et al., 2017) is the first end-to-end neural model for coreference resolution which jointly detects and groups entity mention spans. Kong and Jian (2019) improves the Lee et al. (2017)'s model by treating constituent parsing trees as constraints to filter invalid candidate mentions and encoding the traversal node sequence of parsing trees to enhance contextualized document embeddings. C2F-COREF (Lee et al., 2018) extends the Lee et al. (2017)'s model by introducing a *coarse-to-fine* candidate mention pruning strategy and a higher-order span refinement mechanism. Joshi et al. (2020) improves over Lee et al. (2018) with the document encoder replaced by SpanBERT. CorefQA (Wu et al., 2020) employs the machine reading comprehension framework to recast the coreference resolution problem as a query-based span-prediction task, which achieves current state-of-the-art performance. Jiang and Cohn (2021) is the method we proposed in Chapter 3 which enhances neural coreference resolution by incorporating dependency syntax and semantic role labels using heterogeneous graph attention networks.

4.4.2 Main Results

Table 4.3 shows the results of our model compared with a range of high-performing neural coreference resolution models on English and Chinese. For English, we observe that our replicated baseline surpasses the SpanBERT baseline (Joshi et al., 2020) by 0.7% and 1.1%, demonstrating the effectiveness of the sliding-window based document encoding approach and modified representations of speaker identities (§4.3.1). Our model further improves the replicated baseline significantly with improvements of 1.9% and 1.4%, respectively, a

	MUC			B^3			$CEAF_{\phi_4}$			
	P	R	F1	P	R	F1	P	R	F1	Avg. F1
OntoNotes English Test Data										
E2E-COREF (Lee et al., 2017)	78.4	73.4	75.8	68.6	61.8	65.0	62.7	59.0	60.8	67.2
C2F-COREF (Lee et al., 2018)	81.4	79.5	80.4	72.2	69.5	70.8	68.2	67.1	67.6	73.0
SpanBERT-base (Joshi et al., 2020)	84.3	83.1	83.7	76.2	75.3	75.8	74.6	71.2	72.9	77.4
Our baseline + SpanBERT-base*†	83.9	84.2	84.0	76.2	76.9	76.6	74.3	73.1	73.7	78.1 (±0.1)
Jiang and Cohn (2021) + SpanBERT-base†	85.1	84.5	84.8	77.4	77.2	77.3	75.5	73.3	74.4	78.8 (±0.1)
Our model + SpanBERT-base†	**85.6**	**85.8**	**85.7**	**78.2**	**79.0**	**78.6**	**76.3**	**74.8**	**75.5**	**80.0** (±0.2)
CorefQA + base (Wu et al., 2020)	85.2	87.4	86.3	78.7	76.5	77.6	76.0	75.6	75.8	79.9
SpanBERT-large (Joshi et al., 2020)	85.8	84.8	85.3	78.3	77.9	78.1	76.4	74.2	75.3	79.6
Our baseline + SpanBERT-large*†	86.0	86.0	86.0	79.6	79.6	79.6	77.2	75.8	76.5	80.7 (±0.1)
Jiang and Cohn (2021) + SpanBERT-large†	86.8	86.3	86.5	80.0	79.7	79.8	78.0	75.9	76.9	81.1 (±0.2)
Our model + SpanBERT-large†	**87.3**	**87.1**	**87.2**	**81.1**	**80.9**	**81.0**	**78.8**	**77.2**	**78.0**	**82.1** (±0.2)
CorefQA + large (Wu et al., 2020)	88.6	87.4	88.0	82.4	82.0	82.2	79.9	78.3	79.1	83.1
OntoNotes Chinese Test Data										
Clark and Manning (2016)	73.9	65.4	69.4	67.5	56.4	61.5	62.8	57.6	60.1	63.7
Kong and Jian (2019)	77.0	64.6	70.2	70.6	54.7	61.6	64.9	55.4	59.8	63.9
Our baseline + BERT-wwm-base*†	76.7	70.9	73.7	68.3	62.4	65.2	67.4	60.8	63.9	67.6 (±0.3)
Our model + BERT-wwm-base†	**84.1**	**78.6**	**81.3**	**77.4**	**71.5**	**74.4**	**76.5**	**70.0**	**73.1**	**76.3** (±0.2)
Our baseline + RoBERTa-wwm-ext-large*†	79.9	72.2	75.8	71.6	64.3	67.7	70.8	62.8	66.5	70.0 (±0.3)
Our model + RoBERTa-wwm-ext-large†	**85.8**	**80.9**	**83.3**	**79.8**	**74.5**	**77.0**	**78.7**	**72.9**	**75.7**	**78.7** (±0.2)

Table 4.3: The results on the test set of the OntoNotes English and Chinese shared task compared with previous systems. The main evaluation metric is the averaged F1 of MUC, B^3 and CEAFϕ_4. $*$ indicates our reimplemented baseline. † indicates average performance over 5 runs using different random seeds.

result which is also comparable to the state-of-the-art performance of CorefQA (Wu et al., 2020).[7] Improvements can also be observed compared to our proposed method (Jiang and Cohn, 2021) in Chapter 3 (1.2% and 1.0%). For Chinese, our replicated baseline has already achieved state-of-the-art performance. With the help of constituent syntax, our model again beats the baseline model with substantial improvements of 8.7%. This indicates that constituent syntax is far more useful to Chinese than English, and we suspect that word-level segmentation encoded in constituent trees brings extra benefits in Chinese.

[7] We do not use their model as a baseline mainly due to hardware limitations.

Modified Module	Avg. F1	$\Delta F1$
-	80.0	-
Vanilla Tree	79.7	-0.3
No Gate	77.3	-2.7
Only Type Embedding	79.5	-0.5
Dependency Syntax	79.2	-0.8

Table 4.4: Results when modifying different modules compared to our base model on the English test set.

4.4.3 Analysis

Effects of Constituency Quality To evaluate how the quality of parsing trees will affect the performance, we test two off-the-shelf parsers (Zhang et al., 2020a) (achieving 95.26% and 91.40% F1 score on PTB and CTB7) to obtain predicted trees. When using predicted trees with our base model, we get Avg. F1 of 78.7% (+0.6%) and 73.0% (+5.4%) on both languages, which consistently outperforms the baseline. However, the performance is still worse than using gold trees, indicating the necessity of high-quality constituency parsers.

Ablation Study We modify several components of our model to validate their effects. 1) we use vanilla constituent parsing trees by only keeping parent-child edges; 2) we remove the gating mechanism and directly use representations encoded via graph neural networks; 3) we change the way of representing constituent node to initialize only with type embeddings; 4) we incorporate dependency trees rather than constituent trees.[8]

From Table 4.4 we observe that: 1) The dual graph and different edge types show positive impacts in capturing long-range dependencies; 2) Removing the gate mechanism results in the worst performance, which is below the baseline. We suspect that some information such as position embeddings may be lost after the graph attention network; 3) Although only using type embeddings to initialize constituent node representations also yields competitive performance, our span yield enhanced initialization method can capture span-boundary information more effectively; 4) Incorporating dependency syntax instead

[8]The graph is constructed by adding edges between two token nodes starting from head to dependent. Higher order edges and dual graphs are also employed. This is different from the method for encoding dependency trees proposed in Chapter 3, where we do not add dual graphs and higher-order edges.

Dataset	Model	Mention Length					Overall
		1-2	3-4	5-7	8-10	11+	
English	baseline + SpanBERT-base	90.8	83.2	78.5	69.9	63.8	87.4
	our method + SpanBERT-base	**91.4**	**85.0**	**80.6**	**76.1**	**73.7**	**88.7**
Chinese	baseline + BERT-wwm-base	84.4	78.6	71.1	70.0	67.0	79.3
	our method + BERT-wwm-base	**88.7**	**86.5**	**80.8**	**81.0**	**78.7**	**86.0**

Table 4.5: The F1 performance of mention spans with different lengths on the English and Chinese OntoNotes dataset.

Dataset	Methods	Avg. F1	$\Delta F1$
English	Baseline	78.1	-
	Our Method	80.0	+1.9
	Baseline + Mention Filter	77.3	-0.8
Chinese	Baseline	67.6	-
	Our Method	76.3	+8.7
	Baseline + Mention Filter	71.8	+4.2

Table 4.6: Results when utilising syntactic parse trees as mention filter compared to our base model on the English test set.

of constituent syntax achieves inferior performance.[9]

Mention With Different Lengths Table 4.5 shows the performance comparison in terms of different mention lengths on both datasets. As shown in the table, we can observe that our proposed model consistently outperforms the baseline model for both two languages. This indicates that the improved overall performance in the coreference resolution task has benefited largely from better mention detectors, which is consistent with our previous findings in Section 3.6.4 and Lu and Ng (2020). The performance gain is more significant for mentions with longer length on both languages, demonstrating that leveraging constituent syntax is highly effective for modelling long-range dependencies.

[9]Using the combination of constituent and dependency syntax also does not give us a performance boost but requires much more training time.

Constituent Tree as Mention Filter An alternative use of syntax is through constraining mention types. We use the constituent parse tree as hard constraints on top of the baseline to filter out invalid candidate mentions, assuming that only candidate mentions that have matched phrases in the parse tree are valid. We observe that about 99% of gold mentions correspond to a small set of syntactic phrases and POS types.[10] We thus use these two phrase sets as filters to prune unlikely candidate mentions. Table 4.6 shows the corresponding results. We can find that the syntactic constraint harms the performance slightly on the English baseline (-0.8%) but improves the Chinese baseline by 4.2%. However, in both cases this constrained baseline is substantially worse than using the syntax tree as part of our neural model, as proposed in this chapter (with scores of 2.7% and 4.5% lower for English and Chinese, respectively).

4.5 Summary

In this chapter, we successfully leveraged constituent parsing trees with added higher-order edges and dual graphs, which are encoded via bidirectional graph attention networks and our designed information propagation mechanism. Experimental results confirm the superiority of our proposed method with significant improvements achieved against the strong baseline on English and new state-of-the-art performance established on Chinese.

[10]en: 99.63% gold mentions are included in the set of phrases tagged with NP, NML, PRP, PRP$, WP, WDT, WRB, NNP, VB, VBD, VBN, VBG, VBZ, VBP (Wu and Gardner, 2020). zh: the set of VV, NT, PN, DFL, NR, NP, QP, NN covers 99.79% gold mentions.

Chapter 5

Conclusions

In this thesis, we leverage the potential of syntax in the form of dependency trees and constituent parse trees and semantics in the form of semantic role labels in the coreference resolution task. Our empirical results confirm the positive impacts of incorporating syntax and semantics in neural end-to-end coreference resolution models.

In Chapter 2, we walked through the history of coreference resolution, which dates back to the early 1970s. We first gave a detailed description of early coreference resolution systems, which are mainly driven by heuristics and hand-written rules. Next, we talked about hand-designed feature-based statistical machine learning methods for coreference resolution, which were discussed according to different model paradigms. Lastly, we presented recent progress in neural network-based coreference resolution models with detailed model architecture descriptions. The strengths and weaknesses of all reviewed methods in the literature were also thoroughly discussed.

Chapter 3 proposed a heterogeneous graph-based method to improve coreference resolution by effectively incorporating dependency syntax and SRL semantics. Empirically, our proposed method achieves promising results and significantly outperforms a strong baseline (Joshi et al., 2020). We demonstrate that the proposed model can indeed capture those structures to benefit coreference reasoning through our detailed analysis. We also evaluated our model on different mention sub-categories, where results show that it especially gains improvements on traditionally difficult resolution classes. However, the proposed model still suffers and can be easily fooled in resolving pronouns, especially in dialogues and multi-party conversations. We also find that our model is limited by its dependency on

dependency syntactic and SRL parsers, indicating its good performance cannot be easily maintained in real-world settings when gold parses are not available.

In Chapter 4, we first argued that compared to dependency syntax, the constituent syntax is more natural to coreference resolution task because of the embedded explicit span boundary information. We introduced graph-based methods to effectively leverage constituent syntax by adding higher-order edges and interpreting parse trees using different views. We conducted large-scale experiments on English and Chinese, and empirical results show that our model achieves state-of-the-art performance on Chinese while outperforms a strong baseline and our proposed method (Jiang and Cohn, 2021) in Chapter 3 on English by a large margin. Besides, we also analysed the performance of our model on the mention detection subtask. The result confirms that the leveraged constituent syntax is indeed helpful in identifying mention span boundaries. Lastly, we compared our graph-based method with the mention-filter method, which treats constituent parse trees as hard constraints to filter invalid candidate mentions. The result again shows the superiority of our proposed method.

5.1 Future Directions

Although our two proposed methods have shown strong performance, there are still some promising avenues remained to explore for future research. We identify three possible future research directions in this section.

Different Ways to Incorporate Syntax The dependency syntax captures the relations between pairs of words, and the self-attention mechanism (Vaswani et al., 2017) also learns to compute the importance scores of word pairs. Thus, it is possible that we can incorporate the dependency tree into the self-attention head of pretrained language models by encouraging words to attend to their specific head and dependent words corresponding to the syntactic structure of the sentence. Through this method, we expect the self-attention can learn syntax-constrained knowledge, thereby forming syntax-enhanced word representations. Another way to do this would be to generate dependency trees as an auxiliary task using the multi-task learning framework and do so as a neural network with a shared document encoder with the main coreference resolution task. One benefit of this method is

that it does not require extra dependency trees as input at inference time, thus reducing the need for external dependency parsers. In the meantime, the structure of constituent parse trees can also be employed in the self-attention mechanism, although the method may not be as straightforward as that of employing dependency trees. For example, we could use the head-finding rules to find the heads of each internal non-terminal node, and then we can derive the relations between word pairs based on the structure of constituent trees. The performance of these two potential methods can be compared with our proposed one in Chapter 4 to find which method is more effective, and we can also explore which method is more robust under the setting of predicted syntax trees. Furthermore, above mentioned methods can also extend to other NLP tasks to test their generality.

Discourse-Level Features One drawback of our proposed two methods is that the incorporated syntax and semantics are sentence-level features, indicating that inter-sentence information and document-level features are not utilised. However, as a discourse-level task, coreference resolution should benefit from document-level information. The discourse relation captures the relationships between intertwined sentences, where each utterance is considered as a single Elementary Discourse Unit (EDU), and they are linked through specific predefined relation edges. It has been shown positive effects on dialogue understanding like identifying the decisions in multi-party conversations (Bui et al., 2009). As mentioned in our error analysis in §3.7, our model makes most errors in resolving pronouns, especially in dialogues and multi-party conversations with frequent speaker switching. Current neural models often struggle to address the long-range dependency between different utterances (Xu et al., 2020), especially when there are frequent interruptions (e.g., frequent speaker switching). As a result, we aim to evaluate the impact of incorporating the discourse relations for coreference resolution task in the future, especially under the setting of dialogues and email conversations (Dakle and Moldovan, 2020).

Knowledge-Driven Methods The development of large-scale knowledge graphs, such as ConceptNet (Speer et al., 2017) and ATOMIC (Sap et al., 2019), allows neural models to take full advantage of knowledge, typically in the form of triples, for a deeper understanding of documents and making better decisions. Zhang et al. (2019b) successfully incorporated the ConceptNet into a neural pronoun resolution model, but the empirical results show that the benefits of commonsense knowledge are limited in documents of

general domains. Another potential direction is to utilise knowledge facts extracted from input documents. Structural fact-event knowledge graphs, in the form of (subject, predicate, object), capture the relevant concepts of the same entity across multiple sentences. Recent studies (Huang et al., 2020; Chen and Yang, 2021) have demonstrated its effectiveness on abstractive summarisation through a better understanding of factual details in input documents. Thus, for both external knowledge graphs and automatically extracted fact triples from input documents, evaluating their utility for coreference resolution by employing graph-based methods can be an exciting research direction in the future.

Bibliography

1995. *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

1998. *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*.

Chinatsu Aone and Scott William. 1995. Evaluating automated and manual acquisition of anaphora resolution strategies. In *33rd Annual Meeting of the Association for Computational Linguistics*, pages 122–129, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566.

Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1957–1967, Copenhagen, Denmark. Association for Computational Linguistics.

Shane Bergsma and Dekang Lin. 2006. Bootstrapping path-based pronoun resolution. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 33–40, Sydney, Australia. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 47–57, Baltimore, Maryland. Association for Computational Linguistics.

Trung Bui, Matthew Frampton, John Dowding, and Stanley Peters. 2009. Extracting decisions from multi-party dialogue using directed graphical models and semantic similarity. In *Proceedings of the SIGDIAL 2009 Conference*, pages 235–243, London, UK. Association for Computational Linguistics.

Xavier Carreras and Lluís Màrquez. 2004. Introduction to the CoNLL-2004 shared task: Semantic role labeling. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 89–97, Boston, Massachusetts, USA. Association for Computational Linguistics.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar. Association for Computational Linguistics.

Hong Chen, Zhenhua Fan, Hao Lu, Alan Yuille, and Shu Rong. 2018. PreCo: A large-scale dataset in preschool vocabulary for coreference resolution. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 172–181, Brussels, Belgium. Association for Computational Linguistics.

Jiaao Chen and Diyi Yang. 2021. Structure-aware abstractive conversation summarization via discourse and action graphs.

Noam Chomsky. 1988. *Language and Problems of Knowledge: The Managua Lectures*. MIT Press, Cambridge, MA.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. 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 4925–4936.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Kevin Clark and Christopher D. Manning. 2015. Entity-centric coreference resolution with model stacking. 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 1405–1415, Beijing, China. Association for Computational Linguistics.

Kevin Clark and Christopher D. Manning. 2016. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Berlin, Germany. Association for Computational Linguistics.

Aron Culotta, Michael Wick, and Andrew McCallum. 2007. First-order probabilistic models for coreference resolution. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 81–88, Rochester, New York. Association for Computational Linguistics.

Ido Dagan and Alon Itai. 1990. Automatic processing of large corpora for the resolution of anaphora references. In *COLING 1990 Volume 3: Papers presented to the 13th International Conference on Computational Linguistics*.

Parag Pravin Dakle and Dan Moldovan. 2020. CEREC: A corpus for entity resolution in email conversations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 339–349, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. 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 5925–5932.

Pascal Denis and Jason Baldridge. 2008. Specialized models and ranking for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 660–669, Honolulu, Hawaii. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017*.

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.

Niyu Ge, John Hale, and Eugene Charniak. 1998. A statistical approach to anaphora resolution. In *In Proceedings of the Sixth Workshop on Very Large Corpora*, pages 161–170.

Carlos Gómez-Rodríguez and David Vilares. 2018. Constituent parsing as sequence labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1314–1324, Brussels, Belgium. Association for Computational Linguistics.

Barbara J. Grosz. 1977. The representation and use of focus in a system for understanding dialogs. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'77, page 67–76, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1983. Providing a unified account of definite noun phrases in discourse. In *21st Annual Meeting of the Association for*

Computational Linguistics, pages 44–50, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

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

Graeme Hirst. 1981. Anaphora in natural language understanding: A survey. In *Lecture Notes in Computer Science*.

J.R. Hobbs. 1978. Resolving pronoun references. *Lingua 44*, pages 311–338.

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

Luyang Huang, Lingfei Wu, and Lu Wang. 2020. Knowledge graph-augmented abstractive summarization with semantic-driven cloze reward. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5094–5107, Online. Association for Computational Linguistics.

Ryu Iida, Kentaro Inui, Hiroya Takamura, and Yuji Matsumoto. 2003. Incorporating contextual cues in trainable models for coreference resolution. In *Proceedings of the 2003 EACL Workshop on The Computational Treatment of Anaphora*.

Fan Jiang and Trevor Cohn. 2021. Incorporating syntax and semantics in coreference resolution with heterogeneous graph attention network. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1584–1591, Online. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019. BERT for coreference resolution: Baselines and analysis. 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 5803–5808.

Ben Kantor and Amir Globerson. 2019. Coreference resolution with entity equalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 673–677.

Andrew Kehler, Douglas Appelt, Lara Taylor, and Aleksandr Simma. 2004. The (non)utility of predicate-argument frequencies for pronoun interpretation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 289–296, Boston, Massachusetts, USA. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Fang Kong and Fu Jian. 2019. Incorporating structural information for better coreference resolution. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5039–5045. International Joint Conferences on Artificial Intelligence Organization.

Fang Kong and Guodong Zhou. 2011. Combining dependency and constituent-based syntactic information for anaphoricity determination in coreference resolution. In *Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation*, pages 410–419, Singapore. Institute of Digital Enhancement of Cognitive Processing, Waseda University.

Fang Kong, Guodong Zhou, Longhua Qian, and Qiaoming Zhu. 2010. Dependency-driven anaphoricity determination for coreference resolution. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 599–607, Beijing, China. Coling 2010 Organizing Committee.

Fang Kong, GuoDong Zhou, and Qiaoming Zhu. 2009. Employing the centering theory in pronoun resolution from the semantic perspective. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 987–996.

Jonathan K. Kummerfeld and Dan Klein. 2013. Error-driven analysis of challenges in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 265–277, Seattle, Washington, USA. Association for Computational Linguistics.

Sandra Kübler and Desislava Zhekova. 2016. Multilingual coreference resolution. *Language and Linguistics Compass*, 10(11):614–631.

Shalom Lappin and Herbert J. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34, Portland, Oregon, USA. Association for Computational Linguistics.

HEEYOUNG LEE, MIHAI SURDEANU, and DAN JURAFSKY. 2017. A scaffolding approach to coreference resolution integrating statistical and rule-based models. *Natural Language Engineering*, 23(5):733–762.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana. Association for Computational Linguistics.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Fei Liu, Luke Zettlemoyer, and Jacob Eisenstein. 2019. The referential reader: A recurrent entity network for anaphora resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5918–5925, Florence, Italy. Association for Computational Linguistics.

Lu Liu, Zhenqiao Song, and Xiaoqing Zheng. 2020. Improving coreference resolution by leveraging entity-centric features with graph neural networks and second-order inference.

Jing Lu and Vincent Ng. 2020. Conundrums in entity coreference resolution: Making sense of the state of the art. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6620–6631, Online. Association for Computational Linguistics.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3036–3046.

Hongyin Luo and Jim Glass. 2018. Learning word representations with cross-sentence dependency for end-to-end co-reference resolution. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4829–4833, Brussels, Belgium. Association for Computational Linguistics.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla, and Salim Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 135–142, Barcelona, Spain.

Xiaoqiang Luo and Imed Zitouni. 2005. Multi-lingual coreference resolution with syntactic features. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 660–667, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark. Association for Computational Linguistics.

Diego Marcheggiani and Ivan Titov. 2020. Graph convolutions over constituent trees for syntax-aware semantic role labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3915–3928, Online. Association for Computational Linguistics.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8.

Joseph F. McCarthy and Wendy G. Lehnert. 1995. Using decision trees for coreference resolution. *CoRR*, cmp-lg/9505043.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Vincent Ng. 2005. Supervised ranking for pronoun resolution: Some recent improvements. In *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July*

9-13, 2005, Pittsburgh, Pennsylvania, USA, pages 1081–1086. AAAI Press / The MIT Press.

Vincent Ng. 2010. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1396–1411, Uppsala, Sweden. Association for Computational Linguistics.

Vincent Ng and Claire Cardie. 2002a. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Vincent Ng and Claire Cardie. 2002b. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 104–111, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Yuyang Nie, Yuanhe Tian, Yan Song, Xiang Ao, and Xiang Wan. 2020a. Improving named entity recognition with attentive ensemble of syntactic information. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4231–4245.

Yuyang Nie, Yuanhe Tian, Xiang Wan, Yan Song, and Bo Dai. 2020b. Named entity recognition for social media texts with semantic augmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1383–1391.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Simone Paolo Ponzetto and Michael Strube. 2006a. Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 192–199.

Simone Paolo Ponzetto and Michael Strube. 2006b. Semantic role labeling for coreference resolution. In *Demonstrations*.

Amir Pouran Ben Veyseh, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020. Graph transformer networks with syntactic and semantic structures for event argument extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3651–3661.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. CoNLL-2011 shared task: Modeling unrestricted coreference in OntoNotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27, Portland, Oregon, USA. Association for Computational Linguistics.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 492–501, Cambridge, MA. Association for Computational Linguistics.

Altaf Rahman and Vincent Ng. 2009. Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 968–977, Singapore. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016*

Conference on Empirical Methods in Natural Language Processing, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Leonardo F. R. Ribeiro, Claire Gardent, and Iryna Gurevych. 2019. Enhancing AMR-to-text generation with dual graph representations. 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 3183–3194, Hong Kong, China. Association for Computational Linguistics.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:3027–3035.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. Modeling relational data with graph convolutional networks.

Peng Shi and Jimmy Lin. 2019. Simple BERT models for relation extraction and semantic role labeling. *CoRR*, abs/1904.05255.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451. AAAI Press.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks.

Dario Stojanovski and Alexander Fraser. 2018. Coreference and coherence in neural machine translation: A study using oracle experiments. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 49–60.

Veselin Stoyanov and Jason Eisner. 2012. Easy-first coreference resolution. In *Proceedings of COLING 2012*, pages 2519–2534, Mumbai, India. The COLING 2012 Organizing Committee.

Veselin Stoyanov, Nathan Gilbert, Claire Cardie, and Ellen Riloff. 2009. Conundrums in noun phrase coreference resolution: Making sense of the state-of-the-art. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 656–664, Suntec, Singapore. Association for Computational Linguistics.

Nikolaos Stylianou and Ioannis Vlahavas. 2021. A neural entity coreference resolution review. *Expert Systems with Applications*, 168:114466.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory 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 1556–1566, Beijing, China. Association for Computational Linguistics.

Shubham Toshniwal, Allyson Ettinger, Kevin Gimpel, and Karen Livescu. 2020. PeTra: A Sparsely Supervised Memory Model for People Tracking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5415–5428, Online. Association for Computational Linguistics.

Hai-Long Trieu, Anh-Khoa Duong Nguyen, Nhung Nguyen, Makoto Miwa, Hiroya Takamura, and Sophia Ananiadou. 2019. Coreference resolution in full text articles with BERT and syntax-based mention filtering. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*, pages 196–205, Hong Kong, China. Association for Computational Linguistics.

Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A

simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, Uppsala, Sweden. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. pages 45–52.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. 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 5784–5789.

C. Walker and Linguistic Data Consortium. 2005. *ACE 2005 Multilingual Training Corpus*. LDC corpora. Linguistic Data Consortium.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational graph attention network for aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3229–3238.

Minjie Wang, Lingfan Yu, Da Zheng, Quan Gan, Yu Gai, Zihao Ye, Mufei Li, Jinjing Zhou, Qi Huang, Chao Ma, Ziyue Huang, Qipeng Guo, Hao Zhang, Haibin Lin, Junbo Zhao, Jinyang Li, Alexander J. Smola, and Zheng Zhang. 2019a. Deep graph library: Towards efficient and scalable deep learning on graphs. *CoRR*, abs/1909.01315.

W. Wang, Kevin Knight, and D. Marcu. 2007. Binarizing syntax trees to improve syntax-based machine translation accuracy. In *EMNLP-CoNLL*.

Yufei Wang, Mark Johnson, Stephen Wan, Yifang Sun, and Wei Wang. 2019b. How to best use syntax in semantic role labelling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5338–5343, Florence, Italy. Association for Computational Linguistics.

Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. 2018. Mind the gap: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 994–1004, San Diego, California. Association for Computational Linguistics.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. CorefQA: Coreference resolution as query-based span prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6953–6963.

Zhaofeng Wu and Matt Gardner. 2020. Understanding mention detector-linker interaction for neural coreference resolution.

Patrick Xia, João Sedoc, and Benjamin Van Durme. 2020. Incremental neural coreference resolution in constant memory. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8617–8624, Online. Association for Computational Linguistics.

Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. 2015. Empirical evaluation of rectified activations in convolutional network.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031, Online. Association for Computational Linguistics.

Liyan Xu and Jinho D. Choi. 2020. Revealing the myth of higher-order inference in coreference resolution. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8527–8533.

Yinchuan Xu and Junlin Yang. 2019. Look again at the syntax: Relational graph convolutional network for gendered ambiguous pronoun resolution. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 96–101, Florence, Italy. Association for Computational Linguistics.

Xiaofeng Yang, Jian Su, Guodong Zhou, and Chew Lim Tan. 2004. An np-cluster based approach to coreference resolution. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, page 226–es, USA. Association for Computational Linguistics.

Xiaofeng Yang, Guodong Zhou, Jian Su, and Chew Lim Tan. 2003. Coreference resolution using competition learning approach. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 176–183, Sapporo, Japan. Association for Computational Linguistics.

Hongming Zhang, Yan Song, and Yangqiu Song. 2019a. Incorporating context and external knowledge for pronoun coreference resolution. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 872–881, Minneapolis, Minnesota. Association for Computational Linguistics.

Hongming Zhang, Yan Song, Yangqiu Song, and Dong Yu. 2019b. Knowledge-aware pronoun coreference resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 867–876, Florence, Italy. Association for Computational Linguistics.

Rui Zhang, Cícero Nogueira dos Santos, Michihiro Yasunaga, Bing Xiang, and Dragomir Radev. 2018. Neural coreference resolution with deep biaffine attention by joint mention detection and mention clustering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 102–107, Melbourne, Australia. Association for Computational Linguistics.

Yu Zhang, Houquan Zhou, and Zhenghua Li. 2020a. Fast and accurate neural CRF constituency parsing. In *Proceedings of IJCAI*, pages 4046–4053.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020b. Semantics-aware BERT for language understanding. In *the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-2020)*.

Zhuosheng Zhang, Yuwei Wu, Junru Zhou, Sufeng Duan, Hai Zhao, and Rui Wang. 2020c. SG-Net: Syntax-guided machine reading comprehension. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Yanbin Zhao, Lu Chen, Zhi Chen, Ruisheng Cao, Su Zhu, and Kai Yu. 2020. Line graph enhanced AMR-to-text generation with mix-order graph attention networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 732–741, Online. Association for Computational Linguistics.

GuoDong Zhou and Fang Kong. 2009. Global learning of noun phrase anaphoricity in coreference resolution via label propagation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 978–986, Singapore. Association for Computational Linguistics.