Learning To Understand Entities In Text

A dissertation
submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy
University of Washington
2019

Reading Committee:
Luke S. Zettlemoyer, Co-Chair
Yejin Choi, Co-Chair
Daniel S. Weld

Program Authorized to Offer Degree:
Computer Science and Engineering

University of Washington

Abstract

Learning to Understand Entities In Text

Eunsol Choi

Co-Chair of the Supervisory Committee:

Associate Professor Luke Zettlemoyer

Associate Professor Yejin Choi

Computer Science and Engineering

Real world entities such as people, organizations and countries play a critical role in text. Reading text offers rich information about these entities, both explicit, such as historical facts and scientific findings, and implicit, such as social relationships and personal opinions. Automatically extracting rich entity information promises exciting opportunities, such as question answering systems which can reason across facts mentioned in different documents, and analytic models which can help us understand constantly evolving social relations.

This dissertation studies how machines can read natural language text, gather rich entity information, and map this information to a structured format. We present three independent studies, each focusing on different aspects of entity centric text understanding. First, we introduce a semantic parser populating entity attributes embedded in noun phrases to a large scale knowledgebase. Our method addresses the challenges arise from incompleteness in schema (i.e., existing ontologies cannot represent the meaning of many English phrases) and in KB (i.e., most KB misses many facts). Second, we study rich entity categories that can be inferred from the sentence that the entity occurs in. Our new formulation with virtually unrestricted types allows us to expand the standard KB-based training methodology with typing information from Wikipedia definitions and naturally-occurring head-word supervision. Lastly, we introduce a document-level model to infer dynamic entity-entity relationships. Unlike prior work which mostly focused on factual relationships, our work considers sentiment relationship between a pair of entities and presents a model which considers the document and social context jointly. For each of these studies, we address the limited annotated data

challenge via crowdsourcing and/or harvesting large-scale naturally occurring weak supervision. Each study presents a new model and learning framework exploiting new sources of supervision to organize entity information. Together, this thesis expands the scope of information that can be learned about entities from text and points towards future work for entity centric document understanding.

Acknowledgement

I would like to thank my two advisors: Luke Zettlemoyer and Yejin Choi. Luke, I learned so much from you, as a researcher, scientist, manager, and person. You led me to think critically and empirically, and to communicate research clearly, both verbally and in writing. Yejin, I was truly lucky to have you on my side during this journey. You showed me the courage to think outside the box, and how to learn from and care for others. Because of both of you, I can dream of pursuing an academic career. I would also like to thank my committee members, Dan Weld and Emily Bender, for their support throughout my time at UW. Dan, thank you for always reminding me of a bigger picture. I wish to thank Lillian Lee for teaching me principles earlier in my career, as well as showing me a glimpse of how to develop a taste in research.

This work would not have been possible without my collaborators. Tom Kwiatkowski and Yoav Artzi, you helped me get started on research and stay on the path. Omer Levy, Minjoon Seo, Mandar Joshi, Hannah Rashkin, Ge Gao, Chenhao Tan, Xiaochuang Han, Matic Horvat, Jonathan May, Illia Polosukhin, Alexandre Lacoste, Jin Yea Jang, Svitlana Volkova, thank you for broadening my research. Thank you Allen Institute of Artificial Intelligence, for enabling fruitful collaborations with such cool folks, Mohit Iyyer, Mark Yatskar, Hsin-Yuan Huang, Scott Yih, He He, and Percy Liang. Thank you Daniel Marcu and Kevin Knight for my summer at Information Science Institute. I would also like to thank Jonathan Berant, Jakob Uszkoreit, Daniel Hewlett for hosting me at Google, allowing me to experience industry research. I am not attempting to name everyone here (knowing that I would miss some), but I sincerely appreciate friendship, technical discussions and support from the UW community and global NLP community.

Friendship that I have formed during this journey truly transformed me. Caitlin Bonnar, I was blessed to start this journey with you. Jonathan and Danielle Bragg, I appreciate your steady support and care. Mark Yatskar, thank you for more than I can describe. My friends from SBS – Donglok Kim, Jintae Kim,

Diane Kim, Toby Kim, Jamie Namkung, you have been my family in Seattle. Ravi Bhoraskar, Nicholas FitzGerald, Mark Yatskar, Yoav Artzi, Julija Lazautkaite, Ricardo Bruella, Seungyeop Han, Geunhong Park, Junha Rho, Mike Chung, Younghoon Kim, Srini Iyer, Jialin Li, Arunkumar Byravan, Naveen Sharma, Adriana Szekeres, Christen Chen, Robbie Webber, Aaron Walsman, Mia Suh, Lanu Kim, Haejin Lee, Henri Astre, Roy Seo, Sola Park, Siyu Zheng, thank you for sharing laughter, food, drink and stories with me in Seattle. Lindsay Michimoto, Elise deGoede Dorough, Chiemi Yamaoka-Vismale, thank you for building a supportive environment at UW.

My old friends, especially Yeara, Hyemin and Yeji, thank you for supporting me, lessening my frustrations and anxiety over countless phone calls and occasional meet-ups. Lastly, I wish to thank my family. Dasol and Wonjae, you would not know how much your sincere wishes for my future means for me. Yoonsook Jung and Youngkook Choi, I would not stand here without your sacrifices, care, love and prayers.

DEDICATION

To my parents

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Real world entities such as people, organizations and countries play a critical role in text. Reading text offers rich information about entities, such as the categories they belong to, relationships they have with other entities, and events they participate in. Automatically extracting entity information promises exciting opportunities, such as question answering systems which can reason across facts mentioned in different documents, analytic models which can help us understand constantly evolving social relations, and practical applications to handle legal or medical documents.

While documents contain explicitly stated facts about the world and entities inside it, further information is hiding between the lines. Consider the sentence, *The characters in Courtney Maum's new novel are inspired by a historical figure, Peggy Guggenheim.* From context, we can infer that Courtney Maum is a novelist, author and artist. Furthermore, we can also assume that Maum holds a positive opinion of Peggy Guggenheim. Through inference, we learn rich information about entities, including individual entity's attributes and relationships between them. Understanding such implied relationships becomes crucial to support applications like recommendation engines and open ended question answering systems. In this dissertation, we present approaches to extract rich entity information from text, both explicitly stated and implied.

We present three independent studies, each focusing on different aspects of entity centric text understanding. In all three studies, we bring entity information scattered in unstructured text into a structured space. As diversity is the hallmark of human language, the same entity relation can be represented in

multiple surface patterns. For example, there are countless ways to make us infer one entity will have a positive opinion of another entity. Lacking sizable annotated data to cover all these variations, we introduce new ideas to address limited data for each of these studies. Each study presents a new model and learning framework exploiting new sources of supervision to extract entity information. The aspect of entity information each study addresses ranges from static entity attributes to dynamic implied entity-entity relationships. Together, this dissertation expands the scope of information that can be extracted about entities from unstructured text. Tying these diverse aspects of entity attributes and relationships would be crucial for a robust machine reading system which can reason based on the entity information.

1.1 Background

Extracting entity information from large corpus has been framed as knowledgebase (KB) construction. Entities are the basic unit for a KB, often represented as nodes in knowledge graphs. KB schema, i.e. ontology, lists a set of entity types (i.e., categories they belong to) and a set of ways in which entities can be related to one another. Entity-entity relationships and types covered in KB are mostly static and factual, such as <entity X, PARENT, entity Y>, <entity X, NATIONALITY, entity Y>, and <entity X, IS A, TENNIS PLAYER>. Given this schema and raw text, systems first identify entity mentions from text, their types, and then discover semantic relationships between two entities. Most prior work [Mintz et al., 2009; Hoffmann et al., 2011; Riedel et al., 2013] adopted a fixed schema, listing static semantic types and the relationships between entities. A notable exception to this is open information extraction (OpenIE)[Etzioni et al., 2011], where the schema is not predefined but comes from natural language. In this framework, the relations are simply strings of words (usually beginning with a verb), as stated in running text. The OpenIE systems do not unify different surface patterns of the same semantic relation.

KB-based approaches aim to provide a highly accurate and efficient solution which can scan a large corpus for a limited set of relations. After the facts are populated into the schema, inference and reasoning across information in KB is studied as a graph reasoning problem [Neelakantan et al., 2015; Toutanova et al., 2016]. A drawback of KB-based approach is that question answering models based on KB can only retrieve pre-defined relations and attributes.

Reading comprehension (RC) [Hirschman et al., 1999] has been proposed as an alternative way to query

information. Instead of constructing KB from raw text and then using the KB to answer queries, reading comprehension systems take question and evidence document as an input, and generates answers directly. Under this paradigm, the answers can be natural language text, not confined to be a set of entities or concepts in the KB. To facilitate evaluation and make answer space tractable, the task has been framed either as multiple choice [Richardson et al., 2013; Mostafazadeh et al., 2016] or span prediction [Rajpurkar et al., 2016; Joshi et al., 2017] where the answer space is defined as a span in the evidence document. While the questions with span-based answers are more constrained than the multiple choice questions, they can retrieve rich, useful information as long as the answers were stated verbatim in the evidence document. Recent neural models [Devlin et al., 2019] have achieved an impressive progress, reaching human level accuracy on span-based reading comprehension benchmarks. A few limitations of span-based RC are recognized and studied: many RC models focus on surface pattern match between the question and the text [Weissenborn et al., 2017], and current models and datasets have a limited capability to combine pieces of information scattered across multiple documents.

While these two directions are studied relatively independently, we [Levy et al., 2017] recently showed that two approaches are studying a similar phenomena, reducing relation extraction (KB approach) to reading comprehension. We will put aside reading comprehension approach in this thesis, and focus on the structured organization of entity centric information. We study mapping natural language text to pre-specified structured space, following KB construction literature. The structured space is defined as a large scale knowledgebase schema containing relations and types, entity type ontology covering tens of thousands of concepts, and a simplified sentiment relation categories (positive, neutral, and negative). Bringing information in unstructured text into structured space comes with many technical challenges, which we will describe in next section.

1.2 Challenges

Several core challenges emerge when extracting entity information embedded in text to structured representation. This thesis will address three major challenges: implied semantics, incomplete schema, and limited data.

Implied Semantics Language not only conveys facts but also evokes imagery and emotion to readers. Readers actively engage with text and learn more than explicitly stated facts, using world knowledge and common sense. Grasping rich implied semantics, including the relationship between entities and entity types, is onerous for automatic text understanding. There is little lexical overlap between the inferred statements and the text itself, and models should resolve many to many mapping between textual surface patterns and implied semantics. Information that can be inferred from even a single sentence can be extensive, thus scoping the space of implied semantics and framing the research question requires a careful design. Researchers have studied implied semantics through a lens of entailment task [Giampiccolo et al., 2005] or sentiment analysis [Pang et al., 2002]. In this thesis, we will study fine-grained entity categories that can be inferred from the context (Chapter 3) and entity-entity sentiment relationships (Chapter 4). Implied semantics is one pathway leading to robust models that do not rely on surface pattern matches alone.

Incomplete Schema Even a very large KB can capture only a very small fraction of information that can appear in raw text, limited by its schema. For example, consider the phrase “the relaxed, seaside capital of Mozambique." It includes detailed phrases such as “seaside", and subjective phrases such as “relaxed". Both concepts are not found in any existing large scale schemas, such as DBPedia and Freebase, despite being explicitly mentioned in text.

Two research challenges arise when dealing with incomplete schema. First, the models should resolve the mismatch in expressivity between KB and text, distinguishing concepts that lie inside the schema from those that do not. The second challenge is improving schema coverage. To support a variety of applications, extraction systems should handle a large scale schema which covers as many concepts as possible, going beyond a small set of pre-defined relations and entity types. Improving schema coverage requires reasoning across a large label space, which contains concepts that are similar with subtle differences, as well as finding sufficient training data.

Limited Data Gathering large, high quality datasets [Baker et al., 1998; Prasad et al., 2008; Pang et al., 2002] is crucial for progress in the field. It has become even more relevant with newer data intensive models with many parameters. While carefully collected human annotation provides high quality data, annotating large-scale dataset can be prohibitively expensive. To overcome this limitation, distant supervision

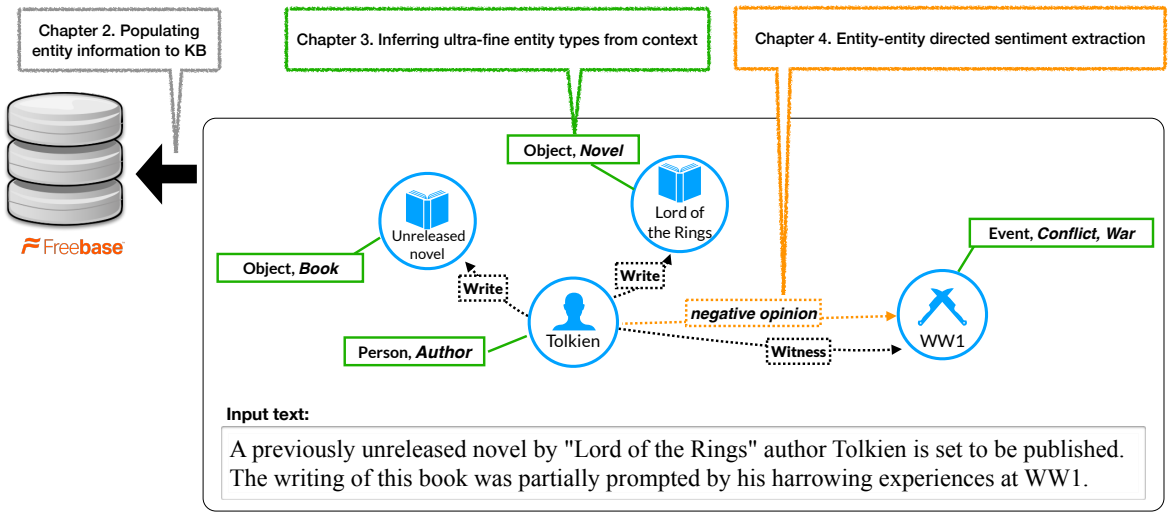


Figure 1.1: Thesis overview: Chapter 2 will present a parser which can map text to schema for KB population, focusing on explicitly mentioned facts. The latter chapters will focus on implied facts and opinions, chapter 3 (green) predicting fine-grained entity types, and chapter 4 (yellow) extracting directed sentiment relationship between a pair of entities.

approaches [Hoffmann et al., 2011; Riedel et al., 2013; Ratner et al., 2019] created large training data by heuristically matching entity relations in the pre-populated knowledgebase to text, and such distant supervision greatly improved the extraction systems. This thesis also follows the idea of distant supervision, and discusses new forms of distant supervision from community constructed resources and observation of linguistic phenomena.

Crowdsourcing platforms such as Mechanical Turk offer a significantly cheaper and faster way to collect non expert annotations. Such crowdsourced datasets [Deng et al., 2009; Rajpurkar et al., 2016] have enabled progress in the field. In this thesis, we design new crowdsourcing tasks which can alleviate the limited data problem, and also investigate how to combine annotated data with distant supervision data during training.

1.3 Thesis Outline

This dissertation studies how machines can read unstructured text, gather information about entities, and map this information into a structured format. Figure 1.1 visualizes three chapters, each focusing on different aspect of entity analysis. Chapter 2 will learn a parser that maps natural language text into knowl-

edgebase schema, following prior KB population work most closely. We will focus on *explicitly stated* fact exclusively, covering binary entity-entity relationship and entity types over a large-scale KB. This semantic parser addresses the incomplete schema challenge, handling the mismatch between unconstructured text and KB schema. Chapter 3 studies learning fine-grained entity types, that can be *inferred* from the sentence that entity appears in. We propose a new formulation, expanding the coverage of a fixed entity type schema from hundreds to tens of thousands of concepts. Lastly, chapter 4 studies the *relationship* between two entities. We focus on a high-level, implied relationship between a pair entities that can be inferred from context, specifically entity-entity directed sentiment relations.

In all aspects of automatic entity analysis, data plays a significant role. We bring innovations in data regime by introducing new sources of distant supervision and carefully designed crowdsourcing tasks. Automatically converting natural language text into a structured representation opens up new applications, and this dissertation broadens the scope of information that can be learned about entities.

1.4 Approach

Here, we will give a brief overview of each chapter. Since there is little overlap in terms of methodology, we discuss them separately.

Knowledgebase Population via Semantic Parser With Partial Ontology Chapter 2 explores how to map natural language text to a KB query, and applying this for KB population for entities. We will focus on parsing noun phrases describing entity category (e.g., Symphonic Poems by Jean Sibelius) to a KB query (e.g., $\lambda \mathtt{x}.\mathtt{composition.form}(\mathtt{x}, \mathtt{Symphonicpoems}) \wedge \mathtt{composer}(\mathtt{JeanSibelius}, \mathtt{x})$). This chapter focuses on limited schema challenge, addressing the limitations of fixed ontology by modeling open concepts, i.e., concepts lies beyond the schema, and KB concepts, jointly.

While our parser has to reason across thousands of concepts in a large-scale KB (Freebase [Bollacker et al., 2008]), we have limited access to annotated data. To address this, we introduce a two-stage learning: first computing broad coverage lexical statistics computed from Wikipedia category pages and their paired entities, which are incorporated as features in a full parsing model.

Ultra-Fine Entity Typing Chapter 3 presents a model to capture contextual cues from a sentence in which entity occurs to predict the semantic types of entities. We expand limited schema from prior work, which considered hundreds of type labels, by introducing an expressive label space (10K) and studying entity mentions in all surface forms. In this rich label space, models should learn rich implied semantics, as many types can only be inferred from its context. To allow predictions in rich label space with limited annotation, we bring novel, large-scale, distantly supervised training set, which also improve the performance of existing fine-grained named entity typing.

Entity Entity Sentiment In Chapter 4, we study high-level entity-entity relationships that can be inferred from a document. We study implied semantics, specifically document-level sentiment graphs, representing entity-to-entity sentiment - i.e., *who* feels positively (or negatively) towards *whom* - and present a model for retrieving such sentiment graphs. Sentiment relationships between entities are often implied, and readers infer a complex web of facts and opinions that hold among the entities. Our model jointly optimizes (1) sentence- and discourse-level sentiment cues (such as a sentiment lexicon linked to entities), (2) factual evidence about which entity belongs to another entity (i.e., Moscow often refers to Russia), and (3) global constraints based on social science theories, such as homophily and social balance theory. To handle challenges from limited data (i.e., lacking document level entity-entity sentiment annotation), we develop a two-stage crowd sourcing scheme and present a new dataset.

Chapter 2

Knowledgebase Population via Semantic Parser With Partial Ontology

In this chapter, we present a semantic parser for a large scale knowledgebase and apply this model for two tasks: entity attribute extraction and referring expression resolution. We follow two stage semantic parser for question answering [Kwiatkowski et al., 2013], but focus on addressing incomplete schema. In prior work, natural language queries were filtered to ensure that the the semantics of a query is completely covered by KB concepts. We address a more challenging scenario, where not all concepts in natural language text can be mapped to KB concepts.

We study this problem on two newly introduced large-scale noun phrase datasets, and present a new semantic parsing model and semi-supervised learning approach for reasoning with partial ontological support. Experiments demonstrate strong performance: our parser can populate KB with up to 12 million facts at 72% accuracy. The partial analyses allow us to improve precision over strong baselines, while parsing many phrases that would be ignored by existing techniques. This chapter is based on the work originally described in Choi et al. [2015], and the data associated with this work can be found at `https://tinyurl.com/yxvpvugk`.

Wikipedia	Haitian human rights activists
	Art museums and galleries in New York
	School buildings completed in 1897
	Olympic gymnasts of Norway
Appos.	the capital of quake-hit Sichuan Province
	a major coal producing province
	the relaxed seaside capital of Mozambique

Figure 2.1: Example noun phrases from Wikipedia category labels and appositives in newswire text.

2.1 Introduction

A significant progress has been made in learning semantic parsers for large knowledge bases (KBs) such as Freebase (FB) [Cai and Yates, 2013; Berant et al., 2013; Kwiatkowski et al., 2013; Reddy et al., 2014]. Although these methods can build general purpose meaning representations, they are typically evaluated on question answering tasks and are designed to only parse questions that have complete ontological coverage, in the sense that there exists a logical form that can be executed against Freebase to get the correct answer.[1] In this chapter, we instead consider the problem of learning semantic parsers for open domain text containing concepts that may or may not be representable using the Freebase ontology.

Even very large knowledge bases have two types of incompleteness that provide challenges for semantic parsing algorithms. They (1) have partial ontologies that cannot represent the meaning of many English phrases and (2) are typically missing many facts. For example, consider the phrases in Figure 2.1. They include subjective or otherwise unmodeled phrases such as “relaxed" and “quake-hit." Freebase, despite being large-scale, contains a limited set of concepts that cannot represent the meaning of these phrases. They also refer to entities that may be missing key facts. For example, a recent study [West et al., 2014a] showed that over 70% of people in FB have no birth place, and 99% have no ethnicity. In our work, we introduce a new semantic parsing approach that explicitly models ontological incompleteness and is robust to missing facts, with the goal of recovering as much of a sentence’s meaning as the ontology supports. We argue that this will enable the application of semantic parsers to a range of new tasks, such as information extraction (IE), where phrases rarely have full ontological support and new facts must be added to the KB.

Because existing semantic parsing datasets have been filtered to limit incompleteness, we introduce

[1] To ensure all questions are answerable, the data is manually filtered. For example, the WebQuestions dataset introduced by Berant et al. [2013] contains only the 7% of the originally gathered questions.

(a) Wikipedia category	x :	Symphonic Poems by Jean Sibelius
	e :	{The Bard, Finlandia,Pohjola's Daughter, En Saga, Spring Song, Tapiola... }
	l_0 :	$\lambda x.Symphonic(x) \wedge Poems(x) \wedge by(JeanSibelius, x)$
	y :	λx.composition.form(x, Symphonicpoems) $\wedge$ composer(JeanSibelius, x)
	x :	Defunct Korean football clubs
	e :	{ Goyang KB Kookmin Bank FC, Hallelujah FC, Kyungsung FC }
	l_0 :	$\lambda x.defunct(x) \wedge korean(x) \wedge football(x) \wedge clubs(x)$
	y :	λx.OpenType[defunct](x) $\wedge$ OpenRel(x, KOREA) $\wedge$ football_clubs(x))
(b) Appos	x :	a driving force behind the project
	e :	Germany
	l_0 :	$\lambda x.driving(x) \wedge force(x) \wedge behind(x, theproject)$
	y :	λx.OpenType[driving_force](x) $\wedge$ OpenRel[behind](x, OpenEntity[the_project])
	x :	an EU outpost in the Mediterranean
	e :	Malta
	l_0 :	$\lambda x.outpost(x) \wedge EU(x) \wedge in(x, theMediterranean)$
	y :	λx.OpenRel(x, EU) $\wedge$ OpenType[outpost](x) $\wedge$ contained_by(x, MediterraneanSea)

Figure 2.2: Examples of noun phrases x, from the Wikipedia category and apposition datasets, paired with the set of entities **e** they describe, their underspecified logical form l_0, and their final logical form y.

two new corpora that pair complex noun phrases with one or more entities that they describe. The first new dataset contains 365,000 Wikipedia category labels (Figure 2.1, top), each paired with the list of the associated Wikipedia entity pages. The second has 67,000 noun phrases paired with a single named entity, extracted from the appositive constructions in KBP 2009 newswire text (Figure 2.1, bottom).This new data is both large scale, and unique in the focus on noun phrases. Noun phrases contain a number of challenging compositional phenomena, including implicit relations and noun-noun modifiers (e.g. see Gerber and Chai [2010]).

To better model text with only partial ontological support, we present a new semantic parser that builds logical forms with concepts from a target ontology and *open* concepts that are introduced when there is no appropriate concept match in the target ontology. Figure 2.2 shows examples of the meanings that we extract. Only the first of these examples can be fully represented using Freebase, all other examples require explicit modeling of open concepts. To build these logical forms, we follow recent work for Combinatory Categorical Grammar (CCG) semantic parsing with Freebase [Kwiatkowski et al., 2013], extended to model when open concepts should be used. We develop a two-stage learning algorithm: we first compute broad coverage lexical statistics over all of the data, which are then incorporated as features in a full parsing model.

The parsing model is tuned on a hand-labeled data set with gold analyses.

Experiments demonstrate the benefits of the new approach. It significantly outperforms strong baselines on both a referring expression resolution task, where much like in the QA setting we directly evaluate if we recover the correct logical form for each input noun phrase, and on entity attribute extraction, where individual facts are extracted from the groundable part of the logical form. We also see that modeling incompleteness significantly boosts precision; we are able to more effectively determine which words should not be mapped to KB concepts.

2.2 Overview

Semantic Parsing with Open Concepts Our goal is to learn to map noun phrase referring expressions x to logical forms y that describe their meaning. In this work, y is built using both concepts from a knowledge base $\mathcal{K}$ and *open concepts* that lie outside of the scope of $\mathcal{K}$. For example, in Figure 2.2 the phrase "Defunct Korean football clubs" is modeled using a logical form y that contains the $\mathcal{K}$ concept $\texttt{footbal\_clubs}(\texttt{x})$ as well as the open concepts $\texttt{OpenType}[\texttt{defunct}](\texttt{x})$.

In this chapter we describe a new method for learning the mapping from x to y from corpora of referring expression noun phrases, paired with a sets of entities e that these referring expressions describe. Figure 2.2 shows examples of these data drawn from two sources.

Tasks We introduce two new datasets (Sec. 2.3) that pair referring noun phrases x with one or more entities $\mathbf{e}$ that they describe. These data support evaluation for two tasks: referring expression resolution and information extraction.

In referring expression resolution, the parser is given x and is used to predict the referring expression logical form y that describes $\mathbf{e}$. Since the majority of our data cannot be fully modeled with Freebase, we evaluate each y against a hand labeled gold standard instead of trying to extract $\mathbf{e}$ from $\mathcal{K}$.

The entity attribute extraction task also involves mapping phrases x to logical forms y, with the goal of adding new facts to the knowledge base $\mathcal{K}$. To do this, we assume each x is additionally paired with an set of entities $\mathbf{e}$. We also define an *entity attribute* to be a literal in y that uses only concepts from $\mathcal{K}$. Finally, we extract, for each entity in $\mathbf{e}$, all of the attributes listed in y. For example, the first logical form y in Figure 2.2

has two entity attributes: $\texttt{composer}(\texttt{JeanSibelius}, \texttt{x})$ and $\texttt{composition.form}(\texttt{x}, \texttt{Symphonic\_poems})$ which can be added to $\mathcal{K}$ for the entities $\{\texttt{TheBard}, \texttt{Finlandia}\}$.

Model and Learning Our approach extends the two-stage semantic parser introduced by Kwiatkowski et al 2013. We use CCG to build domain-independent logical forms l_0 and then introduce a new method for reasoning about how to map this intermediary representation onto both open concepts and $\mathcal{K}$ concepts (Sec. 2.4).

To learn this model, we assume access to data with two different types of annotations. The first contains noun phrase descriptions x and described entity sets e (as in Figure 2.2), which can be easily gathered at scale with no manual data labeling effort. However, this data, in general, has significant amount of knowledge base incompleteness; many described concepts and entity attributes will be missing from $\mathcal{K}$ (see Sec. 2.3 for more details). Therefore, to support effective learning, we will also use a small hand-labeled dataset containing x, $\mathbf{e}$, a gold logical form y, an intermediary CCG logical form l_0, and a mapping from words in x to constants in $\mathcal{K}$ and open concepts. Our full learning approach (Sec. 2.5) estimates a linear model on the small labeled dataset, with broad coverage features derived from the larger dataset.

2.3 Data

We gathered two new datasets that pair complex noun phrases with one or more Freebase entities. After our model parses the noun phrase into a logical form, this dataset can be used to populate knowledgebase automatically.

The Wikipedia category dataset contains 365,504 Wikipedia category names paired with the list of entities in that category.[2] Table 2.3a shows the details of this dataset and examples are given in Figure 2.2. For each development and test data, we randomly select 500 categories consisted of 3-10 words and describing fewer than 100 entities.

[2]Compiled by the YAGO project, available at: www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/

The apposition dataset is a large set of complex noun phrases paired with named entities, extracted from appositive constructions such as "Gustav Bayer, a former Olympic gymnast for Norway." For this example, we extract the entity "Gustav Bayer" and pair it with the noun phrase "a former Olympic gymnast for Norway." To identify appositive constructions, we ran the Stanford dependency parser on the newswire section of the KBP 2009 source corpus,[3] and selected noun phrases composed of 3 to 10 words, starting with an article, and paired with a named entity that is in Freebase.

This procedure of identifying complex entity descriptions allows for information extraction from a wide range of sources. However, it is also noisy and challenging. The dependency parser makes errors, for example "the next day against the United States, Spain" is falsely detected as an apposition. Furthermore, addressing context and co-reference is often necessary. For example, "Puerto Montt, a city south of the capital" or "the company's parent, Shenhua Group" requires reference resolution. We gathered 67 thousand appositions, which will be released to support future work, and randomly selected 300 for testing.

Measuring Incompleteness To study the amount of incompleteness in this data, we hand labeled logical forms for 500 Wikipedia categories in the development set. Examples of annotations are given in the rows labeled y in Figure 2.2. We use these to measure the schema and fact coverage of Freebase. Many of the entities in this dataset do not have the Freebase attributes described by the category phrases. When a concept is not in Freebase, we annotate it as `OpenType` or `OpenRel`, as shown in Figure 2.2. On average, each Wikipedia category name describes 2.58 Freebase attributes, and 0.39 concepts that cannot be mapped to FB. Overall, 27.2% of the phrases contain concepts that do not exist in the Freebase schema.

Each category may have multiple correct logical forms. For example, "Hotels" can be mapped to: $\texttt{hotel}(\texttt{x})$, $\texttt{accomodation.type}(\texttt{x}, \texttt{hotel})$, or $\texttt{building\_function}(\texttt{x}, \texttt{hotel})$. There are also genuine ambiguities in meaning. For example, "People from Bordeaux" can be interpreted as $\texttt{people}(\texttt{x}) \wedge \texttt{place\_lived}(\texttt{x}, \texttt{Bordeaux})$ or $\texttt{people}(\texttt{x}) \wedge \texttt{place\_of\_birth}(\texttt{x}, \texttt{Bordeaux})$. We made a best effort attempt to gather as many correct logical forms as possible, finding on average 1.8 logical forms per noun phrase. There were 97 unique binary relations, and 247 unique unary attributes in the annotation.

Given these logical forms, we also measured factual coverage. For the 72.8% of phrases that can be completely represented using Freebase, we executed the logical forms and compared the result to the labeled

[3] http://www.nist.gov/tac/2009/

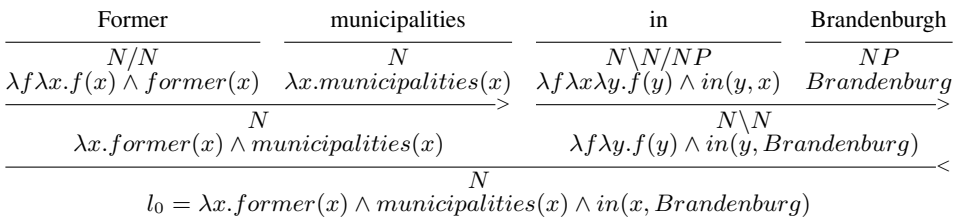
	entire set	dev	test
# categories	365,504	500	500
# words per category	4.1	4.4	4.3
# unique words	84,996	1,100	1,063
# entities per category	19.9	19.1	18.7
# entities	2,813,631	9,511	9,281
# entity-category pairs	7,292,326	9,549	9,331

(a) Data statistics.

	entire set	test set
# appositions	66,924	300
# unique words	25,472	817
# words per apposition	5.73	5.93

(b) Appositive data statistics.

(a) CCG parse builds an underspecified semantic representation of the sentence.



(b) Constant matches replace underspecified constants with Freebase concepts

$\mathsf{l_0} = \lambda x.former(x) \wedge municipalities(x) \wedge in(x, Brandenburg)$

$\mathsf{l_1} = \lambda x.former(x) \wedge municipalities(x) \wedge in(x, \texttt{Brandenburg})$

$\mathsf{l_2} = \lambda x.former(x) \wedge municipalities(x) \wedge \texttt{location.containedby}(x, \texttt{Brandenburg})$

$\mathsf{l_3} = \lambda x.former(x) \wedge \texttt{OpenRel}(x, \texttt{Municipality}) \wedge \texttt{location.containedby}(x, \texttt{Brandenburg})$

$\mathsf{l_4} = \lambda x.\texttt{OpenType}(x) \wedge \texttt{OpenRel}(x, \texttt{Municipality}) \wedge \texttt{location.containedby}(x, \texttt{Brandenburg})$

Figure 2.4: Derivation of the analysis for "Former municipalities in Brandenburgh". This analysis contains a placeholder type and a placeholder relation as described in Section 2.4.

entity set. In total, 56% of the queries returned no entities and those that did return results have on average 15% overlap with the Wikipedia entity set. We also measured how often attributes from the labeled logical forms were assigned to the Wikipedia entities in FB, finding that only 33.6% were present. Given this rate, we estimate that it is possible to add 12 million new facts into FB from the 7 million entity-category pairs.

2.4 Mapping Text to Meaning

We adopt a two-stage semantic parsing approach [Kwiatkowski et al., 2013]. We first use a CCG parser to define a set $\text{CCG}(x)$ of possible logical forms l_0. Then we will choose the logical form l_0 that closely matches the linguistic structure of the input text x, according to a learned linear model, and use an ontological match

step that defines a set of transformations $\text{ONT}(l_0, \mathcal{K})$ to map this meaning to a Freebase query y. Figure 2.2 shows examples of x, l_0 and y. In this section we describe our approach with the more detailed example derivation in Figure 2.4. We also describe the parameterization of a linear model that scores each derivation.

CCG parsing We use a CCG [Steedman, 1996] semantic parser [Kwiatkowski et al., 2013] to generate an underspecified logical form l_0. Figure 2.4a shows an example parse. The constants $Former$, $Municipalities$, in, $Brandenburgh$ in l_0 are not tied to the target knowledge base, causing the logical form to be underspecified. They can be replaced with Freebase constants in the later ontology matching step.

Ontological Matching The ontological match step has *structural match* and *constant match* components. Structural match operators can collapse or expand sub-expressions in the logical forms to match equivalent typed concepts in the target knowledge base. We adopt existing structural match operators [Kwiatkowski et al., 2013] and refer readers to that work for details.

Constant match operators replace underspecified constants in the underspecified logical form l_0 with concepts from the target knowledge base. There are four constant match operations used in Figure 2.4. The first two constant matches, shown below, match underspecified constants with constants of the same type from Freebase.

$$in \rightarrow \texttt{location.containedby}$$

$$Brandenburgh \rightarrow \texttt{BRANDENBURGH}$$

However, because we are modeling the semantics of phrases that are not covered by the Freebase schema, we also require the following two constant matches:

$$Former(x) \rightarrow \texttt{OpenType}$$

$$municipalities(x) \rightarrow \texttt{OpenRel}(\texttt{x}, \texttt{Municipality})$$

Here, the word 'former' has been associated with a placeholder typing predicate since Freebase has no way

of expressing end dates of administrative divisions. There is also no Freebase type representing the concept 'municipalities.' However, this word is associated with an entity in Freebase. Since there is no suitable linking predicate for the entity `Municipality`, we introduce a placeholder linking predicate `OpenRel` in the step from $l_2 \rightarrow l_3$. Our constant match operators can also introduce placeholder entities `OpenEntity` when there is no good match in Freebase.

We also allow the creation of typing predicates from matched entities through the introduction of linking predicates. For example, there is no native type associated with the word 'actor' in Freebase. Instead we create a typing predicate by matching the word to a Freebase entity `Actor` using Freebase API and allowing the introduction of linked predicates such as `person.profession` :

$$actor(x) \rightarrow \texttt{person.profession}(\texttt{x}, \texttt{Actor})$$

Scoring Full Parses Our goal is to learn a function from the phrase x to the correct analysis y. We score each parse using a linear model with features that signal attributes of the underspecified parse ϕ_p and those that signal attributes of the ontological match ϕ_{ont}. Since the model factors over the two stages of parser, we split the prediction problem similarly. First, we select the maximum scoring underspecified logical form:

$$l^* = \arg \max_{l \in \text{CCG}(x)} (\theta_p \cdot \phi_p(l))$$

and then we select the highest scoring Freebase analysis y^* that can be built from l^*:

$$y^* = \arg \max_{r \in \text{ONT}(l^*, \mathcal{K})} (\theta_{ont} \cdot \phi_{ont}(r))$$

We describe an approach to learning the parameter vectors θ_p and θ_{ont} below.

2.5 Learning

We introduce a learning approach that first collates aggregate statistics from the 7 million Wikipedia entity-category pairs and existing facts in FB, and then uses a small labeled training set to tune the weights for features that incorporate these statistics.

Wikipedia Category	
Wars involving the Grand Duchy of Lituania	
Entity	Attribute
BattleOfGrunwald	type(x, military.conflict)
GollubWar	type(x, military.conflict)
BattleOfGrunwald	time.event.loc(x, Grunwald)
...	...
Entity	Relation
BattleOfGrunwald	military_conflict.combatants
GollubWar	time.event.start_time
BattleOfGrunwald	military_conflict.commanders
...	...

Figure 2.5: Labeled entities are associated with attributes and relations.

Broad Coverage Lexical Statistics Each Wikipedia category is associated with a number of entities, most of which exist in FB. We use these entities to extract relations and attributes in FB associated with that category. For example, in Figure 2.5 the category 'Wars involving the Grand Duchy of Lithuania' is associated with the relation military_conflict.combatants and the attribute type(x, military.conflict) multiple times, because they are present in many of the category's entities. For each of the sub-phrases in the category name we count these associations over the entire Wikipedia category set.

We use these counts to calculate Pointwise Mutual Information (PMI) between words and Freebase attributes or relations. We choose PMI to avoid overcompensating common words, attributes, or relations. For example, the word 'Wars' is seen with the incorrect analysis type(x, time.event) more frequently than the correct analysis type(x, military.conflict). However, PMI penalizes the attribute type(x, time.event) for its popularity and the correct analysis is preferred. As PMI has a tendency to emphasize rare counts, we chose PMI squared, which takes the squared value of the co-occurence count (PMI^2(a, b) = $\log\frac{count(a \wedge b)^2}{count(a) * count(b)}$), as a feature.

Structural KB Statistics Existing semantic parsers typically make use of type constraints to limit the space of possible logical forms. These strong type constraints are not feasible when the knowledge base is incomplete. For example, in Freebase the relation military_conflict.combatants expects an entity of type military_conflict.combatant as its object. However, many countries that have been involved in wars are not assigned this type.

We instead calculate type overlap statistics for all Freebase entities, to find likely missing types. For example, including the fact that the object of `military_conflict.combatants` is very often of type `location.country`.

Learning from Labeled Data We train each half of the prediction problem separately, as defined in Section 2.4, using the labeled training data introduced in Section 2.3. We use structured max-margin perceptrons to learn feature weights for both the underspecified parse and the ontological match step following [Kwiatkowski et al., 2013]. The aggregate statistics collected from 7 million category-entity pairs produce very useful lexical features. We integrate these statistics into our linear model by summing their values for each derivation and treating them as a feature. All of the other features described in Section 2.6 are not word specific and are therefore far less sparse.

2.6 Features

We include a number of features that enable soft type checking on the output logical form, described first below, along with other features that measure different aspects of the analysis.

Coherency features For example, consider the phrase "The UK home city of the Queen," with Freebase logical form $y = \lambda \mathtt{x}.\mathtt{home}(\mathtt{QEII}, \mathtt{x}) \wedge \mathtt{in}(\mathtt{x}, \mathtt{UK}) \wedge \mathtt{city}(\mathtt{x})$. Each of the relations has expected types for their argument: the relation $\langle \mathtt{home} \rangle$ expects a subject of type $\langle \mathtt{person} \rangle$ and an object of type $\langle \mathtt{location} \rangle$. Each type in Freebase lives in a hierarchy, so the type `city` implies $\{\mathtt{location}, \mathtt{admin\_division}, ...\}$. The next four features test agreement of these types on different parts of the output logical form.

Relation arguments trigger a feature if their type is in the set of types expected by the relation. `QEII` is a person so this feature is triggered for the relation-argument application in $\mathtt{home}(\mathtt{QEII}, \mathtt{x})$.

Relation - Relation pairs can share variable arguments. For example, the variable x is the object of $\langle \mathtt{home} \rangle$ and the subject of $\langle \mathtt{in} \rangle$. Each relation expects a set of types of x. We have features to signal if: these sets are disjoint; one set subsumes the other; and the PMI between the highest level expected type (described in Section 2.5) if the sets are disjoint. In the example given here, the type $\langle \mathtt{location} \rangle$ expected by $\langle \mathtt{in} \rangle$

subsumes the type $\langle \texttt{city} \rangle$ expected by $\langle \texttt{home} \rangle$ so the second feature fires. We treat types such as $\texttt{city}(x)$ as unary relations and include them in this feature set.

Type domain measures compatibility among domains in Freebase. Freebase is split into high-level domains and some of these are relevant, such as 'football' and 'sports'. We identify those by counting their co-occurrences. This becomes an indicator feature that signals their co-occurrence in y.

Named entity type features test if the entity e that we are extracting attributes for have Freebase type "person", "location" or "organization". If it does, we have a feature indicating if y defines a set of the same type. This features is not used in the referring expression task presented in Section 2.7 since we cannot assume access to the entities that are described.

CCG parse feature signals which lexical items were used in the CCG parse. Another feature fires if capitalized words map to named entities.

String similarity features signal exact string match, stemmed string match, and length weighted string edit distance between a phrase in the sentence and the name of the Freebase element it was matched on. We also use the Freebase search API to generate scores for phrase, entity pairs and include the log of this score as a features.

Lexical PMI feature includes the lexical Pointwise Mutual Information described in Section 2.5.

Freebase constant features signal the use of linking predicates, as defined in Section 2.4, and the log frequency count of the Freebase attributes across all entities in the Wikipedia category set.

Other features indicate the use of `OpenRel`, `OpenEntity`, `OpenType` in y and count repetitions of Freebase concepts in y.

2.7 Experiments

2.7.1 Experimental Setup

Knowledge base We use the Jan. 26, 2014 Freebase dump. After pruning binary predicates taking numeric values, it contains 9351 binary predicates, 2754 unary predicates, and 1.2 billion assertions.

Pruning and Feature Initialization We perform beam search at each semantic parsing stage, using the Freebase search API to determine candidate named entities (10 per phrase), binary predicates (300 per phrase), and unary predicates (500 per phrase). The ontology matching stage considers the highest scored underspecified parse.

The features are initialized to prefer well-typed logical forms. Type checking features are initially set to -2 for mismatch. Features signalling incompatible topic domains and repetition are initialized as -10. All other initial feature weights are set to 1.

Datasets and Annotation We evaluate on the Wikipedia category and appositive datasets introduced in Sec. 2.3. On the Wikipedia development data, we annotated 500 logical forms, underspecified logical forms and constant mappings for ontology matching. The Wikipedia test data is composed of 500 unseen categories. We did not train on the appositive dataset, as it contains challenges such as co-reference and parsing errors as described in Sec. 2.3. Instead, we chose 300 randomly selected examples for evaluation, and ran on the model trained on the Wikipedia development data.

Evaluation Metrics We report five-fold cross validation for development but ran the final model once on the test data, manually scoring the output.

For evaluation on the referring expression resolution performance (as defined in Sec. 2.2), we include accuracy for the final logical form (*Exact Match*). We also evaluate precision and recall for predicting individual literals in this logical form on the development set. To control for missing facts, we did not evaluate the set of returned entities.

To evaluate entity attribute extraction performance (as defined in Sec. 2.2), we identified three classes of predictions. Extractions can be correct, benign, or false. Correct attributes are actually described in the

System	Exact Match	Partial Match		
		P	R	F1
KCAZ13	1.4	9.6	6.3	7.0
IE Baseline	6.8	37.0	23.3	28.6
No PMI	11.0	23.7	20.8	21.6
No OpenSchema	13.7	35.8	30.0	31.1
No Typing	9.6	37.6	29.3	31.8
Our Approach	15.9	39.3	33.5	35.1
with Gold NE	20.8	46.6	40.5	42.3

Table 2.1: Referring expression resolution performance on the development set on gold references.

Data	System	Exact Match Accuracy
Wikipedia	IE Baseline	21.8%
	Our Approach	28.4%
Appos	IE Baseline	0.0%
	Our Approach	4.7%

Table 2.2: Manual evaluation for referring expression resolution on the test sets.

phrase, benign extraction may not have been described but are still true, and false extractions are not true. For example, if the phrase “the capital of the communist-ruled nation" is mapped to the pair of attributes `capital_of_administrative_division`(`x`) , `location`(`x`), the first is correct and the second is benign. Other incorrect facts would be false.

On the development set, we report precision and recall against the union of the FB attributes in our annotations without adjusting for benign extractions or the fact that the annotations are not complete. For the test sets, we computed precision (P) where benign extractions are considered to be wrong, as well as an adjusted precision metric (P*) where benign extractions are counted as correct. As we do not have full test set annotations, we cannot report recall. Finally, we report the average number of facts extracted per noun phrase (fact #).

Comparison Systems We compare performance to a number of ablated versions of the full system, where we have removed the open-constant ontology matching operators (NoOpenSchema), the PMI fea-

System	Top n	P	R	F1	fact #
IE Baseline	-	37.3	26.5	30.6	1.6
Our Approach	1	44.2	32.8	37.7	1.9
	2	36.9	38.0	37.5	2.6
	3	30.7	42.7	35.7	3.6
	4	27.0	44.7	33.6	4.2
	5	23.7	47.2	31.6	5.1
	10	15.9	52.0	24.3	8.5

Table 2.3: Entity attribute extraction performance on the Wikipedia category development set.

Data	System	P	P*	fact #
Wikipedia	IE Baseline	56.7	58.7	1.6
	Our Approach	61.2	72.6	2.0
Appos	IE Baseline	4.9	13.9	1.3
	Our Approach	33.2	61.4	0.9

Table 2.4: Manual evaluation for entity attribute extraction on the test sets.

tures (NoPMI), or the type checking features (NoTyping). For the referring expression resolution task, we excluded the named entity type feature, as this assumes typing information about the entity we are extracting attributes for.

We report results without the PMI features and the open schema matching operators (KCAZ13), which is a reimplementation of a recent Freebase QA model [Kwiatkowski et al., 2013]. We also learn with gold named entity linking (Gold NE).

For the entity attribute extraction, we built a supervised learning baseline that combines the output of two discrete SVMs, one for predicting unary relations and one for binary relations. Each classifier is trained using the annotated Wikipedia categories. This dataset contains hundreds of unary and binary relations, which the IE baseline can predict. Each classifier is further anchored on a specific word, and includes n-gram and POS context features around that word, following features from Mintz et al Mintz et al. [2009]. To predict binary relations, we used named entities as anchors. For unary attributes we anchored on all possible nouns and adjectives. The final logical form includes the best relation predicted by each classifier. We use

the Stanford CoreNLP[4] toolkit for tokenization, named entity recognition, and part-of-speech tagging.

2.7.2 Results

Tables 2.1 and 2.2 show performance on the referring expression resolution task. Tables 2.3 and 2.4 show performance on the extraction task. Reported precision is lower on the labeled development set than on the test set, where predicted logical forms are manually evaluated. This reflects the fact that, despite our best attempts, the development set labels are incomplete, as discussed in Section 2.3.

Referring expression resolution The systems retrieve the full meaning with 28.4% accuracy on the Wikipedia test set, and 15.9% on the development set. The gold named entity input improves performance by modest amounts. This suggests that the errors stem from ontology mismatches, as we will describe in more detail later in the qualitative analysis. We also see that all of the ablations hurt performance, and that the KCAZ13 model performs extremely poorly. The independent classifier baseline performs well at the sub-clause level, but fails to form a full logical form of the referring expression. Partial grounding and broad-coverage data statistics are essential for this problem.

Entity attribute extraction In the two test sets, the approach achieves high benign precision levels (P*) of 72.6 and 61.4. However, the appositives data is significantly more challenging, and the model misses many of the true facts that could be extracted. Many errors comes in the early stages of the pipeline, which can be attributed at least in part to both (1) the higher levels of noise in the input data (see Section 2.3), and (2) the fact that the CCG parser was developed on the Wikipedia category labels. While the IE baseline performs reasonably on the Wikipedia test data, its performance degrades significantly on appositions. As it is trained to predict pre-determined relations, it does not generalize to different domains.

For the development set, Table 2.3 also shows the precision-recall trade off for the set of Freebase attributes that appear in the top-n predicted logical forms. Precision drops quickly but recall can be improved significantly, showing that the model can produce many of the labeled facts.

[4] http://nlp.stanford.edu/software/corenlp.html

Qualitative evaluation We sampled 100 errors from the Wikipedia test set for qualitative analysis. 10% came from entity linking. About 30% come from choosing a superset or subset of the desired meaning, for example by mapping “novel" to `book`. About 10% of the errors are from domain ambiguity, such as mapping “stage actor" to `film.film_actor`. 10% of the cases are from spurious string similarity, such as mapping “Hungarian expatriates“ to $\texttt{nationality}(\texttt{x}, \texttt{Hungary})$. 15% of the failures were due to incorrect under-specified logical forms and, finally, about 10% of the errors were because the typing features encouraged compound nouns to be split into separate attributes. On the apposition dataset, 65% of errors stems from parsing, either in apposition detection or CCG parsing. Better modeling the complex attachment decisions for the noun phrases in the apposition dataset remains an area for future work.

One advantage of our approach, especially in comparison to classifier based models like the IE baseline, is the ability to predict previously unseen relations. Counting only the correctly predicted triples, we see that over 40% of the unique relations we predict is not in the development set; our model learns to generalize based on the learned PMI features and other lexical cues.

Our approach extracted 2.0 entity attributes per Wikipedia phrase and 0.9 per apposition on average. This matches our intuition that the apposition dataset contains many more words that cannot be modeled with concepts in Freebase.

2.8 Summary

In this chapter, we present a semantic parser which handles knowledge base incompleteness, applied to the problem of information extraction from noun phrases. When run on all of the Wikipedia category data, the approach would extract up to 12 million new Freebase facts at 72% precision. While the results were encouraging on the Wikipedia category dataset, the performance on the apposition dataset was much lower. For this general domain, filtering phrases that do not constitute valid entity categories and gathering better distant supervision might be necessary to achieve high accuracy. It would be interesting to gather data with compositional phenomena, such as negation, conjunction and disjunction, and study its impact on the performance of the semantic parser.

This semantic parsing approach aims to map the natural language text to equivalent KB concepts without exploring implied semantics. Our approach can handle a large scale knowledgebase and address the chal-

lenges arising from limited data and incomplete schema. In the next chapter, we study a more challenging objective, i.e. predicting a set of entity attributes that can be *inferred* from a sentence the entity occurs in.

Chapter 3

Ultra-Fine Entity Typing

In the preceding chapter, we introduced a semantic parser which extracts static entity attributes embedded in noun phrases describing entity categories. By studying Wikipedia categories describing entities, we extracted global, context independent entity attributes, such as encyclopedic entity facts. An example global entity attribute is the nationality of an entity, which does not change based on the context. In this chapter, we study context sensitive entity typing: given a sentence with an entity mention, the goal is to predict a set of noun phrases (e.g. skyscraper, songwriter, or criminal) that describe appropriate types for the target entity implied from its context. This formulation allows us to study a richer set of entity types and semantic roles from the context (e.g., defendant, supporter). This new formulation also enables us to use a new type of distant supervision at large scale: head words, which indicate the type of the noun phrases they appear in.

We show that these ultra-fine types can be crowd-sourced, and introduce new evaluation sets that are much more diverse and fine-grained than existing benchmarks. We present a model that can predict diverse types, and is trained using a multitask objective that pools our new head-word supervision with prior supervision from entity linking. Experimental results demonstrate that our model is effective in predicting entity types at varying granularity; it achieved state of the art performance on an existing fine-grained entity typing benchmark, and sets baselines for our newly-introduced typing task. This chapter is based on the work originally described in Choi et al. [2018]. The data and model associated with this work is available at `http://nlp.cs.washington.edu/entity_type`.

Sentence with Target Entity	Entity Types
During the Inca Empire, {**the Inti Raymi**} was the most important of four ceremonies celebrated in Cusco.	event, festival, **ritual, custom, ceremony, party, celebration**
{**They**} have been asked to appear in court to face the charge.	person, **accused, suspect, defendant**
Ban praised Rwanda's commitment to the UN and its role in {**peacemaking operations**}.	event, **plan, mission, action**

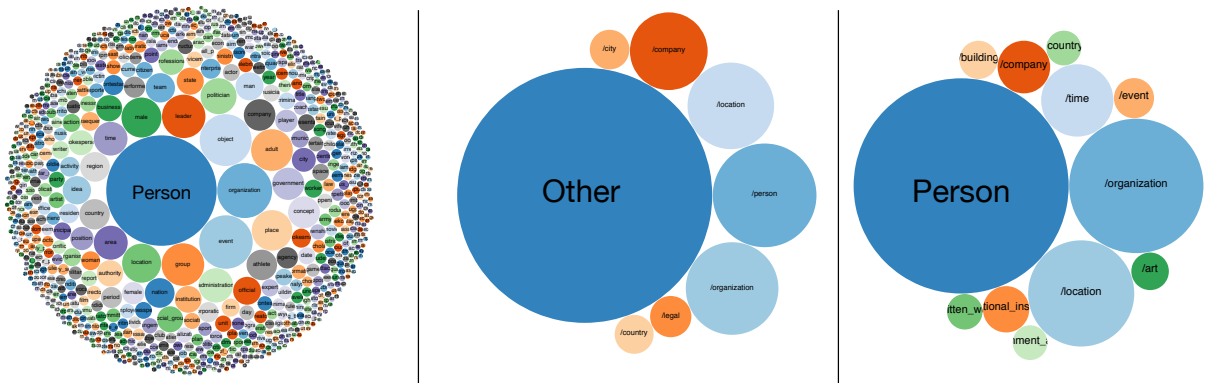
Table 3.1: Examples of entity mentions and their annotated types, as annotated in our dataset. The entity mentions are bold faced and in the curly brackets. The bold blue types do not appear in existing fine-grained type ontologies.

3.1 Introduction

Entities can often be described by very fine grained types. Consider the sentences "Bill robbed John. He was arrested." The noun phrases "John," "Bill," and "he" have very specific types that can be inferred from the text. This includes the facts that "Bill" and "he" are both likely "criminal" due to the "robbing" and "arresting," while "John" is more likely a "victim" because he was "robbed." Such fine-grained types (victim, criminal) are important for context-sensitive tasks such as coreference resolution and question answering (e.g. "Who was the victim?"). Inferring such types for each mention (John, he) is not possible given current typing models that only predict relatively coarse types and only consider named entities.

To address this challenge, we present a new task: given a sentence with a target entity mention, predict free-form noun phrases that describe appropriate types for the role the target entity plays in the sentence. Table 3.1 shows three examples that exhibit a rich variety of types at different granularities. Our task effectively subsumes existing fine-grained named entity typing formulations due to the use of a very large type vocabulary and the fact that we predict types for all noun phrases, including named entities, nominals, and pronouns.

Incorporating fine-grained entity types has improved entity-focused downstream tasks, such as relation extraction [Yaghoobzadeh et al., 2017], question answering [Yavuz et al., 2016], query analysis [Balog and Neumayer, 2012], and coreference resolution [Durrett and Klein, 2014]. These systems used a relatively coarse type ontology. However, manually designing the ontology is a challenging task, and it is difficult to cover all possible concepts even within a limited domain. This can be seen empirically in existing datasets, where the label distribution of fine-grained entity typing datasets is heavily skewed toward coarse-grained



a) Our Dataset b) OntoNotes c) FIGER

Figure 3.1: A visualization of all the labels that cover 90% of the data, where a bubble's size is proportional to the label's frequency. Our dataset is much more diverse and fine grained when compared to existing datasets (OntoNotes and FIGER), in which the top 5 types cover 70-80% of the data.

types. For instance, annotators of the OntoNotes dataset [Gillick et al., 2014] marked about half of the mentions as "other," because they could not find a suitable type in their ontology (see Figure 3.1 for a visualization and Section 3.2.2 for details).

Our more open, ultra-fine vocabulary, where types are free-form noun phrases, alleviates the need for hand-crafted ontologies, thereby greatly increasing overall type coverage. To better understand entity types in an unrestricted setting, we crowdsource a new dataset of 6,000 examples. Compared to previous fine-grained entity typing datasets, the label distribution in our data is substantially more *diverse* and *fine-grained*. Annotators easily generate a wide range of types and can determine with 85% agreement if a type generated by another annotator is appropriate. Our evaluation data has over 2,500 unique types, posing a challenging learning problem.

While our types are harder to predict, they also allow for a new form of contextual distant supervision. We observe that text often contains cues that explicitly match a mention to its type, in the form of the mention's head word. For example, "the incumbent chairman of the African Union" is a type of "chairman." This signal complements the supervision derived from linking entities to knowledge bases, which is context-

oblivious. For example, “Clint Eastwood” can be described with dozens of types, but context-sensitive typing would prefer “director” instead of “mayor” for the sentence “Clint Eastwood won ‘Best Director’ for Million Dollar Baby."

We combine head-word supervision, which provides ultra-fine type labels, with traditional signals from entity linking. Although the problem is more challenging at finer granularity, we find that mixing fine and coarse-grained supervision helps significantly, and that our proposed model with a multitask objective exceeds the performance of existing entity typing models. Lastly, we show that head-word supervision can be used for previous formulations of entity typing, setting the new state-of-the-art performance on an existing fine-grained NER benchmark.

3.2 Task

Given a sentence and an entity mention e within it, the task is to predict a set of natural-language phrases T that describe the type of e. The selection of T is context sensitive; for example, in “Bill Gates has donated billions to eradicate malaria,” Bill Gates should be typed as “philanthropist” and not “inventor.” This distinction is important for context-sensitive tasks such as coreference resolution and question answering (e.g. “Which philanthropist is trying to prevent malaria?”). We annotate a dataset of about 6,000 mentions via crowdsourcing (Section 3.2.1), and demonstrate that using an large type vocabulary substantially increases annotation coverage and diversity over existing approaches (Section 3.2.2).

3.2.1 Crowdsourcing Entity Types

To capture multiple domains, we sample sentences from Gigaword [Parker et al., 2011], OntoNotes [Hovy et al., 2006], and web articles [Singh et al., 2012]. We select entity mentions by taking maximal noun phrases from a constituency parser [Manning et al., 2014] and mentions from a coreference resolution system [Lee et al., 2017].

We provide the sentence and the target entity mention to five crowd workers on Mechanical Turk, and ask them to annotate the entity’s type. To encourage annotators to generate fine-grained types, we require at least one general type (e.g. person, organization, location) and two specific types (e.g. doctor, fish, religious institute).We provided workers with an auto-complete drop-down menu consisting of our type vocabulary

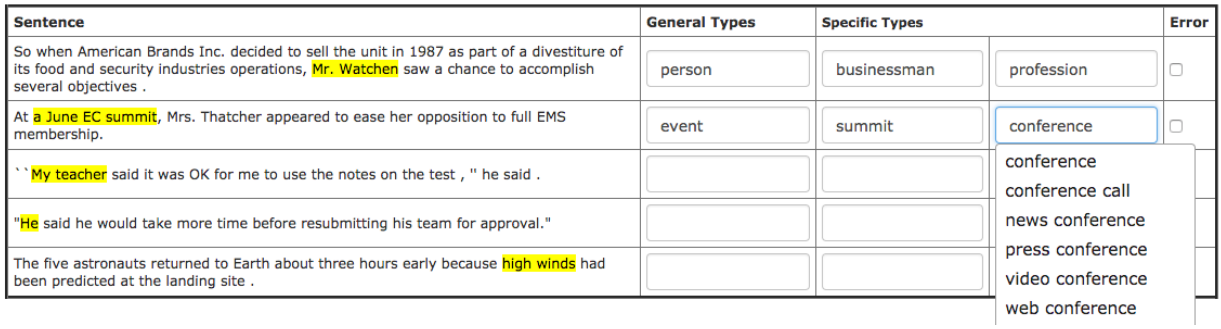


Figure 3.2: Data collection framework screenshot. The crowdworkers are provided with auto-complete vocabulary which lists all nouns in the Wikitionary.

(see Figure 3.2). The type vocabulary was constructed by taking all the singular nouns in Wiktionary (including multiword expressions, such as "prime minister"), then pruning words that appeared less than 5 times in our training data or were not in 40,000 frequent terms in the GloVe vocabulary. After this process, the final list contained 10,331 nouns. Five workers annotated each example, producing an initial set of types T_0. We expanded T_0 to T_1 by adding synonyms and hypernyms from WordNet [Miller, 1995], as well as randomly selected negative types. We then asked five different annotators which types in T_1 fit the context c. To ensure good annotation quality, we selected the most agreed-upon type in T_0 as a true-positive and one of the random negative types as a true-negative, and prevented annotators who misclassified them from completing the task.

Each pair of annotators agreed on 85% of the binary decisions (i.e. whether a type is suitable or not), and 0.47 in Fleiss's κ. To further improve consistency, the final type set T contained only types selected by at least 3/5 annotators. We removed examples without any types after the pruning. Our collection process focuses on precision. Thus, the final set is diverse but not comprehensive, making evaluation non-trivial (see Section 3.6.4).

3.2.2 Data Analysis

We collected about 6,000 examples. For analysis, we classified each type into three disjoint bins:

- 9 **general** types: person, location, object, organization, place, entity, object, time, event
- 121 **fine-grained** types, mapped to fine-grained entity labels from prior work [Ling and Weld, 2012; Gillick et al., 2014] (e.g. film, athlete)

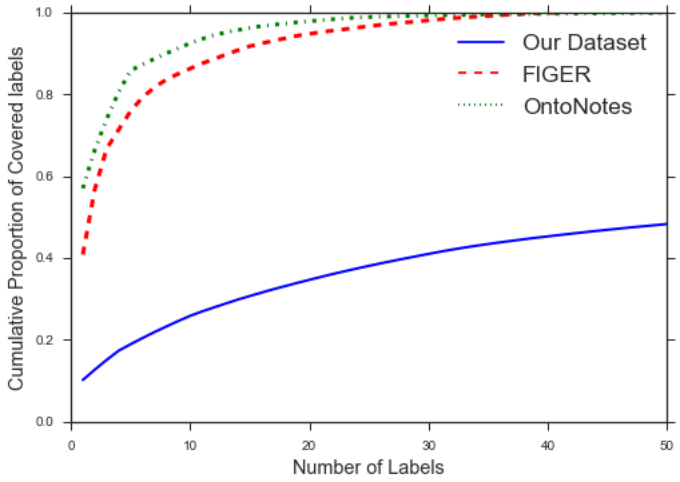


Figure 3.3: The label distribution across different evaluation datasets. In existing datasets, the top 4 or 7 labels cover over 80% of the labels. In ours, the top 50 labels cover less than 50% of the data.

- 10,201 **ultra-fine** types, encompassing every other label in the type space (e.g. detective, lawsuit, temple, weapon, composer)

On average, each example has 5 labels: 0.9 general, 0.6 fine-grained, and 3.9 ultra-fine types. Among the 10,000 ultra-fine types, 2,300 unique types were actually found in the 6,000 crowdsourced examples. Nevertheless, our distant supervision data (Section 3.4) provides positive training examples for every type in the entire vocabulary, and our model (Section 3.5) can and does predict from a 10K type vocabulary. For example, the model correctly predicts “television network” and “archipelago" for some mentions, even though that type never appears in the 6,000 crowdsourced examples.

Improving Type Coverage We observe that prior fine-grained entity typing datasets are heavily focused on coarse-grained types. To quantify our observation, we calculate the distribution of types in FIGER [Ling and Weld, 2012], OntoNotes [Gillick et al., 2014], and our data. For examples with multiple types ($|T| > 1$), we counted each type $1/|T|$ times.

Figure 3.3 shows the percentage of labels covered by the top N labels in each dataset. In previous enitity typing datasets, the distribution of labels is highly skewed towards the top few labels. To cover 80% of the examples, FIGER requires only the top 7 types, while OntoNotes needs only 4; our dataset requires 429 different types.

Source	Example Sentence	Labels	Size	Prec.
Head Words	**Western powers that brokered the proposed deal in Vienna** are likely to balk, said Valerie Lincy, a researcher with the Wisconsin Project.	power	20M	80.4%
	Alexis Kaniaris, CEO of the organizing company Europartners, explained, speaking in a radio program in **national radio station NET**.	radio, station, radio_station		
Entity Linking + Definitions	**Toyota** recalled more than 8 million vehicles globally over sticky pedals that can become entrapped in floor mats.	manufacturer	2.7M	77.7%
Entity Linking + KB	Iced Earth's musical style is influenced by many traditional heavy metal groups such as **Black Sabbath**.	person, artist, actor, author, musician	2.5M	77.6%

Table 3.2: Distant supervision examples and statistics. We extracted the headword and Wikipedia definition supervision from Gigaword and Wikilink corpora. KB-based supervision is mapped from prior work, which used Wikipedia and news corpora.

Figure 3.1 takes a deeper look by visualizing the types that cover 90% of the data, demonstrating the diversity of our dataset. It is also striking that more than half of the examples in OntoNotes are classified as "other," perhaps because of the limitation of its predefined ontology.

Improving Mention Coverage Existing datasets focus mostly on named entity mentions, with the exception of OntoNotes, which contained nominal expressions. This has implications on the transferability of FIGER/OntoNotes-based models to tasks such as coreference resolution, which need to analyze all types of entity mentions (pronouns, nominal expressions, and named entity mentions). Our new dataset provides a well-rounded benchmark with roughly 40% pronouns, 38% nominal expressions, and 22% named entity mentions. The case of pronouns is particularly interesting, since the mention itself provides little information.

3.3 Related Work

Fine-grained named entity recognition has received growing attention, and is used in many applications [Gupta et al., 2017; Ren et al., 2017; Yaghoobzadeh et al., 2017; Raiman and Raiman, 2018]. Researchers studied typing in varied contexts, including mentions in specific sentences [Ling and Weld, 2012; Gillick et al., 2014; Yogatama et al., 2015; Dong et al., 2015; Schutze et al., 2017], corpus-level prediction [Yaghoobzadeh

and Schütze, 2016], and lexicon level (given only a noun phrase with no context) [Yao et al., 2013]. Recent work introduced fine-grained type ontologies [Rabinovich and Klein, 2017; Corro et al., 2015; Murty et al., 2018], defined using Wikipedia categories (100), Freebase types (1K) and WordNet senses (16K). However, they focus on named entities, and data has been challenging to gather, often approximating gold annotations with distant supervision. In contrast, (1) our ontology contains any frequent noun phrases that depicts a type, (2) our task goes beyond named entities, covering every noun phrase (even pronouns), and (3) we provide crowdsourced annotations which provide context-sensitive, fine grained type labels.

Contextualized fine-grained entity typing is related to selectional preference [Resnik, 1996; Pantel et al., 2007; Zapirain et al., 2013; de Cruys, 2014], where the goal is to induce semantic generalizations on the type of arguments a predicate prefers. Rather than focusing on predicates, we condition on the entire sentence to deduce the arguments' types, which allows us to capture more nuanced types. For example, not every type that fits "**He** played the violin in his room" is also suitable for "**He** played the violin in the Carnegie Hall". Entity typing here can be connected to argument finding in semantic role labeling.

To deal with noisy distant supervision for KB population and entity typing, researchers used multi-instance multi-label learning [Surdeanu et al., 2012; Yaghoobzadeh et al., 2017] or custom losses [Abhishek et al., 2017; Ren et al., 2016a]. Our multitask objective handles noisy supervision by pooling different distant supervision sources across different levels of granularity.

3.4 Distant Supervision

Training data for fine-grained NER systems is typically obtained by linking entity mentions and drawing their types from knowledge bases (KBs). This approach has two limitations: recall can suffer due to KB incompleteness [West et al., 2014a], and precision can suffer when the selected types do not fit the context [Ritter et al., 2011]. We alleviate the recall problem by mining entity mentions that were linked to Wikipedia in HTML, and extract relevant types from their encyclopedic definitions (Section 3.4.1). To address the precision issue (context-insensitive labeling), we propose a new source of distant supervision: automatically extracted nominal head words from raw text (Section 3.4.2). Using head words as a form of distant supervision provides fine-grained information about named entities and nominal mentions. While a KB may link "the 44th president of the United States" to many types such as author, lawyer, and professor,

head words provide only the type "president", which is relevant in the context.

We experiment with the new distant supervision sources as well as the traditional KB supervision. Table 3.2 shows examples and statistics for each source of supervision. We annotate 100 examples from each source to estimate the noise and usefulness in each signal (precision in Table 2).

3.4.1 Entity Linking

For KB supervision, we leveraged training data from prior work [Ling and Weld, 2012; Gillick et al., 2014] by manually mapping their ontology to our 10,000 noun type vocabulary, which covers 130 of our labels (general and fine-grained).[1] Section 6 defines this mapping in more detail.

To improve both entity and type coverage of KB supervision, we use definitions from Wikipedia. We follow Shnarch et al. Shnarch et al. [2009] who observed that the first sentence of a Wikipedia article often states the entity's type via an "is a" relation; for example, "Roger Federer is a Swiss professional tennis player." Since we are using a large type vocabulary, we can now mine this typing information.[2] We extracted descriptions for 3.1M entities which contain 4,600 unique type labels such as "competition," "movement," and "village."

We bypass the challenge of automatically linking entities to Wikipedia by exploiting existing hyperlinks in web pages [Singh et al., 2012], following prior work [Ling and Weld, 2012; Yosef et al., 2012]. Since our heuristic extraction of types from the definition sentence is somewhat noisy, we use a more conservative entity linking policy[3] that yields a signal with similar overall accuracy to KB-linked data.

3.4.2 Contextualized Supervision

Many nominal entity mentions include detailed type information within the mention itself. For example, when describing Titan V as "the newly-released graphics card", the head words and phrases of this mention ("graphics card" and "card") provide a somewhat noisy, but very easy to gather, context-sensitive type signal.

We extract nominal head words with a dependency parser [Manning et al., 2014] from the Gigaword corpus as well as the Wikilink dataset. To support multiword expressions, we included nouns that appear

[1] Data from: `https://github.com/shimaokasonse/NFGEC`

[2] We extract types by applying a dependency parser Manning et al. [2014] to the definition sentence, and taking nouns that are dependents of a copular edge or connected to nouns linked to copulars via appositive or conjunctive edges.

[3] Only link if the mention contains the Wikipedia entity's name *and* the entity's name contains the mention's head.

next to the head if they form a phrase in our type vocabulary. Finally, we lowercase all words and convert plural to singular.

Our analysis reveals that this signal has a comparable accuracy to the types extracted from entity linking (around 80%). Many errors are from the parser, and some errors stem from idioms and transparent heads (e.g. "parts of capital" labeled as "part"). While the headword is given as an input to the model, with heavy regularization and multitasking with other supervision sources, this supervision helps encode the context.

3.5 Model

We design a model for predicting sets of types given a mention in context. The architecture resembles the neural AttentiveNER model [Shimaoka et al., 2017], while improving the sentence and mention representations, and introducing a new multitask objective to handle multiple sources of supervision.

3.5.1 Context Representation

Given a sentence $x_1, \ldots, x_n$, we represent each token x_i using a pre-trained word embedding w_i. We concatenate an additional location embedding l_i which indicates whether x_i is before, inside, or after the mention. We then use $[x_i; l_i]$ as an input to a bidirectional LSTM, producing a contextualized representation h_i for each token; this is different from the architecture of Shimaoka et al. 2017, who used two separate bidirectional LSTMs on each side of the mention. Finally, we represent the context c as a weighted sum of the contextualized token representations using MLP-based attention:

$$a_i = \text{SoftMax}_i(v_a \cdot \text{relu}(W_a h_i))$$

Where W_a and v_a are the parameters of the attention mechanism's MLP, which allows interaction between the forward and backward directions of the LSTM before computing the weight factors.

3.5.2 Mention Representation

We represent the mention m as the concatenation of two items: (a) a character-based representation produced by a CNN on the entire mention span, and (b) a weighted sum of the pre-trained word embeddings

in the mention span computed by attention, similar to the mention representation in a recent coreference resolution model [Lee et al., 2017]. The final representation is the concatenation of the context and mention representations: $r = [c; m]$.

3.5.3 Label Prediction

We learn a type label embedding matrix $W_t \in \mathbb{R}^{n \times d}$ where n is the number of labels in the prediction space and d is the dimension of r. This matrix can be seen as a combination of three sub matrices, $W_{general}, W_{fine}, W_{ultra}$, each of which contains the representations of the general, fine, and ultra-fine types respectively. We predict each type's probability via the sigmoid of its inner product with r: $y = \sigma(W_t r)$. We predict every type t for which $y_t > 0.5$, or $\arg\max y_t$ if there is no such type.

3.5.4 Multitask Objective

The distant supervision sources provide partial supervision for ultra-fine types; KBs often provide more general types, while head words usually provide only ultra-fine types, without their generalizations. In other words, the absence of a type at a different level of abstraction does not imply a negative signal; e.g. when the head word is "inventor", the model should not be discouraged to predict "person".

Prior work used a customized hinge loss [Abhishek et al., 2017] or max margin loss [Ren et al., 2016a] to improve robustness to noisy or incomplete supervision. We propose a multitask objective that reflects the characteristic of our training dataset. Instead of updating all labels for each example, we divide labels into three bins (general, fine, and ultra-fine), and update labels only in bin containing at least one positive label. Specifically, the training objective is to minimize J where t is the target vector at each granularity:

$$J_{\text{all}} = J_{\text{general}} \cdot \mathbb{1}_{\text{general}}(t) + J_{\text{fine}} \cdot \mathbb{1}_{\text{fine}}(t) + J_{\text{ultra}} \cdot \mathbb{1}_{\text{ultra}}(t)$$

Where $\mathbb{1}_{\text{category}}(t)$ is an indicator function that checks if t contains a type in the category, and J_{category} is the category-specific logistic regression objective:

$$J = -\sum_i t_i \cdot \log(y_i) + (1 - t_i) \cdot \log(1 - y_i)$$

Hyperparameters We use 300 dimensional pre-trained GloVe word vectors, [4] 50 dimensions for the location vector, and set the LSTMs' dimensions to 100. For the attention mechanism, we used 100 for the hidden dimension. For the mention span representation, the character embedding dimension was 100, and the filter number for character CNN was 50. We used pytorch for implementations. We used dropout for regularization, with a probability of 0.2 for the input sequence pre-trained embeddings, and 0.5 for mention representations. The sentences are cut off after 50 tokens, mention spans are cut off after 25 characters and we ignored mentions with longer than 10 words during training. The model parameters are optimized with Adam, with an initial learning rate of 0.001, over batches of 1000 examples.

3.6 Evaluation

3.6.1 Experiment Setup

The crowdsourced dataset (Section 3.2.1) was randomly split into train, development, and test sets, each with about 2,000 examples. We use this relatively small manually-annotated training set (*Crowd* in Table 3.4) alongside the two distant supervision sources: entity linking (KB and Wikipedia definitions) and head words. To combine supervision sources of different magnitudes (2K crowdsourced data, 4.7M entity linking data, and 20M head words), we sample a batch of equal size from each source at each iteration. We report macro-averaged precision, recall, and F1, and the average mean reciprocal rank (MRR).

3.6.2 Comparison Systems

We reimplement the recent AttentiveNER model [Shimaoka et al., 2017] as a baseline. We use the AttentiveNER model with no engineered features or hierarchical label encoding (as a hierarchy is not clear in our label setting) and let it predict from the same label space, training with the same supervision data.

After our paper is released, two follow-up work reported their results on the dataset. We summarize their models briefly here. Onoe and Durrett [2019] presented a method which denoises the distant supervision dataset. They uses a two stage process to improve distant supervision dataset. Using 2K crowdsourced training portion of the dataset, they construct two classifiers, one to filter noise examples and another to relabel

[4] http://nlp.stanford.edu/data/glove.840B.300d.zip

Model	Dev				Test			
	MRR	P	R	F1	MRR	P	R	F1
AttentiveNER	0.221	**53.7**	15.0	23.5	0.223	**54.2**	15.2	23.7
Our Model	**0.229**	48.1	**23.2**	**31.3**	**0.234**	47.1	**24.2**	**32.0**
Xiong et al. [2019]	0.250	50.5	28.7	36.6	0.253	50.3	29.2	36.9
Onoe and Durrett [2019] w/ GloVe	-	46.4	23.3	31.0	-	47.6	23.3	31.3
w/ ELMo	-	55.6	28.1	37.3	-	55.8	27.7	37.0
w/ ELMo & augmentation	-	50.7	33.1	40.1	-	51.5	33.0	40.2
BERT (reported in Onoe and Durrett [2019])	-	51.6	32.8	40.1	-	51.6	33.0	40.2

Table 3.3: Performance on the new entity typing benchmark. We show results for both development and test sets. The top section are the results presented in the original paper, and the bottom section describes recent models.

Train Data	Total				General (1918)			Fine (1289)			Ultra-Fine (7594)		
	MRR	P	R	F1	P	R	F1	P	R	F1	P	R	F1
All	**0.229**	48.1	**23.2**	**31.3**	60.3	61.6	61.0	40.4	**38.4**	**39.4**	42.8	8.8	14.6
– Crowd	0.173	40.1	14.8	21.6	53.7	45.6	49.3	20.8	18.5	19.6	54.4	4.6	8.4
– Head	0.220	**50.3**	19.6	28.2	58.8	**62.8**	60.7	**44.4**	29.8	35.6	**46.2**	4.7	8.5
– EL	0.225	48.4	22.3	30.6	**62.2**	60.1	**61.2**	40.3	26.1	31.7	41.4	**9.9**	**16.0**

Table 3.4: Results on the development set for different type granularity and for different supervision data with our model. In each row, we remove a single source of supervision. Entity linking (EL) includes supervision from both KB and Wikipedia definitions. The numbers in the first row are example counts for each type granularity.

surviving examples with more complete type label set. In addition to improving the training data, they used contextualized word embeddings [Peters et al., 2018] for better context and mention representation. Xiong et al. [2019] focused on modeling relations between different labels, capturing hierarchy and correlation among the labels. Specifically, the method constructs a type label graph and learns graph convolutional network (GCN) [Kipf and Welling, 2016] on it.

Lastly, we report the performance of large-scale language model [dev], as reported in Onoe and Durrett [2019]. This baseline encodes each example as a "[CLS] sentence [SEP] mention [SEP]" sequence, and take the output vector at the position of the [CLS] token as the feature vector, and fine-tune the language model on the 2K crowdsourced examples.

3.6.3 Results

Table 3.3 reports the performance on proposed ultra fine entity typing dataset. The top section of the table shows the performance of our model and our reimplementation of AttentiveNER. Our model, which uses a multitask objective to learn finer types without punishing more general types, shows recall gains at the cost of drop in precision. The MRR score shows that our model is slightly better than the baseline at ranking correct types above incorrect ones than the baseline.

The bottom section of the table 3.3 presents more recent results. Both Xiong et al. [2019] and Onoe and Durrett [2019] share the multitask loss proposed in Section 3.5.4 and general BiLSTM architecture for encoding the sentence. Xiong et al. [2019] improved typing model by encoding label correlations with GCN. While our label set does not impose a strict hierarchy, there is strong correlations among type labels and hierarchical patterns (e.g., city must be a location and editors tends to be writers). Their method learns such relations and improved the performance. Onoe and Durrett [2019] presents an orthogonal contribution, focusing on improving distant supervision and using better word representation. As shown in other benchmark tasks such as coreference resolution and question answering [Peters et al., 2018], using contextualized word embedding instead of fixed word embedding to represent text brings a significant performance gain (from 31.0 F1 to 37.3 F1). Improving the distant supervision (second to last row) showed an additional gain, setting a new state-of-the-art performance of 40.1 F1.

Fine-tuning a large scale language model [Devlin et al., 2019] alone also showed a very strong performance, matching the best performance of Onoe and Durrett [2019]. The reported performance is from a smaller BERT model (12 layers, 110M parameters), and a preliminary study showed even stronger results can be achieved with a large language model (24 layers, 340M parameters).[5] The learning objective of Devlin et al. [2019] is very similar to our head word supervision objective, and their model is trained with more data with a more expressive architecture. Specifically, they used 12 layers of Transformers [Vaswani et al., 2017] to represent the context while our model used a single layer LSTM. Given the strong performance of BERT and the similarity of the objectives, we can assume large scale language model might contain a superset of information from our distant supervision. Better utilizing unsupervised learning (language model) on

[5]Recent work Wang et al. [2019] explored a bigger language model [Devlin et al., 2019] for this task. They reframed the task as a binary decision task of deciding whether a single type is adequate for a context and mention. Under this setting, larger language model had 82.3 accuracy, while smaller model had the accuracy of 68.7. As a reference, humans had 90.8 accuracy under this setting.

(a)	Example	Bruguera said {**he**} had problems with his left leg and had grown tired early during the match .
	Annotation	**person, athlete, player, adult, male, contestant**
	Prediction	**person, athlete, player, adult, male, contestant**, defendant, man
(b)	Example	{**The explosions**} occurred on the night of October 7 , against the Hilton Taba and campsites used by Israelis in Ras al-Shitan.
	Annotation	**event**, *calamity, attack, disaster*
	Prediction	**event,** accident
(c)	Example	Similarly , Enterprise was considered for refit to replace Challenger after {**the latter**} was destroyed , but Endeavour was built from structural spares instead .
	Annotation	*object, spacecraft, rocket, thing, vehicle, shuttle*
	Prediction	event
(d)	Context	" There is a wealth of good news in this report , and I 'm particularly encouraged by the progress {**we**} are making against AIDS , " HHS Secretary Donna Shalala said in a statement.
	Annotation	**government, group,** *organization,hospital,administration,socialist*
	Prediction	**government, group**, person

Table 3.5: Example and predictions from our best model on the development set. Entity mentions are marked with curly brackets, the correct predictions are boldfaced, and the missing labels are italicized and written in red.

top of semi-supervised learning (headword and entity linking supervision) will be an interesting direction for future work.

Table 3.4 shows the performance breakdown for different type granularity and different supervision of our initial results. Overall, as seen in previous work on fine-grained NER literature [Gillick et al., 2014; Ren et al., 2016a], finer labels were more challenging to predict than coarse grained labels, and this issue is exacerbated when dealing with ultra-fine types. All sources of supervision appear to be useful, with crowd-sourced examples making the biggest impact. Head word supervision is particularly helpful for predicting ultra-fine labels, while entity linking improves fine label prediction. The low general type performance is partially because of nominal/pronoun mentions (e.g. "it"), and because of the large type inventory (sometimes "location" and "place" are annotated interchangeably).

3.6.4 Analysis

We manually analyzed 50 examples from the development set, four of which we present in Table 3.5. Overall, the model was able to generate accurate general types and a diverse set of type labels. Despite our efforts to annotate a comprehensive type set, the gold labels still miss many potentially correct labels (example (a): "man" is reasonable but counted as incorrect). This makes the precision estimates lower than the actual performance level, with about half the precision errors belonging to this category. Real precision errors include predicting co-hyponyms (example (b): "accident" instead of "attack"), and types that may be true, but are not supported by the context.

We found that the model often abstained from predicting any fine-grained types. Especially in challenging cases as in example (c), the model predicts only general types, explaining the low recall numbers (28% of examples belong to this category). Even when the model generated correct fine-grained types as in example (d), the recall was often fairly low since it did not generate a complete set of related fine-grained labels.

Estimating the performance of a model in an incomplete label setting and expanding label coverage are interesting areas for future work. Our task also poses a potential modeling challenge; sometimes, the model predicts two incongruous types (e.g. "location" and "person"), which points towards modeling the task as a joint set prediction task, rather than predicting labels individually. We provide sample outputs on the project website.

3.7 Improving Existing Fine-Grained NER with Better Distant Supervision

We show that our model and distant supervision can improve performance on an existing fine-grained NER task. We chose the widely-used OntoNotes [Gillick et al., 2014] dataset which includes nominal and named entity mentions.[6]

[6]While we were inspired by FIGER [Ling and Weld, 2012], the dataset presents technical difficulties. The test set has only 600 examples, and the development set was labeled with distant supervision, not manual annotation. We therefore focus our evaluation on OntoNotes.

	Acc.	Ma-F1	Mi-F1
AttentiveNER++	51.7	70.9	64.9
AFET [Ren et al., 2016a]	55.1	71.1	64.7
LNR [Ren et al., 2016b]	57.2	71.5	66.1
Ours (ONTO+WIKI+HEAD)	**59.5**	**76.8**	**71.8**
Xiong et al. [2019]	59.2	77.8	72.2
Onoe and Durrett [2019]	64.9	84.5	79.2

Table 3.6: Results on the OntoNotes fine-grained entity typing test set. The first two models (AttentiveNER++ and AFET) use only KB-based supervision. LNR uses a filtered version of the KB-based training set. Our model uses all our distant supervision sources.

3.7.1 Augmenting the Training Data

The original OntoNotes training set (ONTO in Tables 3.6 and 3.7) is extracted by linking entities to a KB. We supplement this dataset with our two new sources of distant supervision: Wikipedia definition sentences (WIKI) and head word supervision (HEAD) (see Section 3.4). To convert the label space, we manually map a single noun from our natural-language vocabulary to each formal-language type in the OntoNotes ontology. 77% of OntoNote's types directly correspond to suitable noun labels (e.g. "doctor" to "/person/doctor"), whereas the other cases were mapped with minimal manual effort (e.g. "musician" to "person/artist/music", "politician" to "/person/political_figure"). We then expand these labels according to the ontology to include their hypernyms ("/person/political_figure" will also generate "/person"). Lastly, we create negative examples by assigning the "/other" label to examples that are not mapped to the ontology. The augmented dataset contains 2.5M/0.6M new positive/negative examples, of which 0.9M/0.1M are from Wikipedia definition sentences and 1.6M/0.5M from head words.

3.7.2 Experiment Setup

We compare performance to other published results and to our reimplementation of AttentiveNER [Shimaoka et al., 2017]. We also compare models trained with different sources of supervision. For this dataset, we did not use our multitask objective (Section 3.5), since expanding types to include their ontological hypernyms largely eliminates the partial supervision assumption. Following prior work, we report macro- and micro-averaged F1 score, as well as accuracy (exact set match).

Model	Training Data			Performance		
	ONTO	WIKI	HEAD	Acc.	MaF1	MiF1
Attn.	✓			46.5	63.3	58.3
NER	✓	✓	✓	53.7	72.8	68.0
Ours	✓			41.7	64.2	59.5
	✓	✓		48.5	67.6	63.6
	✓		✓	57.9	73.0	66.9
		✓	✓	60.1	75.0	68.7
	✓	✓	✓	**61.6**	**77.3**	**71.8**

Table 3.7: Ablation study on the OntoNotes fine-grained entity typing development. The second row isolates dataset improvements, while the third row isolates the model.

3.7.3 Results

Table 3.6 shows the overall performance on the test set. Our combination of model and training data shows a clear improvement from prior work, setting a new state-of-the art result at the time of publication.[7] Recent models further improved the performance on this benchmark, by modeling label correlation [Xiong et al., 2019] and providing better word representation and improving distant supervision [Onoe and Durrett, 2019].

In Table 3.7, we show an ablation study. Our new supervision sources improve the performance of both the AttentiveNER model and our own. We observe that every supervision source improves performance in its own right. Particularly, the naturally-occurring head-word supervision seems to be the prime source of improvement, increasing performance by about 10% across all metrics.

3.7.4 Predicting Miscellaneous Types

While analyzing the data, we observed that over half of the mentions in OntoNotes' development set were annotated only with the miscellaneous type ("/other"). For both models in our evaluation, detecting the miscellaneous category is substantially easier than producing real types (94% F1 vs. 58% F1 with our best model). For further analysis, we divided mentions into two categories: mentions only annotated with '/other' and all other mentions (typed). We show macro-averaged precision, recall, and F1 for typed mentions, and accuracy for '/other' mentions. In Table 3.8, we show the ablation study of different sets of supervision

[7] We did not compare to a system from [Yogatama et al., 2015], which reports slightly higher test number (72.98 micro F1) as they used a different, unreleased test set.

	Train Data			Total (2202)			Typed (1069)			Other (1133)
	ONTO	WIKI	HEAD	Acc.	Ma-F1	Mi-F1	P	R	Ma-F1	Accuracy
Attentive	✓			46.5	63.3	58.3	60.1	39.8	47.9	73.0
NER	✓	✓	✓	53.7	72.8	68.0	**70.2**	48.2	57.2	82.6
	✓			41.7	64.2	59.5	61.8	48.5	54.4	59.3
	✓	✓		48.5	67.6	63.6	67.1	**51.8**	58.4	70.0
Ours	✓		✓	57.9	73.0	66.9	57.9	42.3	48.9	92.6
		✓	✓	60.1	75.0	68.7	59.6	45.0	51.3	**95.7**
	✓	✓	✓	**61.6**	**77.3**	**71.8**	67.4	**51.8**	**58.6**	92.6

Table 3.8: Performance breakdown on the OntoNotes development set. Both new distant supervision improves the performance, both on our model and the prior model.

and the performance breakdown between miscellaneous and typed mentions. Our data augmentation with negative examples significantly improved detecting mentions beyond the existing ontology (/other), and also improved the overall performance.

3.8 Summary

This chapter presents a new approach to infer a rich set of entity types from the sentence it occurs in. Using virtually unrestricted types allows us to expand the standard KB-based training methodology with typing information from Wikipedia definitions and naturally-occurring head-word supervision. These new forms of distant supervision boost performance on our new dataset as well as on an existing fine-grained entity typing benchmark. Our work established the first performance levels for new evaluation dataset, and the data support future work in entity typing.

Chapter 4

Entity-Entity Sentiment Extraction

In the previous chapters, we studied individual entity attributes, that can be extracted from noun phrases describing categories entities belong to and a sentence in which an entity occurs. Now we study the *relationship* between two entities. Prior work extensively covered explicitly stated, factual relationship between entities (e.g., entity X is born in entity Y). Instead, we focus on a high-level, implied relationship between a pair entities that can be inferred from context. Specifically, we investigate directed sentiment relationships between a pair of entities co-occurring in a document. While news documents rarely mentioned explicitly how one entity feels towards another entity, document context often gives evidences for such inferences. Our model predict directed opinions (*who* feels positively or negatively towards *whom*) for all entities mentioned in a document.

Understanding document level implicit semantics is challenging. To encourage more complete and consistent predictions, we introduce an integer linear programming model that jointly consider (1) sentence- and discourse-level sentiment cues, (2) factual evidence about entity factions, and (3) global constraints based on social science theories such as homophily, social balance, and reciprocity. Together, these cues allow for rich inference across groups of entities, including for example that CEOs and the companies they lead are likely to have similar sentiment towards others. We evaluate performance on new, densely labeled data that provides supervision for all pairs, complementing previous work that only labeled pairs mentioned in the same sentence. Experiments demonstrate that the global model outperforms sentence-level baselines, by providing more coherent predictions across sets of related entities. This chapter is based on the work

Russia criticized *Belarus* for permitting *Georgian* President *Mikheil Saakhashvili* to appear on *Belorussian* television. "The appearance was an unfriendly step towards *Russia*," the speaker of *Russian* parliament *Boris Gryzlov* said. . . . *Saakhashvili* announced Thursday that *he* did not understand *Russia*'s claims. *Moscow* refused to have any business with *Georgia*'s president after the armed conflict in 2008 . . .

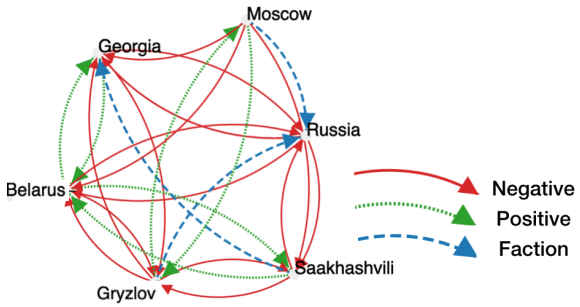


Figure 4.1: Example text excerpt paired with the document-level sentiment graph we aim to recover. The graph includes edges with direct textual support (e.g., from Russian to Belarus given the verb "criticized") as well as ones that must be inferred at the whole-document level (e.g., from Gryzlov to Saakhashvili given the web of relationships and opinions between them, Georgia, Russian, and Belarus).

originally described in Choi et al. [2016].

4.1 Introduction

Documents often present a complex web of facts and opinions that hold among the entities they describe. Consider the international relations story in Figure 4.1. Representatives from three countries form factions and create a network of sentiment. While some opinions are relatively directly stated (e.g., Russia criticizes Belarus), many others must be inferred based on the factual ties among entities (e.g., Moscow, Gryzlov, and Russia probably share the same sentiment towards other entities) and known social context (e.g., Russia probably dislikes Saakhashvili since Russia criticized Belarus for supporting him). In this chapter, we show that jointly reasoning about all of these factors can provide more complete and consistent document-level sentiment predictions.

More concretely, we present a global model for document-level entity-to-entity sentiment, i.e., *who* feels positively (or negatively) towards *whom*. Our goal is to make exhaustive predictions over all entity pairs, including those that require cross-sentence inference. We present a Integer Linear Programming (ILP) model that combines three complementary types of evidence: entity-pair sentiment classification, template-based faction extraction, and sentiment dynamics in social groups. Together, they allow for recovering more complete predictions of both the explicitly stated and implicit sentiment, while preserving consistency.

The sentiment dynamics in social groups, motivated by social science theories, are encoded as soft ILP

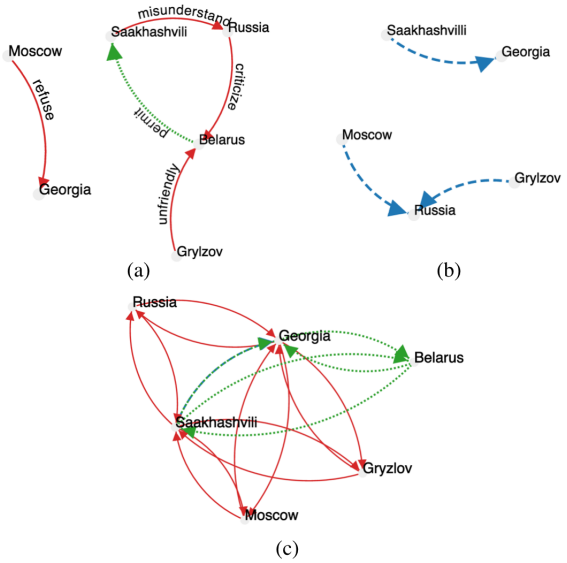


Figure 4.2: Entity subgraphs for the example in Figure 1: (a) shows explicitly stated sentiment, (b) shows faction relationships and (c) shows all edges for Georgia and its representative Saakhashvili. Through Saakhasvili's relationship with Belarus, Georgia forms an alliance with Belarus, providing evidence for an inferred negative stance towards Russia. Green dotted edges represent positive sentiment, red are negative, and blue dashed lines show faction relationship.

constraints. They include a notion of homophily, that entities in the same group tend to have similar opinions [Lazarsfeld and Merton, 1954]. For example, Figure 2b shows directed faction edges, where one entity is likely to agree with the other's opinions. They also encode dyadic social constraints (i.e., the likely reciprocity of opinions [Gouldner, 1960]) and triadic social dynamics following social balance theory [Heider, 1946]. For example, from Russia's criticism on Belarus and Belarus' positive attitude towards Saakhashvilli (in Figure 2a), we can infer that Russia is negative towards Saakhashvilli (in Figure 2c). When considered in aggregate, these constraints can greatly improve the consistency over the overall document-level predictions.

Our work stands in contrast to previous approaches in three aspects. First, we apply social dynamics motivated by social science theories to entity-entity sentiment analysis in unstructured text. In contrast, most previous studies focused on social media or dialogue data with overt social network structure when integrating social dynamics [Tan et al., 2011; Hu et al., 2013; West et al., 2014b]. Second, we aim to recover sentiment that can be inferred through partial evidence that spans multiple sentences. This complements prior efforts for accessing implied sentiment where the key evidence is, by and large, at the sentence level [Zhang and Liu, 2011; Yang and Cardie, 2013; Deng and Wiebe, 2015a]. Finally, we present the first

approach to model the relationship between factual and subjective relations.

We evaluate the approach on a newly gathered corpus with dense document-level sentiment labels in news articles. This data includes comprehensively annotated sentiment between all entity pairs, including those that do not appear together in any single sentence. Experiments demonstrate that the global model significantly improves performance over a pairwise classifier and other strong baselines. We also perform a detailed ablation and error analysis, showing cases where the global constraints contribute and pointing towards important areas for future work.

4.2 Global Model

Given a news document d, and named entities e_1,...,e_n in d, where each entity e_i has mentions $m_{i1} \cdots m_{ik}$, the task is to decide directed sentiment between all pairs of entities. We predict the directed sentiment from e_i to e_j at the document level, i.e., sent($e_i \rightarrow e_j$) $\in$ {positive, unbiased, negative}, for all e_i, $e_j \in d$ where $i \neq j$, assuming that sentiment is consistent within the document.

We introduce a document-level ILP that includes base models and soft social constraints. ILP has been used successfully for a wide range of NLP tasks [Roth and Yih, 2004], perhaps because they easily support incorporating different types of global constraints. We use two base models: (1) a learned pairwise sentiment classifier (Sec 4.3.1) that combines sentence- and discourse-level features to make predictions for each entity pair and (2) a pattern-based faction extractor (Sec 4.3.2) that detects alliances among a subset of the entities.

The ILP is solved by maximizing:

$$F = \psi_{social} + \psi_{fact} + \sum_{i=1}^{n} \sum_{j=1}^{n} \psi_{ij}$$

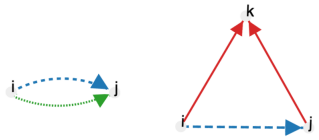
where F combines soft constraints (ψ_{social}, ψ_{fact} defined in detail in this section) with pairwise potentials ψ_{ij} defined as:

$$\psi_{ij} = \phi_{pos_{ij}} \cdot \texttt{pos}_{ij} + \phi_{neg_{ij}} \cdot \texttt{neg}_{ij} + \phi_{neu_{ij}} \cdot \texttt{neu}_{ij}$$

Each potential ψ_{ij} includes the sentiment classifier scores (ϕ_{pos}, ϕ_{neg}, ϕ_{neu}) with binary variables $\texttt{pos}_{ij}$, $\texttt{neu}_{ij}$ and $\texttt{neg}_{ij}$ where, for example, $\texttt{neg}_{ij}$=1 indicates that e_i is negative towards e_j. Decision variables

Sentence	i	j
Canadian Prime Minister Harper. . .	Canada	Harper
. . . Reid, the Democratic leader. . .	Reid	Democratic
Goldman spokesman DuVally	Goldman	DuVally
. . . Djibouti, a key U.S. ally.	Djibouti	U.S.

(a) Detection examples



(b) Visual representation of common inference patterns.

Figure 4.3: An example sentiment inference from faction relationships. Pairs in factions are encouraged to share opinions, and to be positive towards other tied entities. On the right, sentiment edges can be both positive or both negative.

$\texttt{pos}_{ij}$ and $\texttt{neu}_{ij}$ are defined analogously for positive and neutral opinion. Finally, we introduce a hard constraint:

$$\forall i, j \; \texttt{pos}_{ij} + \texttt{neg}_{ij} + \texttt{neu}_{ij} = 1$$

to ensure a single prediction is made per pair.

4.2.1 Inference with factions

Our first soft ILP constraint ψ_{fact} models that fact that entities in supportive social relations tend to share similar sentiment toward others [Lazarsfeld and Merton, 1954], and are often positive towards each other. For now, we assume access to a base extractor to provide such faction relations (Sec. 4.3.2 provides details of our pattern-based extractor). Figure 4.3a illustrates sample detections. We introduce a binary variable $\texttt{tie}_{ij}$, where $\texttt{tie}_{ij}$ = 1 denotes an extracted faction relationship. These variables are tied to the variables regarding sentiment via the following variables:

$$\texttt{tie\_same}_{ijk} = \texttt{tie}_{ij} \wedge \texttt{pos}_{ik} \wedge \texttt{pos}_{jk} + \texttt{tie}_{ij} \wedge \texttt{neg}_{ik} \wedge \texttt{neg}_{jk}$$

$$\texttt{tie\_diff}_{ijk} = \texttt{tie}_{ij} \wedge \texttt{pos}_{ik} \wedge \texttt{neg}_{jk} + \texttt{tie}_{ij} \wedge \texttt{neg}_{ik} \wedge \texttt{pos}_{jk}$$

$$\texttt{itself}_{ij} = \texttt{tie}_{ij} \wedge \texttt{pos}_{ij} - \texttt{tie}_{ij} \wedge \texttt{neg}_{ij}$$

which are used in the following objective term:

$$\psi_{fact} = \sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha_{itself} \cdot \texttt{itself}_{ij} + \sum_{k=1}^{n} (\alpha_{fact} \cdot (\texttt{tie\_same}_{ijk} - \texttt{tie\_diff}_{ijk})))$$

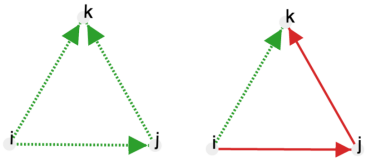


Figure 4.4: Balance theory constraints. When i is positive towards j, sharing same sentiment towards k define a balanced state. When i is negative towards j, differing opinions towards k define a balanced state. Red solid edges represent negative sentiment, green dotted edges represent positive sentiment.

This formulation enables the model to predict implicit sentiment by jointly considering factual and sentiment relations among other entity pairs, essentially drawing a connection between sentiment analysis and information extraction. Figure 4.3 visualizes this inference pattern.

4.2.2 Inference with sentiment relations

We also include constraints ψ_{social} in the objective that model social balance and reciprocity.

Balance theory constraints: Social balance theory [Heider, 1946] models the sentiment dynamics in an interpersonal network. In particular, in balanced states, entities on positive terms have similar opinions towards other entities and those on negative terms have opposing opinions. We introduce a set of variables to capture this insight: for example, the case where e_i is positive towards e_j is shown below (analogous when negative).

$$\texttt{pos\_same}_{ijk} = \texttt{pos}_{ij} \wedge \texttt{pos}_{ik} \wedge \texttt{pos}_{jk} + \texttt{pos}_{ij} \wedge \texttt{neg}_{ik} \wedge \texttt{neg}_{jk}$$

$$\texttt{pos\_diff}_{ijk} = \texttt{pos}_{ij} \wedge \texttt{neg}_{ik} \wedge \texttt{pos}_{jk} + \texttt{pos}_{ij} \wedge \texttt{pos}_{ik} \wedge \texttt{neg}_{jk}$$

and add the term ψ_{bl} to ψ_{social}.

$$\psi_{bl} = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}(\alpha_{bl} \cdot (\ \texttt{pos\_same}_{ijk} + \ \texttt{neg\_diff}_{ijk})$$

$$+ \alpha_{bad_{bl}} \cdot (\ \texttt{pos\_diff}_{ijk} + \ \texttt{neg\_same}_{ijk}))$$

A visualization of these constraints is in Figure 4.4.

	Faction	Balance	Reciprocity
POS	57%	64%	73%
NEG	60%	61%	78%

Table 4.1: Percentage of labels where each constraint holds. For example, positive on reciprocity means when $pos(e_i, e_j)$ is true, 73% of times $pos(e_j, e_i)$ is also true.

Reciprocity constraint: Reciprocity of sentiment has been recognized as a key aspect of social stability [Johnston, 1916; Gouldner, 1960]. To model reciprocity among the real world entities, we introduce variables:

$$\texttt{r\_same}_{ij} = \texttt{pos}_{ij} \wedge \texttt{pos}_{ji} + \texttt{neg}_{ij} \wedge \texttt{neg}_{ji}$$

$$\texttt{r\_diff}_{ij} = \texttt{pos}_{ij} \wedge \texttt{neg}_{ji} + \texttt{neg}_{ij} \wedge \texttt{pos}_{ji}$$

and introduce the following term ψ_r to the ψ_{social}.

$$\psi_r = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_r(\texttt{r\_same}_{ij}) + \alpha_{bad_r}(\texttt{r\_diff}_{ij})$$

4.2.3 Discussion

While many studies exist on homophily, social balance, and reciprocity, no prior work has reported quantitative analysis on the sentiment dynamics among the real world entities that appear in unstructured text. Thus we report the data statistics based on the development set in Table 4.1. We find that the global constraints hold commonly but are not universal, motivating the use of soft constraints (see Sec. 4.5).

4.3 Pairwise Base Models

The global model in Sec. 4.2 uses two base models, one for pairwise sentiment classification and the other for detecting faction relationships.

4.3.1 Sentiment Classifier

The entity-pair classifier considers a holder entity e_i, its mentions $m_{i1} \cdots m_{ip}$, a target entity e_j, its mentions $m_{j1} \cdots m_{jq}$, and document d. It predicts sent($e_i \rightarrow e_j$) $\in$ {positive, unbiased, negative}. The input is plain text and no gold labels are assumed; entity detection, dependency parse and co-reference resolution are automatic, and include common nouns and pronoun mentions (details in Sec. 4.4.1). We trained separate classifiers for pairs that co-occur in a sentence and those that do not, using a linear class-weighted SVM classifier with crowd-sourced data described in Sec. 4.4.2.

In what follows, we describe three different types of features we developed: dependency features, document features, and quotation features. Many of the features test the overall sentiment of a set of words (e.g., the complete document, a dependency path, or a quotation). In each case, we define the *sentiment label* for the text to be positive if it contains more words that appear in the positive sentiment lexicon than that appear in the negative one (and similarly for the negative label). We used MPQA sentiment lexicon [Wilson et al., 2005] for our study, which contains 2,718 positive and 4,912 negative lexicons.

Dependency Features We consider all dependency paths between the head word of e_i and e_j in each sentence, and aggregate over all co-occurring sentences. The features compute:

- The sentiment label of the path containing `dobj` and `nsubj_rev`, up to length three if the path contains sentiment lexicon words (e.g., *Olympic* hero *Skah* accuses *Norway* over custody battle.)
- The sentiment label of the path $e_i \uparrow$ `nsubj` $\downarrow$ `ccomp` $\downarrow$ `nsubj` $\downarrow e_j$, when it exists (e.g., *McCully* said any action against *Henry* is a matter entirely for TVNZ)
- The sentiment label of path when the path does not contain any named entity (e.g., *Nobel* winner , *Shirin Ebadi*)
- An indicator for the link `nmod:against`.

Document Features Previous work has shown that notions related to salience (e.g., proximity to sentiment words) can help to detect sentiment targets [Ben-Ami et al., 2014]. In our data, we found that an entity's occurrence pattern is highly indicative of being involved in sentiment, for example the most frequently

mentioned entity is 3.4 times more likely to be polarized and an entity in the headline is two times more likely to be polarized.

Pairwise features include the NER type of e_i and e_j and the percentage of sentences they co-occur in. We also use features indicating whether e_i and e_j (1) are mentioned in the headline and (2) appear only once in the document. When they are the two most frequent entities, we add the document sentiment label as a feature. For entity pairs that do not appear together in any sentence, we also include the rank of holder and target in terms of overall number of mentions in the document.

Quotation Features Quotations often involve subjective opinions towards prominent entities in news articles. Thus we include document-level features encoding this intuition. For example, the sentence *"We're pleased to put this behind us," said Michael DuVally* implies positive sentiment from DuVally. We extract direct quotations using regular expressions. We include the sentiment label of the direct quotation from the speaker to the entities in it, excluding entities that appear less than three times in the document. We add the sentiment label of the quotation as a feature to (speaker, the most frequent entity) pair as well.

To extract indirect quotations, we follow studies [Bethard et al., 2004; Lu, 2010] and use a list of 20 verbs indicating speech events (e.g., say, speak, and announce) to detect direct quotations and their opinion holders. We then add the sentiment label of words connected to e_j via a dependency path of length up to two that also includes the subject of quotation verb to e_j (e.g. *Hassanal* said that cooperation between *Brunei* and *China* were fruitful). We also include an indicator feature for whether e_i is the subject of the quotation verb.

4.3.2 Faction Detector

We use a simple pattern-based detector that extracts a faction relationship between a pair of entities if the dependency path between them either:

- contains only one link of modifier or compound label (`nmod`, `nmod : poss`, `amod`, `nn`, or `compound`).
- or contains less than three links and has a possessive or appositive label (`poss` or `appos`).

Example extractions for this approach, which we adopted for its simplicity and the fact that it works reasonably well in practice, are shown in Figure 4.3a. On average we detect 1.7 ties per document on a

	KBP	MPQA	Crowdsourced
Document count	154	54	914
Avg. sentence count	10.0	12.7	14.8
Avg. entity count	7.9	10.6	8.8
Avg. mentions / entity	3.6	2.7	3.5

Table 4.2: Corpus Statistics

small development set with roughly 30% recall and 60% precision. Improving performance and adding more relation types is an important area for future work.[1]

4.4 Data

We collected new datasets that densely label sentiment among entities in news articles, including: 208 documents, 2,226 sentences, and 15,185 entity pair labels. It complements existing datasets such as MPQA which provides rich annotations at the sentence-level Deng and Wiebe [2015b] and the recent KBP challenge which provides sparse annotations at the corpus-level Ellis et al. [2014], by providing document-level annotations for all entity pairs.

4.4.1 Document Preprocessing

All-pair annotation can be expensive, as there are N^2 pairs to annotate for each document with N entities. We determined that it would be more cost efficient to cover a large number of short documents than a small number of very long documents. We therefore selected articles with less than eleven entities from KBP and less than fifteen from MPQA and took the first 15 sentences for annotation. We used Stanford CoreNLP Manning et al. [2014] for sentence splitting, part-of-speech tagging, named entity recognition, co-reference resolution and dependency parsing. We discarded entities of type date, duration, money, time and number and merged named entities using several heuristics, such as merging acronyms, merging named entity of person type with the same last name (e.g., Tiger Woods to Woods). We merged names listed as alias in when there is an exact match from Freebase. We included all mentions in a co-reference chain with the named entity, discarding chains with more than one entity. The corpus statistics are shown in Table 4.2.

[1] We experimented with using relations from an external knowledge base (Freebase), but KB sparsity and entity linking errors posed major challenges.

4.4.2 Sentiment Data Collection

We annotated data using two methods: freelancers ($7.6 per article on average) covering all entity pairs and crowd-sourcing ($1.6 per article on average) covering a subset of entity pairs.

Evaluation Dataset We provide exhaustive annotations covering all pairs for the evaluation set. We hired freelancers from UpWork,[2] after examining performance on five documents. They labeled entity pairs with one of the following classes.

POS: positive towards the target.

NOTNEG: positive or unbiased towards the target.

UNB: unbiased towards the target

NOTPOS: negative or unbiased towards the target.

NEG: negative towards the target.

Here, we introduced the NOTPOS and NOTNEG classes to mark more subjective cases where we expect agreement might be lower. For example, one assigned NOTPOS to sentiment(Goldman, FINRA), *The FINRA said Goldman lacked adequate procedures to* . . . and another assigned NOTNEG to sentiment(Macalintal, Arroyo) in the next example. . . . *Arroyo's election lawyer, Romulo Macalintal*. Arguments could be made for NEG or POS, respectively, but the decision is inherently subjective and requires careful reading.[3]

We also asked annotators to mark the label as inferred when not explicitly stated but implied from the context or world knowledge. Allowing for inferred labels and finer-grained labels encouraged annotators to capture implicit sentiment. For each judgement, we acquired two labels. Inter-annotator agreement, in Table 4.4, is high for the relaxed metrics, confirming our intuitions about the ambiguity of the NOTNEG and NOTPOS labels.

For experiments, we combine the fine grained labels as follows: POS or NEG is assigned when both marked it as such. When only one of the annotators marked it, we assigned the weaker sentiment (POS to NOTNEG, NEG to NOTPOS). NOTNEG and NOTPOS are assigned when either annotator marked it without

[2] https://www.upwork.com

[3] In the construction of MPQA3.0 dataset, entity-entity/event sentiment corpus, even with iterative expert annotation, 31% of disagreements are caused by negligence.

Label	KBP	MPQA
POS	3.93	3.52
NOT NEG	5.73	8.06
UNBIASED	44.64	91.04
NOT POS	2.73	6.70
NEG	2.27	2.94

Table 4.3: Sentiment Label Statistics. Each count represents the average number per document.

	Exact	Strict	Relaxed
Positive	0.35	0.54	0.67
Negative	0.50	0.64	0.74

Table 4.4: Inter-annotator Agreement. Cohen's kappa score: Exact counts only exact matches, Strict counts allows NOT NEG labels to match POS, and Relaxed allows NOT NEG to match POS or UNBIASED (analogously for negative).

'Inferred' label. When the labels contradict in polarity or the labels are inferred weaker sentiment, UNB was assigned.

Crowdsourced Dataset We also randomly selected news articles from the Gigaword corpus,[4] and collected labels to train the base sentiment classifier (Sec. 4.3.1). We designed a pipelined approach, with three steps:

We used CrowdFlower,[5] where annotators were randomly presented test questions for quality control. We collected labels from three annotators for each entity pair, and considered labels when at least two agreed. The resulting annotation contains total 2,995 labels on 914 documents, 682 positive, 836 negative and 474 without sentiment, which we discarded.

4.4.3 Insights Into Data

This data supports the study of sentiment-laden entity pairs across sentence boundaries and inferred labels among entities, as we show here.

[4] LDC2014E13:TAC2014KBP English Corpus
[5] http://www.crowdflower.com

	POS	NOT NEG	NOT POS	NEG
KBP	25%	29%	30%	28%
MPQA	35%	49%	46%	50%

Table 4.5: Percentage of entity pairs that do not co-occur in a sentence.

	POS	NOTNEG	NOTPOS	NEG
KBP	70%	94%	88%	58%
MPQA	68%	74%	83%	66%

Table 4.6: Percentage of sentiment labels marked as inferred.

Sentiment Beyond Sentence Boundary Approximately 25% of polarized sentiment labels are between entities that do not co-occur[6] in a sentence (see Table 4.5). For example, in the article with headline 'Russia heat, smog trigger health problems',

> . . . "We never care to work with a future perspective in mind," *Alexei Skripkov* of the *Federal Medical and Biological Agency* said. "It's a big systemic mistake."

Skripkov never appears together with Russia in any sentence, but he manifests negative sentiment towards it. When a document revolves around a theme (in this example Russia), sentiment is often directed to it without being explicitly mentioned.

Inferred sentiment Annotators marked labels as inferred frequently, especially on less polarized sentiment (see Table 4.6). Various clues led to sentiment inference. For example, in the following document, we can read *Sam Lake*'s positive attitude towards *Paul Auster* from his 'citing' action:

> Ask most video-game designers about their inspirations . . . *Sam Lake* cites *Paul Auster*'s "*The Book of Illusions*"

Sentiment can also be inferred through reasoning over another entity.

> The *U.N.* imposed an embargo against *Eritrea* for helping insurgents opposed to the *Somali* government.

By considering relations with Eritrea, we can infer U.N. would be positive towards Somalia.

[6]This is an estimate due to co-reference resolution errors.

	Development Set (KBP)						KBP						MPQA					
	Positive			Negative			Positive			Negative			Positive			Negative		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
KM_Gold	90.9	2.5	4.8	93.8	8.6	15.8	93.9	4.3	8.3	93.5	6.6	12.4	61.5	1.3	2.5	90.0	5.2	9.8
Random	16.6	13.1	14.7	4.9	4.0	4.4	13.3	12.7	13.0	10.1	6.9	8.2	10.9	15.4	12.8	8.9	6.7	7.7
Sentence	**60.0**	16.3	25.7	21.7	**43.1**	28.8	40.9	20.6	27.4	21.0	31.4	25.2	18.9	3.7	6.2	16.7	18.2	17.4
Pairwise	47.3	36.9	41.4	25.6	36.8	30.2	36.2	**35.5**	35.9	27.6	**41.2**	33.1	**28.7**	23.0	25.6	**23.2**	16.3	19.2
Global	58.2	**37.9**	**45.9**	**37.2**	35.1	**36.1**	**45.5**	32.7	**38.1**	**34.6**	36.8	**35.7**	25.2	**29.3**	**27.1**	17.6	**24.4**	**20.4**

Table 4.7: Performance on the evaluation datasets: including implicit and explicit sentiment.

4.5 Experiment

4.5.1 Experimental Setup

Data and Metrics We randomly split the densely labeled KBP document set, using half as a test data and half as a development data. One half of the development set was used to tune hyper parameters,[7] and the other for error analysis and ablations. After development, we ran on the test sets composed of KBP documents and MPQA documents. For MPQA we did not create a separate development set and reserved all of the relatively modest amount of data for a more reliable test set. For the pairwise classifier, we report development results using five-fold cross validation on the training data.

We report micro-averaged precision, recall, and F-measure for both sentiment labels.

Comparison Systems We compare performance to two simple baselines and two adaptations of existing sentiment classifiers. The baselines include our base pairwise classifier (Pair) and randomly assigning labels according to their empirical distribution (Random).

The first existing method adaptation (Sentence) uses the publicly released sentence-level RNN sentiment model from Socher et al. [2013]. For each entity pair, we collect sentiment labels from sentences they co-occur in and assign a positive label if a positive-labeled sentence exists, negative if there exists more than one sentence with a negative label and no positives.[8]

We also report a proxy for doing similar aggregation over a state-of-the-art entity-entity sentiment classifier. Here, because we added our new labels to the original KBP and MPQA3.0 annotations, we can simply predict the union of the original gold annotations using mention string overlap to align the entities

[7] We used the following values (α_r, α_{bad_r}, α_{itself}, $\alpha_{faction}$, α_{bl}, $\alpha_{bad_{bl}}$) = (0.7, -0.8, 0.4, 0.5, 0.1, -0.5).

[8] Due to domain difference, the system predicted negative labels more (73% of sentences were classified as negative).

	Positive			Negative		
	P	R	F1	P	R	F1
ILP base	56.7	25.2	34.9	36.9	27.6	31.6
+ Reciprocity	53.5	30.0	38.4	33.9	33.9	33.9
+ Balance	49.6	30.4	37.7	32.0	32.8	32.4
+ Faction	58.9	30.2	39.9	37.6	33.9	35.6

Table 4.8: ILP constraints ablation study.

	Positive			Negative		
	P	R	F1	P	R	F1
All	34.5	39.7	36.9	35.7	37.6	36.6
- Dependency	32.9	32.1	32.5	31.7	38.5	34.8
- Document	32.6	41.0	35.8	39.4	23.8	28.0
- Quotation	33.6	39.5	36.3	34.5	34.6	34.6

Table 4.9: Pairwise classifier feature ablation study.

(KM_Gold). This provides a reasonable upper bound on the performance of any extractor trained on this data.[9]

Implementation Details We use CPLEX4[10] to solve the ILP described in Sec. 4.2. For computational efficiency and to avoid erroneous propagation, soft constraints associated with reciprocity and balance theory are introduced only on pairs for which a high-precision classifier assigned polarity. For the pairwise classifier, we use a class-weighted linear SVM.[11] We include annotated pairs, and randomly sample negative examples from pairs without a label in the crowd-sourced training dataset. We made two versions of pairwise classifiers by tuning weight on polarized classes and negative sampling ratio by grid search. One is tuned for high precision to be used as a base classifier for ILP (ILP base), and the other is tuned for the best F1 (Pairwise).[12]

[9] We consider this gold evaluation a direct proxy for the recent work Deng and Wiebe [2015a], which is the most related recent entity-entity sentiment model trained on the gold data whose predictions we are evaluating against.

[10] `http://tinyurl.com/joccfqy`

[11] `http://scikit-learn.org/`

[12] We use 10 as the weights for the polarized classes. Pairwise and base classifier for MPQA sampled 4%, base classifier for KBP sampled 10% of unlabeled pairs.

4.5.2 Results

Table 4.7 shows results on the evaluation datasets. The global model achieves the best F1 on both labels. All systems do significantly better than the random baseline but, overall, we see that entity-entity sentiment detection is challenging, requiring identification of holders, targets, and sentiment jointly. While the numbers are not directly comparable, the best performing system for KBP 2014 sentiment task achieved F1 score of 25.7.

The first row (KM_Gold) shows the comparison against gold annotations from different datasets, highlighting the differences between the task definitions. Our annotations are much more dense, while KBP focuses on specific query entities and MPQA has a much broader focus with less emphasis on covering all entity pairs. The high precision suggests that all of the approaches agree when considering the same entity pairs.

The global model also improves performance over the pairwise classifier (Pairwise) for both datasets, but we see very different behavior due to the different sentiment label distributions (see Table 4.3). The KBP data has many fewer unbiased pairs and many mistakes are from choosing the wrong polarity. For the pairwise classifier 17% of all predictions were assigned the opposite polarity. After the global inference, it is reduced to 11%, contributing to the gain in overall precision. For MPQA the base classifier has a more challenging detection task, due to relatively large amount of the unbaised pairs. Here, the best base classifier misses many pairs and the global model helps to fill in some of these gaps in recall.

In both cases, the document-level model often propagates correct labels by detecting easier, explicit expressions. For example, given the sentence *Buphavanh said Laos creates favorable conditions for Vietnamese companies*, the base classifier detected positive sentiment from Buphavanh to Vietnam, but not between Vietnam and Laos. By detecting the fact that Buphavanh is the prime minister of Laos, it infers the extra sentiment pairs.

We also did ablation studies to measure the contributions of different components. Table 4.8 shows ablations of each soft constraint. The faction constraint is the most helpful, improving both precision and recall for both labels. The reciprocity and social balance constraints tend to improve recall at the cost of precision. Table 4.9 shows ablations of the base classifier features. All features are helpful, with dependency features most helpful for positive labels, and quotation and document-level features more with negatives.

Sentiment expression detection error	21.0%
Missing world knowledge	19.3%
Named entity detection error	17.5%
Co-reference failure	14.8%
Propagation error	12.3%
Missing faction	7.0%

Table 4.10: Error Analysis on the development set.

Error Analysis We manually analyzed errors on 20 articles from the development set (Table 4.10). Our system failed when there were sentiment words not in the lexicon, or negated sentiment words. Capturing subtle sentiment expressions beyond sentiment lexicon should improve the performance. Preprocessing, as a whole, was the largest source of error. It includes co-reference failure and named entity error. Co-reference mistakes happen as a result of not resolving pronouns, referring expressions, as well as named entities co-references (e.g., Financial Industry Regulatory Authority to FINRA), or erroneously merging them. Lengthy quotations or nested mentions triggered co-reference error, affecting mostly recall. Named entity errors includes incorrect named entity detection (e.g., pro-Israel) and mention detection boundary errors. For example, we detected negative sentiment from Mexico to Pakistan from *Mexico condemns Pakistan series suicide bomb attacks.* While actual sentiment is positive. Finally, the ILP propagates sentiment labels erroneously at times. Our constraints often hold among entities of the same type, but are less predictive among entities of different types. For example, when a person supports a peace treaty, the treaty does not have sentiment towards him/her. For future work refining constraints based on entity type should help performance.

4.6 Related Work

Entity-Entity sentiment extraction is studied in the recent KBP sentiment task,[13] in that we aim to find opinion target and holder. While we study the complete document-level analysis over all entity pairs, the KBP task is formulated as query-focused retrieval of entity sentiment from a large pool of potentially relevant documents. Thus, their annotations focus only on query entities and relatively sparse compared to ours (see Sec. 4.5). Another recent dataset is MPQA 3.0 [Deng and Wiebe, 2015b], which captures various aspects

[13] `http://www.nist.gov/tac/2014/KBP/Sentiment`

of sentiment. Their sentiment pair annotations are only at the sentence-level and are therefore much sparser than we provide (see Sec. 4.5) for entity-entity relation analysis.

Several recent studies focused on various aspects of implied sentiment [Greene and Resnik, 2009; Mohammad and Turney, 2010; Zhang and Liu, 2011; Feng et al., 2013; Deng and Wiebe, 2014; Deng et al., 2014; Rashkin et al., 2016]. Deng and Wiebe [2015a] in particular introduced sentiment implicature rules relevant for sentence-level entity-entity sentiment. Our work contributes to these recent efforts by presenting a new model and dataset for document-level sentiment inference over all entity pairs.

Document-level Entity Relation Analysis Stoyanov and Cardie [2011] also studied document-level sentiment analysis based on fine-grained detection of directed sentiment. They aggregate sentence-level detections to make document-level predictions, while our we model global coherency among entities and can discover implied sentiment without direct sentence-level evidence. In the event extraction domain, previous research showed the effectiveness of jointly considering multiple sentences. Yang and Mitchell [2016] proposed joint extraction of entities and events with the document context, improving on the event extraction.

More recent work [Verga et al., 2018; Jia et al., 2019] presented neural approaches which can reason beyond sentence boundaries for biomedical relation extraction. Both methods use BiLSTM to encode the document, and then create entity representations from the BiLSTM output, which will be used for relation classification. Applying these neural models for document based sentiment analysis will be interesting direction for future work. Most work focuses on factual relations and events, while we primarily study sentiment relations.

Social Network Analysis While many previous studies considered the effect of social dynamics for social media analysis, most relied on an explicitly available social network structure or considered dialogues and speech acts for which opinion holders are given [Tan et al., 2011; Hu et al., 2013; Li et al., 2014; West et al., 2014b; Krishnan and Eisenstein, 2015]. Compared to the recent work that focused on relationships among fictional characters in movie summaries and stories [Chaturvedi et al., 2016; Srivastava et al., 2016; Iyyer et al., 2016], we consider a broader types of named entities on news domains.

4.7 Summary

This chapter explored implied high-level semantics from a document, specifically directed entity-entity sentiment classification. We presented a new crowd sourced dataset and an approach to interpreting sentiment among entities in news articles, with global constraints provided by social, faction and discourse context. Experiments demonstrated that the approach can infer implied sentiment from the document, but there is a large room for improving prediction accuracy. This work points toward promising directions for future work, including the incorporation of more varied types of factual relationships for entity-entity opinion analysis.

Chapter 5

Conclusion

This dissertation examines extracting entity information from unstructured text to a structured space. We introduced new tasks, which capture rich human communication in natural language text. By defining new tasks, collecting data and constructing a model for each task, we pushed the boundaries of entity information that can be extracted from natural language text. Each study addressed different units of text, starting from noun phrases, moving to sentences, and concluding with documents. While parsing noun phrases was more reliable, achieving a high accuracy, studying document-level semantics allowed learning high-level understanding of entity-entity interactions. In this last chapter, we discuss the limitations of proposed approaches and suggest directions for future work.

When humans read natural language text, we incorporate rich background knowledge and personal experiences about entities and the world they live in. The models presented in chapter 2 and chapter 4 have a limited capacity to access such world knowledge. Both models are driven by carefully designed lexical features, and the feature weights are trained to optimize task-specific objective with a modest amount of training instances. In contrast, our entity typing model in chapter 3 can access more background information on frequently occurring entities via large-scale distant supervision training dataset.

Our typing model's headword supervision objective, which predicts the missing headword of noun phrase from the sentence, is similar to the learning objective of recently proposed pre-trained language models [Devlin et al., 2019]. When trained with such masked language model objective on a large amount of text, language model learns the correlation between the entity and its context. For instance, the model recog-

nizes that “prime minister", or “Obama" will happen more frequently with events such as “administration", “debate", and “statement". Recent work has shown the power of such pre-trained language model [Peters et al., 2018; Radford, 2018; Devlin et al., 2019] for the panoply of NLP tasks, including coreference resolution, question answering, entity typing, and entity linking. Such large-scale language model, memorizing encyclopedic entity facts, showed a possibility to substitute knowledgebases. Trained from written text, however, language model is susceptible to a reporting bias [Gordon and Durme, 2013] and can hallucinate incorrect, but plausible facts. Incorporating rich entity information present in language models transparently and robustly into NLP system would be an exciting direction for future work.

Reasoning across different aspects of entity information (e.g., entity typing helping to coreference resolution, and social media data on named entities to provide background knowledge for information extraction) is largely unaddressed. In this thesis, we individually studied different aspects of entity analysis. While studying different aspects of entity understanding independently is an important stepping stone, a single unified model which can generate various entity centric information would be more robust and useful. For instance, as suggested in Deng and Wiebe [2016], understanding sentiment relations between entities can help to resolve challenging coreference cases. In chapter 4, we studied the other direction how factual relationship between entities can impact entity-entity sentiment. Similarly, we can think of injecting other consistency constraints based on entity attributes as an auxiliary input and design multitask objectives.

Lastly, we discussed the challenges with a fixed schema in this dissertation. We presented methods to learn a mapping between unrestricted natural language text and concepts in KB and to improve the coverage of concepts in KB. Even with an expressive schema and high-accuracy parser, challenges remain for reasoning with a KB. First, defining an ontology is a challenging task, especially for complex events and N-ary relations. While static and binary relations (i.e., entity X is born in entity Y, entity X is married to entity Y) can be easily defined in the ontology, complex events (i.e., sports competitions, elections, drug reaction involving multiple chemicals) are harder to define. For such complex information needs, reading comprehension could provide a more flexible solution. As users' information needs evolve, it would be preferable for a model to extract novel relations with very little human intervention (e.g., no need to label a large training set). Most existing information extraction approaches cannot extract relations that were not specified in advance. For emerging information needs, autoencoder approach [Iyyer et al., 2016; Han

et al., 2019] exploring schema-less relation discovery could bring promising solutions. There are multiple directions for future work, including exploring the interplay between a structured knowledgebase and a large scale pre-trained language model, transfer to languages other than English, extension from binary relations to n-ary relations, and generalizing relation extraction methods to domains with more complex information needs such as biomedical text.

Bibliography

Abhishek, Ashish Anand, and Amit Awekar. 2017. Fine-grained entity type classification by jointly learning representations and label embeddings. In *Proceedings of European Chapter of Association for Computational Linguistics*.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *COLING-ACL*.

Krisztian Balog and Robert Neumayer. 2012. Hierarchical target type identification for entity-oriented queries. In *Proceedings of the Conference on Information and Knowledge Management*.

Zvi Ben-Ami, Ronen Feldman, and Binyamin Rosenfeld. 2014. Entities' sentiment relevance. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the Empirical Methods in Natural Language Processing*, pages 1533–1544.

Steven Bethard, Hong Yu, Ashley Thornton, Vasileios Hatzivassiloglou, and Dan Jurafsky. 2004. Automatic extraction of opinion propositions and their holders. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*.

Qingqing Cai and Alexander Yates. 2013. Large-scale semantic parsing via schema matching and lexicon extension. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Snigdha Chaturvedi, Shashank Srivastava, Hal Daumé III, and Chris Dyer. 2016. Modeling evolving relationships between characters in literary novels. In *Proceedings of the National Conference on Artificial Intelligence*.

Eunsol Choi, Tom Kwiatkowski, and Luke Zettlemoyer. 2015. Scalable semantic parsing with partial ontologies. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*.

Eunsol Choi, Hannah Rashkin, Luke Zettlemoyer, and Yejin Choi. 2016. Document-level sentiment inference with social, faction, and discourse context. In *Proceedings of the ACL*. Association for Computational Linguistics.

Luciano Del Corro, Abdalghani Abujabal, Rainer Gemulla, and Gerhard Weikum. 2015. Finet: Context-aware fine-grained named entity typing. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*.

Tim Van de Cruys. 2014. A neural network approach to selectional preference acquisition. In *Proceedings of Empirical Methods in Natural Language Processing*.

Jia Deng, Wei ping Dong, Richard Socher, Li-Jia Li, Kehui Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR 2009*.

Lingjia Deng and Janyce Wiebe. 2014. Sentiment propagation via implicature constraints. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*.

Lingjia Deng and Janyce Wiebe. 2015a. Joint prediction for entity/event-level sentiment analysis using probabilistic soft logic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Lingjia Deng and Janyce Wiebe. 2015b. Mpqa 3.0: An entity/event-level sentiment corpus. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*.

Lingjia Deng and Janyce Wiebe. 2016. How can nlp tasks mutually benefit sentiment analysis? a holistic approach to sentiment analysis. In *WASSA@NAACL-HLT*.

Lingjia Deng, Janyce Wiebe, and Yoonjung Choi. 2014. Joint inference and disambiguation of implicit sentiments via implicature constraints. In *Proceedings of International Conference on Computational Linguistics*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the North American Association for Computational Linguistics*, abs/1810.04805.

Li Dong, Furu Wei, Hong Sun, Ming Zhou, and Ke Xu. 2015. A hybrid neural model for type classification of entity mentions. In *Proceedings of International Joint Conference on Artificial Intelligence*.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. In *Transactions of the Association for Computational Linguistics*.

Joe Ellis, Jeremy Getman, and Stephanie M Strassel. 2014. Overview of linguistic resources for the tac kbp 2014 evaluations: Planning, execution, and results. In *Proceedings of TAC KBP 2014 Workshop, National Institute of Standards and Technology*, pages 17–18.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam. 2011. Open information extraction: The second generation. In *IJCAI*.

Song Feng, Jun Seok Kang, Polina Kuznetsova, and Yejin Choi. 2013. Connotation lexicon: A dash of sentiment beneath the surface meaning. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria. Association for Computational Linguistics.

Matthew Gerber and Joyce Y Chai. 2010. Beyond nombank: a study of implicit arguments for nominal predicates. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1583–1592. Association for Computational Linguistics.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B. Dolan. 2005. The pascal recognising textual entailment challenge. In *MLCW*.

Daniel Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. 2014. Context-dependent fine-grained entity type tagging. *CoRR*, abs/1412.1820.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *AKBC@CIKM*.

Alvin W. Gouldner. 1960. The norm of reciprocity: A preliminary statement. *American Sociological Review*, 25(2).

Stephan Greene and Philip Resnik. 2009. More than words: Syntactic packaging and implicit sentiment. In *Proceedings the North American Chapter of the Association for Computational Linguistics*, Boulder, Colorado. Association for Computational Linguistics.

Nitish Gupta, Sameer Singh, and Dan Roth. 2017. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2671–2680.

Xiaochuang Han, Eunsol Choi, and Chenhao Tan. 2019. No permanent friends or enemies: Tracking relationships between nations from news. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, abs/1904.08950.

Fritz Heider. 1946. Attitudes and cognitive organization. *The Journal of psychology*, 21(1).

Lynette Hirschman, Marc Light, Eric Breck, and John D. Burger. 1999. Deep read: A reading comprehension system. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Dan Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the Conference of the Association of Computational Linguistics*.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: the 90% solution. In *Proceedings of the human language technology conference of the North American*

Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers, pages 57–60. Association for Computational Linguistics.

Xia Hu, Lei Tang, Jiliang Tang, and Huan Liu. 2013. Exploiting social relations for sentiment analysis in microblogging. In *Proceedings of the ACM international conference on Web search and data mining*. ACM.

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. 2016. Feuding families and former friends: Unsupervised learning for dynamic fictional relationships. In *Proceedings of North American Association for Computational Linguistics*.

Robin Jia, Cliff Wong, and Hoifung Poon. 2019. Document-level n-ary relation extraction with multiscale representation learning. In *Proceedings of the North American Association for Computational Linguistics*.

G. A. Johnston. 1916. *International Journal of Ethics*, 26(2).

Mandar Joshi, Eunsol Choi, Dan Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Thomas N. Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *Proceedings of International Conference on Learning Representations*, abs/1609.02907.

Vinodh Krishnan and Jacob Eisenstein. 2015. "You're Mr. Lebowski, I'm The Dude": Inducing address term formality in signed social networks. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, and Luke Zettlemoyer. 2013. Scaling semantic parsers with on-the-fly ontology matching. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Paul F Lazarsfeld and Robert K Merton. 1954. Friendship as a social process: A substantive and methodological analysis. *Freedom and control in modern society*, 18:18–66.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the SIGNLL Conference on Computational Natural Language Learning*.

Jiwei Li, Alan Ritter, and Eduard Hovy. 2014. Weakly supervised user profile extraction from twitter. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Xiao Ling and Daniel S Weld. 2012. Fine-grained entity recognition. In *Proceedings of Association for the Advancement of Artificial Intelligence*. Citeseer.

Bin Lu. 2010. Identifying opinion holders and targets with dependency parser in chinese news texts. In *Proceedings of the NAACL HLT Student Research Workshop*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

George A Miller. 1995. WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. 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: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Saif M. Mohammad and Peter D. Turney. 2010. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, CAAGET '10.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende,

Pushmeet Kohli, and James F. Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *HLT-NAACL*.

Shikhar Murty, Patrick Verga, Luke Vilnis, Irena Radovanovic, and Andrew McCallum. 2018. Hierarchical losses and new resources for fine-grained entity typing and linking. In *Proceedings of the Association for Computational Linguistics*.

Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. 2015. Compositional vector space models for knowledge base completion. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Yasumasa Onoe and Greg Durrett. 2019. Learning to denoise distantly-labeled data for entity typing. *Proceedings of the North American Association for Computational Linguistics*, abs/1905.01566.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 79–86.

Patrick Pantel, Rahul Bhagat, Bonaventura Coppola, Timothy Chklovski, and Eduard H. Hovy. 2007. Isp: Learning inferential selectional preferences. In *Proceedings of North American Chapter of the Association for Computational Linguistics*.

Robert Parker, David Graff, David Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition (ldc2011t07). In *Linguistic Data Consortium*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke S. Zettlemoyer. 2018. Deep contextualized word representations. *Proceedings of the North American Association for Computational Linguistics*, abs/1802.05365.

Rashmi Prasad, Nikhil Dinesh, Andrew Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K. Joshi, and Bonnie L. Webber. 2008. The penn discourse treebank 2.0. In *LREC*.

Maxim Rabinovich and Dan Klein. 2017. Fine-grained entity typing with high-multiplicity assignments. In *Proceedings of Association for Computational Linguistics*.

Alec Radford. 2018. Improving language understanding by generative pre-training.

Jonathan Raiman and Olivier Raiman. 2018. Deeptype: Multilingual entity linking by neural type system evolution. In *Association for the Advancement of Artificial Intelligence*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *Proceedings of Empirical Methods in Natural Language Processing*.

Hannah Rashkin, Sameer Singh, and Yejin Choi. 2016. Connotation frames: A data-driven investigation. In *Proceedings of Association for Computational Linguistics*.

Alexander Ratner, Stephen H. Bach, Henry R. Ehrenberg, Jason Alan Fries, Sen Wu, and Christopher Ré. 2019. Snorkel: rapid training data creation with weak supervision. *The VLDB Journal*, pages 1 – 22.

Siva Reddy, Mirella Lapata, and Mark Steedman. 2014. Large-scale semantic parsing without question-answer pairs. *Transactions of the Association for Computational Linguistics*, 2:377–392.

Xiang Ren, Wenqi He, Meng Qu, Lifu Huang, Heng Ji, and Jiawei Han. 2016a. Afet: Automatic fine-grained entity typing by hierarchical partial-label embedding. In *Proceedings Empirical Methods in Natural Language Processing*.

Xiang Ren, Wenqi He, Meng Qu, Clare R. Voss, Heng Ji, and Jiawei Han. 2016b. Label noise reduction in entity typing by heterogeneous partial-label embedding. In *Proceedings of Knowledge Discovery and Data Mining*.

Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare R. Voss, Heng Ji, Tarek F. Abdelzaher, and Jiawei Han. 2017. Cotype: Joint extraction of typed entities and relations with knowledge bases. In *Proceedings of World Wide Web Conference*.

Philip Resnik. 1996. Selectional constraints: an information-theoretic model and its computational realization. *Cognition*, 61 1-2:127–59.

Matthew Richardson, Christopher J. C. Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of Empirical Methods in Natural Language Processing*.

Sebastian Riedel, Limin Yao, Benjamin M. Marlin, and Andrew McCallum. 2013. Relation extraction with matrix factorization and universal schemas. In *Joint Human Language Technology Conference/Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL '13)*.

Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534. Association for Computational Linguistics.

D. Roth and W. Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of Conference on Natural Language Learning*.

Hinrich Schutze, Ulli Waltinger, and Sanjeev Karn. 2017. End-to-end trainable attentive decoder for hierarchical entity classification. In *Proceedings of European Chapter of Association for Computational Linguistics*.

Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2017. An attentive neural architecture for fine-grained entity type classification. In *Proceedings of the European Chapter of Association for Computational Linguistics (ACL)*.

Eyal Shnarch, Libby Barak, and Ido Dagan. 2009. Extracting lexical reference rules from wikipedia. 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 450–458. Association for Computational Linguistics.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2012. Wikilinks: A large-scale cross-document coreference corpus labeled via links to Wikipedia. Technical Report UM-CS-2012-015, University of Massachusetts, Amherst.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Empirical Methods in Natural Language Processing*.

Shashank Srivastava, Snigdha Chaturvedi, and Tom Mitchell. 2016. Inferring interpersonal relations in narrative summaries. In *Proceedings of the National Conference on Artificial Intelligence*.

Mark Steedman. 1996. *Surface Structure and Interpretation*. The MIT Press.

Veselin Stoyanov and Claire Cardie. 2011. Automatically Creating General-Purpose Opinion Summaries from Text. In *Proceedings of Recent Advances in Natural Language Processing*.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *EMNLP-CoNLL*.

Chenhao Tan, Lillian Lee, Jie Tang, Long Jiang, Ming Zhou, and Ping Li. 2011. User-level sentiment analysis incorporating social networks. In *Proceedings of Knowldege Discovery and Data Mining*.

Kristina Toutanova, Victoria Lin, Wen tau Yih, Hoifung Poon, and Chris Quirk. 2016. Compositional learning of embeddings for relation paths in knowledge base and text. In *Proceedings of the Annual Meeting of the 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. In *Proceedings of the Conference on Neural Information Processing Systems*.

Patrick Verga, Emma Strubell, and Andrew McCallum. 2018. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In *Proceedings of the North American Association for Computational Linguistics*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *ArXiv*, abs/1905.00537.

Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. Making neural qa as simple as possible but not simpler. In *Proceedings of the Conference on Computational Natural Language Learning*.

Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. 2014a.

Knowledge base completion via search-based question answering. In *Proceedings of World Wide Web Conference*.

Robert West, Hristo S Paskov, Jure Leskovec, and Christopher Potts. 2014b. Exploiting social network structure for person-to-person sentiment analysis. *In the Proceedings of Transactions of the Association for Computational Linguistics*.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing*.

Wenhan Xiong, Jiawei Wu, Deren Lei, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. Imposing label-relational inductive bias for extremely fine-grained entity typing. *Proceedings of the North American Association for Computational Linguistics*, abs/1903.02591.

Yadollah Yaghoobzadeh, Heike Adel, and Hinrich Schütze. 2017. Noise mitigation for neural entity typing and relation extraction. In *Proceedings of European Chapter of Association for Computational Linguistics*.

Yadollah Yaghoobzadeh and Hinrich Schütze. 2016. Corpus-level fine-grained entity typing using contextual information. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Bishan Yang and Claire Cardie. 2013. Joint inference for fine-grained opinion extraction. In *Proceedings of Association for Computational Linguistics*.

Bishan Yang and Tom Mitchell. 2016. Joint extraction of events and entities within a document context. In *North American Association for Computational Linguistics*.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2013. Universal schema for entity type prediction. In *Automatic KnowledgeBase Construction Workshop at the Conference on Information and Knowledge Management*.

Semih Yavuz, Izzeddin Gur, Yu Su, Mudhakar Srivatsa, and Xifeng Yan. 2016. Improving semantic parsing via answer type inference. In *Proceedings of Empirical Methods in Natural Language Processing*.

Dani Yogatama, Daniel Gillick, and Nevena Lazic. 2015. Embedding methods for fine grained entity type classification. In *Proceedings of Association for Computational Linguistics (ACL)*.

M Amir Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. 2012. Hyena: Hierarchical type classification for entity names. In *Proceedings of the International Conference on Computational Linguistics*.

Beñat Zapirain, Eneko Agirre, Lluís Màrquez i Villodre, and Mihai Surdeanu. 2013. Selectional preferences for semantic role classification. *Computational Linguistics*, 39:631–663.

Lei Zhang and Bing Liu. 2011. Identifying noun product features that imply opinions. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.