Entity Analysis with Weak Supervision: Typing, Linking, and Attribute Extraction

Xiao Ling

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

Doctor of Philosophy

University of Washington

2015

Reading Committee:

Daniel S. Weld, Chair

Luke Zettlemoyer

Sameer Singh

Program Authorized to Offer Degree:
Computer Science and Engineering

University of Washington

Abstract

Entity Analysis with Weak Supervision:
Typing, Linking, and Attribute Extraction

Xiao Ling

Chair of the Supervisory Committee:
Professor Daniel S. Weld
Computer Science and Engineering

With the advent of the Web, textual information has grown at an explosive rate. To digest this enormous amount of data, an automatic solution, Information Extraction (IE), has become necessary. Information extraction is a task of converting unstructured text strings into structured machine-readable data. The first key step of a general IE pipeline is often to analyze entities mentioned in the text before making holistic conclusions. To fully understand each entity, one needs to detect their mentions, categorize them into semantic types, connect them with their knowledge base entries, and identify their attributes as well as the relationships with others.

In this dissertation, we first present the problem of fine-grained entity recognition. Unlike most traditional named entity recognition systems using a small set of entity classes, *e.g.*, *person*, *organization*, *location* or *miscellaneous*, we define a novel set of over one hundred fine-grained entity types. In order to intelligently understand text and extract a wide range of information, it is useful to more precisely determine the semantic classes of entities mentioned in unstructured text. We formulate the recognition problem as multi-class, multi-label classification, describe an unsupervised method for collecting training data, and present the FIGER implementation.

Next, we demonstrate that fine-grained entity types are closely connected with other entity analysis tasks. We describe an entity linking system whose prediction heavily relies on these types and present a simple yet effective implementation, called VINCULUM. An extensive evaluation

on nine data sets, comparing VINCULUM with two state-of-the-art systems, elucidates key aspects of the system that include mention extraction, candidate generation, entity type prediction, entity coreference, and coherence.

Finally, we describe an approach to acquire commonsense knowledge from a massive amount of text on the Web. In particular, a system called SIZEITALL is developed to extract numerical attribute values for various classes of entities. To resolve the ambiguity from the surface form text, we canonicalize the extractions with respect to WordNet senses and build a knowledge base on physical size for thousands of entity classes.

Throughout all three entity analysis tasks, we show the feasibility of building sophisticated IE systems without a significant investment in human effort to create sufficient labeled data.

TABLE OF CONTENTS

Page

List of Figures iii

List of Tables iv

Chapter 1: Introduction 1
- 1.1 Main Tasks of IE 2
- 1.2 General Challenges of IE 5
- 1.3 Fine-Grained Entity Recognition 7
- 1.4 Design Challenges for Entity Linking 8
- 1.5 Size Attribute Extraction 8
- 1.6 Contributions 9

Chapter 2: Fine-Grained Entity Recognition 11
- 2.1 Introduction 11
- 2.2 Fine-Grained Entity Recognition 12
- 2.3 Experimentation 16
- 2.4 Related Work 21
- 2.5 Discussion 24
- 2.6 Summary 24

Chapter 3: Design Challenges in Entity Linking 26
- 3.1 Introduction 26
- 3.2 No Standard Benchmark 27
- 3.3 No Annotation Guidelines 31
- 3.4 A Simple & Modular Linking Method 35
- 3.5 Experiments 40
- 3.6 Related Work 50

3.7 Summary 53

Chapter 4: Attribute Extraction for Physical Object Size 54
4.1 Introduction 54
4.2 Automatically Constructing a Knowledge Base on Object Sizes 56
4.3 Experiments 60
4.4 Related Work 66
4.5 Discussion 68
4.6 Summary 70

Chapter 5: Conclusion 71
5.1 Limitations of This Work 71
5.2 Future Directions 73

Bibliography 74

LIST OF FIGURES

Figure Number Page

2.1 System architecture of FIGER. 12
2.2 112 tags used in FIGER. 13
2.3 Precision / Recall curves for relation extraction. 21

3.1 Entities divided by their types. For named entities, the solid squares represent 4 CoNLL(AIDA) classes; the red dashed squares display 3 TAC classes; the shaded rectangle depicts common concepts. 34
3.2 The process of finding the best entity for a mention. All possible entities are sifted through as VINCULUM proceeds at each stage with a widening range of context in consideration. Details of the approach are presented in Algorithm 1. 37
3.3 Recall@k on an aggregate of nine data sets, comparing three candidate generation methods. 42
3.4 Recall@k using CrossWikis for candidate generation, split by data set. 30 is chosen to be the cut-off value in consideration of both efficiency and accuracy. 42

4.1 Overview of SIZEITALL. The extractor is applied to the source corpus and the resulting extractions are consolidated into a knowledge base. 56

LIST OF TABLES

Table Number Page

2.1 List of features used in FIGER. 17

2.2 F1 scores for entity recognition. 19

2.3 The list of relation types used in the relation extraction experiment. 22

3.1 Characteristics of the nine NEL data sets. 28

3.2 A sample of papers on entity linking with the data sets used in each paper (ordered chronologically). 30

3.3 Performance(%, R: Recall; P: Precision) of the correct mentions using different **mention extraction** strategies. ACE and MSNBC only annotate a subset of all the mentions and therefore the absolute values of precision are less useful than the relative order between methods. 41

3.4 Performance (%) after **incorporating entity types**, comparing two sets of entity types (NER and FIGER). Using a set of fine-grained entity types (FIGER) generally achieves better results. 41

3.5 Performance (%) after re-ranking candidates using coherence scores, comparing two **coherence measures** (NGD and REL). "no COH": no coherence based re-ranking is used. "+BOTH": an average of two scores is used for re-ranking. Coherence in general helps: a combination of both measures often achieves the best effect and NGD has a slight advantage over REL. 45

3.6 **End-to-end performance**: We compare VINCULUM in different stages with two state-of-the-art systems, AIDA and WIKIFIER (%). The column "Overall" lists the average performance of nine data sets for each approach. CrossWikis appears to be a strong baseline. VINCULUM is 0.4% shy from WIKIFIER, each winning in four data sets; AIDA tops both VINCULUM and WIKIFIER on AIDA-test. 46

3.7 Comparison of entity linking pipeline architectures. VINCULUM components are described in detail in Section 3.4, and correspond to Figure 3.2. Components found to be most useful for VINCULUM are highlighted. 47

3.8 We divide linking errors into **six error categories** and provide an example for each class. 48

3.9 **Error analysis:** We analyze a random sample of 250 of VINCULUM's errors, categorize the errors into six classes, and display the frequencies of each type across the nine datasets. 49

4.1 Lexico-syntactic patterns for unary extraction. 58

4.2 A summary of unary datasets. In JH, only height and length attributes are annotated. In TT15U, one single measure was labeled. In KO11U, all three dimensions of the objects are reported if possible (*e.g.*, the length of `tea-bag` is neglected). 61

4.3 Size extraction performance. In the rows of TT15U, KO11U and JH, *average absolute log difference* and coverage numbers are shown for each method. The last row show the percentage of the estimates falling within one standard deviation from the mean estimated from crowdsourced labels. Best results for each dataset are in bold. The performance of TT15U is from the best number in Figure 1(a) of [135]. 63

4.4 A summary of binary datasets. 64

4.5 Size comparison performance. Both accuracy and coverage are reported in each cell. Best results for each dataset are in bold. 66

ACKNOWLEDGMENTS

There are many people I owe a debt of gratitude for helping me through the program.

First, I am very fortunate to have Dan as my advisor. He taught me how to do research, focusing on the important problems and making solid progress while aiming at the ultimate goal. I am always impressed by his curiosity, enthusiasm, and insights to research problems. Dan is one of my lifelong role models and the best advisor one can hope for.

I am lucky to have a great committee. Pedro's sharp comments are always illumintaing. I will keep in mind his famous question, "What problem will you work on if you only have five years to solve AI?" Luke has been warm and helpful whenever I need advice and discussion. Sameer's attention to details is refreshing and his limitless energy constantly amazes me. The last chapter of this dissertation was shaped from a brainstorming discussion with Yejin. I also learned a lot from Emily's Ling566 class.

During the PhD program, it is my great honor to work with Peter Clark at AI2, Alon Halevy and Cong Yu at Google, Evgeniy Gabrilovich, Bo Pang, and Fernando Diaz at Yahoo.

UW CSE has a great group of AI and NLP researchers. Oren Etzioni is a wonderful researcher and earns the credit for the name of SizeItAll. I attended many of Oren's KnowItAll group meetings. The conversations with Mausam and Stephen Soderland often expand my horizons and benefit me with constructive critiques.

I enjoy the time hanging out with my fellow gradudate student collegues at LoudLab, KnowItAll, and LIL. Special thanks to Fei Wu, Hoifung Poon, Raphael Hoffmann, Peng Dai, Alan Ritter, Tony Fader, Janara Christensen, Congle Zhang, Chris Lin, Jonathan Bragg, Adrienne Wang, Yoav Artzi, Kenton Lee, Mark Yatskar, Eunsol Choi, Justin Huang, Victoria Lin, Gagan Bansal.

UW CSE has an awesome group of staff. Lindsay Michimoto is the best academic advisor I have ever met. She can answer any question I have. I am also grateful for the assistance by Elise DeGoede, Andrei Stabrovski, Alicen Smith, and Franklin.

Outside the academic life, I had fun with my friends and receive much help from them. Many thanks to Xu Miao, Kevin Lai, Hao Lu, Hao Du, Yanping Huang, Zhe Xu, Yuyin Sun, Jia Wu, Yang Yang, Wei Shi, Guilan Weng, Bai Xiao, Mu Li.

It would be impossible for me to accomplish anything without the steadfast support from my family. My parents provided abundant encouragement and have been very supportive. Jiamei has been extremely understanding along the way, including maintaining a two year long distance relationship. The dissertation is dedicated to them.

DEDICATION

to my beloved parents, Haiyan and Rong

to my dear wife, Jiamei

Chapter 1

INTRODUCTION

Humans get the majority of their knowledge by reading. Understanding and organizing textual information for decision making is an essential task in many areas. Investment analysts read financial reports of large companies and daily news articles to make predictions about the stock market. Scientific researchers learn about recent observations and breakthroughs in the community from the literature. Military strategists digest intelligence reports to forecast future events and decide the appropriate strategic maneuvers. With the advent of the Web, however, it is becoming an overwhelming task to read and keep track of the explosive growth of information. Statistics show that, almost 10,000 news articles are published every day[1]; Twitter users tweet nearly 300,000 times per minute[2]; millions of papers are published in PubMed, with an average of around 5,500 papers accepted in 2012[3]; in 2014, approximately 8,000 papers per month were submitted to arXiv (arxiv.org), an open repository of electronic preprints[4].

Such information overload makes it imperative to find an automatic solution for natural language understanding (NLU). The ultimate goal of an NLU system is to have the same ability as human to read and understand written language. Consequently, machines could answer questions and provide summaries, while humans could focus on more intelligently challenging tasks.

An early attempt in an automatic solution was a series of Message Understanding Conferences (MUC [51]) hosted by Defense Advanced Research Projects Agency (DARPA). It aimed at encouraging development of new methods for automatic *Information Extraction* (IE). Information

---

[1] A Bing newswire search returns on average around 10,000 articles per day [145].

[2] `http://www.internetlivestats.com/twitter-statistics/`

[3] `http://dan.corlan.net/cgi-bin/medline-trend?Q=`

[4] `http://arxiv.org/stats/monthly_submissions`

extraction is in general to convert unstructured text strings into a structured, machine-readable format (*e.g.*, relation database tables).[5] Along with the growth of the Web, the IE problems started as simple template matching for dozens or hundreds of documents [51] and gradually turned into holistic knowledge extraction and deep language understanding from (hundreds of) millions of documents (e.g. [142, 9]).

1.1 Main Tasks of IE

The key first step of an IE system[6] is recognizing and analyzing the entities in natural language text. We must detect their mentions, categorize them into semantic types, connect them with respect to some knowledge base (KB), identify their attributes or relationships with other entities, and so on. Entity mentions[7] are often the basic units on which other IE systems are built. For example, a relation extraction system takes tuples of identified entity mention and classifies them into relation types [114, 40, 64, 71]. The semantic types of the mentions are then predicted [45], commonly used as salient features for down-stream NLP applications, such as coreference resolution [54, 77], relation extraction [69, 96], and selectional preference [117]. The entity identities not only canonicalize the extractions, but also reciprocally help improve the quality of extractions because of the additional entity-specific information encoded in the KB [125, 55, 21, 71]. Furthermore, the attributes of an entity provide a summary of the entity and its interactions with other entities. For instance, the Knowledge Card feature by Google provides a small snippet of biographical facts when a user searches for some celebrity. In the following sections, we briefly review these major tasks of IE.

---

[5]Broadly speaking, an ideal NLU system is expected to comprehend spoken language as well as write and communicate with human. A broader definition of IE includes extracting information from images, audios and videos. Both are beyond the scope of this work.

[6]after necessary syntactic preprocessing including HTML tag stripping, tokenization, sentence splitting, Part-of-Speech tagging, syntactic parsing, etc.

[7]We use the term *entity* represent a canonical form of an entity and typewriter font for a specific instance in a KB, *e.g.*, `KB:Entity`. An *entity mention* or simply a *mention* refers to the actual noun phrase appearing in text and is denoted in quotes, e.g., "Mention."

1.1.1 Named Entity Recognition

Named Entity Recognition (NER) is a task of identifying regions of text (*mentions*) corresponding to entities and classifying them into a predefined set of classes. A common set of entity types is *person*, *location*, *organization*, and *miscellaneous*. The results are useful for visualizing the document as well as for downstream tasks. For example, many *relation extraction* pipelines start by using NER to identify a relation's possible arguments in a sentence and then classify or extract the relation type [9, 114, 64]. Naturally, the type of the arguments are informative features when determining which relation holds (if any) [69, 71, 145].

1.1.2 Named Entity Linking

Although NER provides a class-level understanding of the entities, it is impossible to retrieve information specific to a certain entity nor its true identity other than its string form. Named Entity Linking (NEL) is to disambiguate a mention in a given context to its corresponding entity in a common knowledge base such as Wikipedia[8] [95, 73, 113, 63, 21, 84]. For example, consider the sentence:

JetBlue begins direct service between Barnstable Airport and JFK International.

where "JetBlue" should be linked to the commonly known entity `KB:JetBlue`, "Barnstable Airport" to `KB:Barnstable Municipal Airport`, and "JFK International" to `KB:John F. Kennedy International Airport`.

Equipped with entity-specific information already stored in the KB, the links not only provide semantic annotations to human readers but also a machine-consumable representation of basic semantic knowledge in the text. Many NLP applications can benefit from such links, such as distantly-supervised relation extraction [28, 114, 64, 71] that uses NEL to create training data, and some coreference systems that use NEL toverify the consistency within each coreference cluster [55, 148, 36].

---

[8] `http://wikipedia.org`

1.1.3 Attribute Extraction

Attribute extraction is a task of identifying attribute values for a class of entities. For instance, what is *part of* a car? What is the typical *height* of a tree? This task is very close to the relation extraction task (also known as relationship identification), which is to detect diverse semantic relationships between entities (*e.g.*, are John and Amy *married*?). Historically, both tasks aim at extracting relations between words. Some early approaches proposed lexical patterns to detect hypernymy (*is-a*) [60] and meronymy (*has-part*) [10, 48]. Other methods include distributional relation analysis [139], pattern bootstrapping [14, 3], etc. One notable line of research is Open Information Extraction (Open IE) [9]. Open IE is not restricted to a predefined set of relation types. Rather, it extracts relational triples with free-text predicates. With the existence of large-scale knowledge bases, a distant supervision technique attracts lots of attention [28, 142, 16, 96, 114, 64, 134]. The basic idea is to match text with some KB facts and use matching sentences as potential positive examples for training relation extractors. This simple heuristic has achieved great success because it avoids laborious data labeling. However, since the string matching heuristic works the best with named entity pairs, most of the relation extraction research became limited to named entities. In contrast, attribute extraction targets classes of entities. A common attribute for a class of entities often takes a range or a set of values (*e.g.*, the price of a smart phone varies from $50 to $1,000; vases can be *made of* ceramics, glass or metals).

1.1.4 Other Tasks

Other main tasks of IE include *coreference resolution* — detecting expressions referring to the same entity [54, 77], *temporal analysis* — putting events described in text onto the timeline [85, 78] and event extraction [11, 118, 145]. All the tasks are to some extent interconnected and could benefit from the output of other tasks, but a thorough exploration is beyond the scope of this work.

1.2 General Challenges of IE

The success of the MUC conferences stimulated more research work with methods ranging from supervised classification [128] to sequence modeling of token tagging [75]. It also motivated new research programs, such as Automatic Content Extraction (ACE, from 1999 to 2007), as well as the creation of a number of datasets (*e.g.*, [97]), which facilitated evaluation on IE research and moved forward the field.

In the seminal paper by Craven *et al.* [29], the authors proposed an ultimate goal of building a machine-readable knowledge base from the whole Web. It required two inputs, a fixed ontology that specifies interested entity types and relation types and a set of labeled data to enable machine learning. This work attracted a lot of research interest (*e.g.*, [39, 98, 110]) and was followed by a set of impactful research projects, such as DARPA-sponsored programs Machine Reading, Deep Exploration and Filtering of Text (DEFT), and Big Mechanism. In [98], Mitchell *et al.* introduced the concept of *macro reading*. In contrast with traditional *micro reading*, which extracts every bit of information from each sentence, macro reading largely relies on corpus statistics and circumvents complex linguistic analysis. The motivation of machine reading and information extraction is well aligned with the grand vision of *semantic web*, that is, to share the Web data in a common format.

As the accuracy of a variety of IE methods improve over time, it becomes more realistic and feasible to automatically construct knowledge bases (AKBC). Much research effort was invested in the direction, including a series of AKBC workshops from 2010 to 2014 and an annual evaluation called Knowledge Base Population (KBP) evaluation hosted by NIST since 2009. One notable KBP evaluation is the cold start KBP track, where participants are required to extract information from an unseen document collection and synergize a knowledge base from the extractions.

Despite the promise of IE, several major challenges are faced by researchers:

- **Ambiguity:** Natural language is inherently ambiguous. A word can denote distinct meanings in two sentences. For example, consider the following sentences,

  I opened the door *lock*.

The ship passed through a *lock*.

The word *lock* should be interpreted as "a fastener fitted to a door" and "an enclosure that controls the water level", respectively. Levesque [79] showed that a pronoun can refer to different antecedent entities given a minimal change of the sentence. Take one example in [79],

Jim comforted Kevin because *he* was so upset.

where *he* is clearly referred to Kevin. However, a simple replacement of "comforted" by "yelled at" resulted in a dramatic change of the reference of *he*. Not only does ambiguity occur in common words, specific names can also have multiple meanings. "Seattle" is commonly used as a city name whereas in the sentence "Seattle beat Portland yesterday," the name is metonymically used as a reference to a sports team, `KB:Seattle Sounders`. On the other hand, the same entity can have multiple synonyms. For instance, `KB:University of Washington` is often called "UW" in short.

- **Knowledge:** It is practically impossible for a computer to tell apart the two uses of *lock* in the first example without the knowledge about the two definitions. To formalize such knowledge, early approaches often turn to manual construction [90, 42], which are very expensive and not scalable. Recently, a dramatically different approach achieved great progress. Wikipedia, contributed by a group of non-expert volunteers from the Web, has quickly grown into a huge repository of over 4 million articles since its birth in 2001. Despite the success, it still remains challenging to fully leverage the knowledge encoded in Wikipedia. For example, the Wikipedia category hierarchy is disorganized [103]; the Wikipedia infoboxes are in free text and unnormalized [142]. Some attempts to canonicalize and extend Wikipedia such as DBpedia [6], Yago [133] and Freebase [12] had great influences on recent IE work (*e.g.*, [64, 74]). On the other hand, knowledge bases are often incomplete. Xu *et al.* [143] checked a random set of true relation triples and found that most are missing from the knowledge base. Similarly, Choi *et al.* [24] showed that 27% of a sample of entities are absent from the

knowledge base. The problem seems circular since machines would have the knowledge if they can read. One possible solution is bootstrapping — teaching machines with existing data and incrementally accumulate knowledge from new extractions.

- **Supervision for training extraction models:** Nowadays, statistical methods dominate the development of the extraction models and most approaches use supervised learning [75, 45, 112, 113, 63]. One key factor of building a successful extractor is the supervision that guides the tuning of the model parameters to achieve an optimal performance on the training data. Unfortunately, large amounts of human labels are often difficult to obtain due to its prohibitive cost in money and time. For example, the widely used Penn Treebank took almost ten years to finish [88]. Even though some datasets are created from various shared tasks and competitions [137, 2], they are still far from sufficient. It is well known that language has a long tail of linguistic variations. Therefore, collecting a set of labels with a reasonable coverage is often extremely difficult or sometimes infeasible.

1.3 Fine-Grained Entity Recognition

Despite considerable amount of research on named entity recognition, the majority of previous NER research has focused on a limited number of these types. For instance, MUC-7 [22] considered 3 classes, *person, location* and *organization*. CoNLL-03 added a miscellaneous type [137], and ACE introduced geo-political entities, weapons, vehicles and facilities [35]. Other examples include Ontonotes' categorization into 18 classes [65] and BBN's 29 answer types [140]. While these representations are more expressive than the person-location-organization standard, they still fail to distinguish entity classes which are common when extracting hundreds or thousands of different relations [64, 19]. Furthermore, as one strives for finer granularity, their assumption of mutual exclusion breaks (*e.g.*, Monopoly is both a *game* and a *product*).

In Chapter 2, we propose to extend the traditional set of entity types with finer-grained types. The creation of the type set is largely motivated by downstream applications. We would like the fine-grained entity types to have a stronger discrimination power so that, for example, using these

types as features could be informative in a relation extraction problem. We also propose to allow each entity mention to have multiple types. To achieve this goal, we create a large set of fine-grained overlapping entity types and build a multi-class multi-instance classifier for type prediction. In addition, we present an automated method of obtaining labeled data for training the classifier and an implementation of the system, FIGER.

1.4 Design Challenges for Entity Linking

Named entity recognition, fine-grained entity recognition and named entity linking may seem different at first glance; in fact we can view them as the same task of partitioning entity mentions into predefined classes, except in different granularities. NER provides a coarse-grained partition of all the entity mentions into several classes. Fine-grained entity recognition further shatters some partitions and produces overlapping partitions. NEL is on the extreme end where each entity itself belongs to one unique partition. In Chapter 3, we propose to build a multi-stage pipeline for entity linking called VINCULUM. The system incorporates the entity type information into the linking decision. Coreference and coherence of the entity mentions are also taken into account.

After investigating a variety of papers on entity linking and several published datasets, we find that there is surprisingly little understanding about state-of-the-art performance. In Chapter 3, we discuss three main reasons for this confusion. To have a better understanding of the importance of various techniques, we analyze our proposed approach in a series of ablation studies and compare it to the two leading sophisticated EL systems on a comprehensive set of nine datasets.

1.5 Size Attribute Extraction

Despite numerous methods designed for relation extraction on named entities or common nouns, there is very limited work on attribute extraction for numerical values. There is often one single correct value for a relation (or an attribute) of a specific entity (*e.g.*, the *capital* of a country and the *birthdate* of a person). In contrast, multiple or a range of values can all be correct for a numerical attribute. For instance, the *width* of passenger cars could varies from 39 inches to 80 inches; the

width of a typical passenger car is around 71 inches. Therefore, it is important to characterize the attribute values with an appropriate distribution.

In Chapter 4, we study a specific problem of extracting size attributes from text. We present SIZEITALL that extracts information about numerical size attributes from the Web text using lexico-syntactic patterns, estimates the distributions for each entity attribute and builds a knowledge base of the dimensions of thousands of entity classes. The resulting knowledge base could be used as a commonsense prior in other tasks. For instance, the knowledge base can inform an object recognition system if an object recognized as "mouse" looks larger than another object predicted as "elephant."

1.6 Contributions

In this dissertation, we investigate three important tasks of entity analysis for IE, namely, entity typing, entity linking and attribute extraction, and report some attempts to these problems. We also present experimental results to show the effectiveness of our proposed methods.

In summary,

- We introduce a novel problem of fine-grained entity recognition. We create a set of 112 fine-grained entity types, which are expectedly useful both to human understanding and other NLP applications. We implement an end-to-end system FIGER, compare it with two state-of-the-art baselines and show its effectiveness in a downstream application.

- We present a simple yet effective, modular, unsupervised system, VINCULUM, for entity linking and investigate several key aspects of the system including mention extraction, candidate generation, entity type prediction, entity coreference, and coherence between entities. We also compare VINCULUM to two state-of-the-art systems on an extensive evaluation of nine datasets.

- We take an initial step in a numerical attribute extraction problem. We implement a system SIZEITALL to extract size attributes from the Web text and build a knowledge base of object

sizes. We compare the constructed KB with a sample of human annotations and show the superior performance against two baseline systems. We also show the effectiveness of the KB in an end task of size comparison.

One universal theme throughout this dissertation is our effort to minimize the need for human labeling when building a system. In Chapter 2, FIGER is trained on an automatically created set of labeled entity mentions with no human intervention. In Chapter 3, we build a deterministic entity linking system that requires no supervision and yet achieves the performance close to state-of-the-art machine learned systems. In Chapter 4, SIZEITALL extracts numerical attribute values from the Web using simple lexico-syntactic patterns. Overall, we show that it is feasible to build NLP systems without a significant investment in time and resources to create sufficient labeled data.

We make all the systems presented in this dissertation available for future research, FIGER (`https://github.com/xiaoling/figer`), VINCULUM (`https://github.com/xiaoling/vinculum`), and SIZEITALL (`https://github.com/xiaoling/sizeitall`).

Chapter 2

FINE-GRAINED ENTITY RECOGNITION

2.1 Introduction

Detecting entity mentions and classifying them into semantic types is one of the key problems in information extraction. Unfortunately, the majority of previous NER research has focused on a limited number of these types, from a classic set of three classes, *person, location* and *organization* [137], to BBN's 29 answer types [140]. These coarse-grained representations fail to distinguish entity classes which are common when extracting hundreds or thousands of different relations [64, 19]. Furthermore, as one strives for finer granularity, their assumption of mutual exclusion breaks (*e.g.*, Monopoly is both a *game* and a *product*).

In this chapter, we address three main challenges impeding the development of a fine-grained entity recognizer: selection of the tag set, creation of training data, and development of a fast and accurate multi-class labeling algorithm. First, we curate a set of 112 unique tags based on Freebase [12] types. The second challenge, creating a training sets for these tags, is clearly too large to rely on traditional, manual labeling. Instead, we exploit the anchor links in Wikipedia text to automatically label entity segments with appropriate tags. Next, we use this heuristically-labeled training data to train a conditional random field (CRF) model for segmentation (identifying the boundaries of text that mentions an entity). The final step is assigning tags to the segmented *mentions*; for this purpose, we use an adapted perceptron algorithm for this multi-class multi-label classification problem.

In order to evaluate our approach empirically, we build a complete system called FIGER and consider two questions: how accurately can FIGER assign tags? And do the fine-grained tags matter? To answer the first question we compare FIGER with two alternative approaches: Stanford's coarse-grained NER system [45] and Illinois' Named-Entity Linking (NEL, aka Wikifier) system [113].

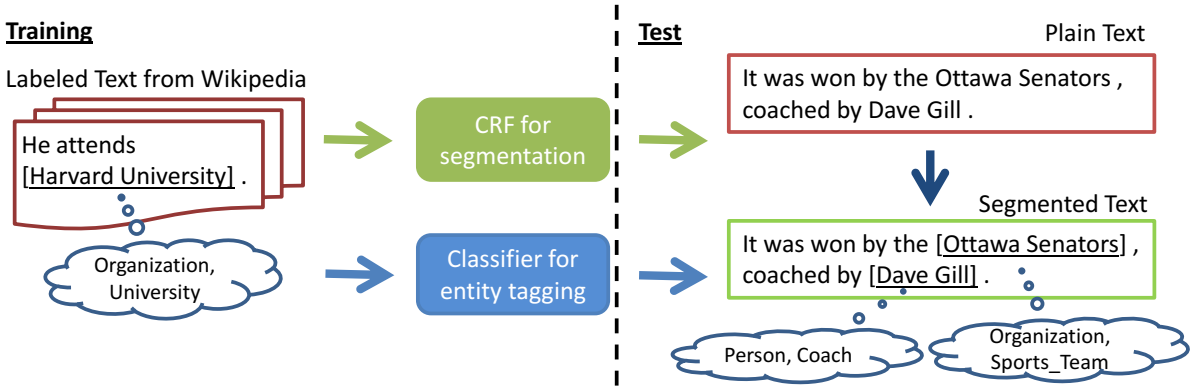

Figure 2.1: System architecture of FIGER.

Although the NEL approach works well for common objects (*e.g.*, Hillary Clinton), our results demonstrate FIGER's advantages on the "long tail" of uncommon entities. To answer the second question, we augment a state-of-the-art relation-extraction system, MultiR [64], to accept the types predicted by FIGER as features for the arguments of each potential relation mention. Here we see a large boost in F1 score compared to using the Stanford NER tags alone.

2.2 Fine-Grained Entity Recognition

Before describing the whole system, we state the problem at hand. Our task is to uncover the type information of the entity mentions from natural language sentences. Formally speaking, given a sentence s as a sequence of tokens $< w_1, \ldots, w_n >$, we need to identify the entity mentions $\{m_1, \ldots, m_k\}$, each of which m_i is a subsequence of s ($< w_1^i, \ldots, w_{n_i}^i >$), as well as associate the correct entity types t_i with each m_i.

2.2.1 Overview

Figure 2.1 is the overview diagram of our system FIGER. We divide the whole process into a pipeline. Given a sentence in plain text as input, we first segment the sentence and find the candidates for tagging. Second, we apply a classifier to the identified segments and output their tags. Traditional

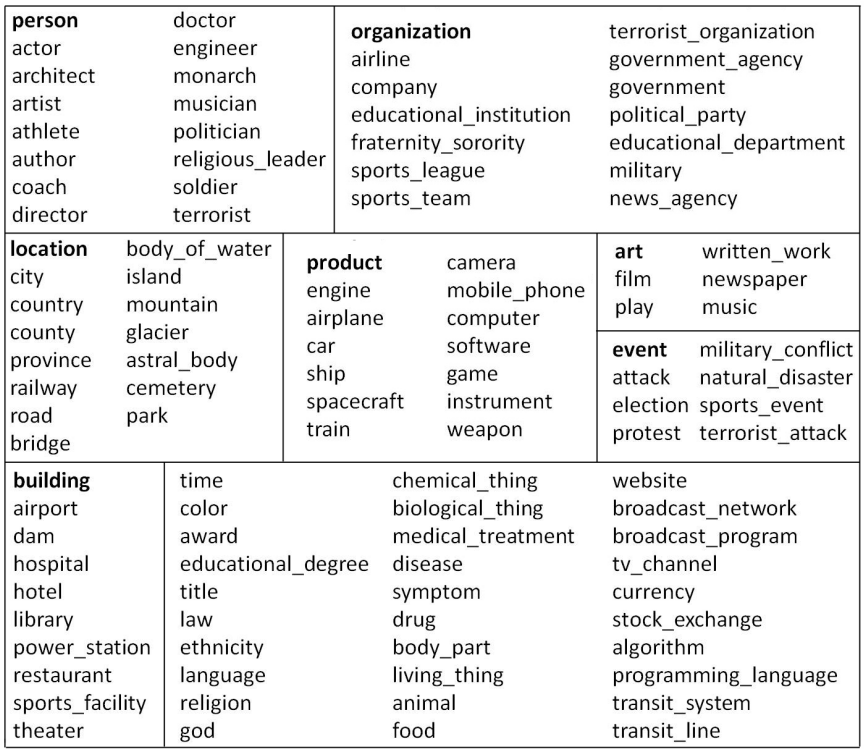

Figure 2.2: 112 tags used in FIGER. The bold-faced tag is a rough summary of each box. The box at the bottom right cornor contains mixed tags that are hard to be categorized.

NER systems [45] use a sequence model for the whole task, usually a linear-chain Conditional Random Field (CRF) [75]. In a sequence model, each token has a corresponding hidden variable indicating its type label. Consecutive tokens with the same type label are treated as one mention with its type. Here the state space of the hidden variables is linear to the size of the type set. However, if one segment is allowed to have multiple labels, the state space will grow exponentially. In practice, this is computationally infeasible when the tag set grows to more than a hundred tags. The pipeline approach avoids this problem and empirically it works reasonably well [26, 38, 116]. The models for segmentation and tagging are trained offline.

2.2.2 *Fine-Grained Tag Set*

The first step in entity tagging is defining the set of types. While there have been a few efforts at creating a comprehensive tag set [122], no consensus has been reached by the research community. On the other hand, a collabrative knowledge base, such as Freebase, provides thousands of types that are used to annotate each entry/entity in the website[1]. Compared to the type set in [122], the advantages of Freebase types are 1) broader coverage of entities in the world and 2) allowance of an entity bearing multiple overlapping types. For instance, *Washington, D.C.* could be annotated as both *City* and *Capital*, the latter of which is useful when it is used to represent the government of the United States.

While Freebase tags are comprehensive, they are also noisy (often created by non-expert users). As a result, we need to filter irrelevant types to reduce the data noise. We only keep well-maintained types (the ones with curated names, *e.g.*, `/location/city`) with more than 5 ground instances in Freebase. We further refine the types by manually merging too specific types, *e.g.*, `/olympics/olympic_games` and `/soccer/football_world_cup` are merged into *Sports_Event*. In the end, 112 types remain for use as our tag set, denoted as $\mathbf{T}$ (shown in Figure 2.2).

2.2.3 *Automatically Labeling Data*

To effectively learn the tagger, we need massive amount of labeled data. For this newly defined tag set, there does not exist such a set of labeled data. Previous researchers have hand-labeled each mention in a corpus with the entity types under consideration, but this process is so expensive that only a small training corpus is practical. Instead, we will use distant supervision, which is fully automatic and hence scalable [28]. Specifically, we utilize the information encoded in anchor links from Wikipedia text[2] in a manner similar to that of Nothman *et al.* [106]. For each linked segment m in a sentence, we find the corresponding Wikipedia entry e_m via its anchor link, get its types

[1]Wikipedia.com also annotates each article with a set of categories; however, the catogories are too noisy to be effectively used without further processing [104].

[2]We use the Wikipedia dump as of 20110513.

from Freebase and mapped the original ones into $t_m \subseteq \mathbf{T}$ using the our tag set. We removed the non-sentential sentences by heuristics, *e.g.*, thresholding the number of commas and semicolons in a sentence. We also remove the functional pages from Wikipedia, *e.g.* the `List` and `Category` pages. This process therefore automatically annotates sentences from Wikipedia using the tag set $\mathbf{T}$.

2.2.4 Segmentation

We use a linear-chain CRF model for segmentation[3] with three standard hidden states, *i.e.* "B", "I" and "O". These states indicate, respectively, a beginning token of a mention, a non-beginning token of a mention and a token not in a mention. A maximum sequence of consecutive tokens with "B" as the starting tag and, if any, "I" for the ones after that, is considered an entity mention / segment.

2.2.5 Multi-class Multi-label Classification

Then FIGER annotates each of the given mentions with a set of types $\hat{t} \subseteq \mathbf{T}$. This tagging problem is characterized in the literature as Multi-class Multi-label Classification [138]. We adapt a classic linear classifier, Perceptron [119] to our problem. A perceptron is in the form of

$$\hat{y} = \arg\max_{y} w^T \cdot f(x, y)$$

where $\hat{y}$ is a predicted label, $f(x, y)$ is the feature vector of a mention x with a label $y \in \mathbf{T}$ and w is the weight vector of the model. The weights are learned via additive updates

$$w \leftarrow w + \alpha(f(x, y) - f(x, \hat{y}))$$

where y is the true label and $\alpha > 0$ is a parameter controlling the learning pace.

We use all the tags $\hat{t}$ whose scores are larger than zero as the final prediction. And therefore one mention might have more than one predicted tags. We modify the update into

$$w \leftarrow w + \alpha(\sum_{y \in t} f(x, y) - \sum_{\hat{y} \in \hat{t}} f(x, \hat{y}))$$

[3]For segmentation, we only use the sentences with all named entities fully labeled.

where t is the set of true tags and $\hat{t}$ is the predicted set. Any spurious mispredictions (i.e. $\hat{t} - t$) will be discouraged. On the other hand, the weights for missed labels (*i.e.* $t - \hat{t}$) will be increased. While learning, the model is trained using gold mentions in the text and their tags.

Features: We include various kinds of features we found useful as shown in Table 2.1. The sentence "CJ Ottaway scored his celebrated 108 to seal victory for Eton ." and the segment "Eton" is used as a running example. We apply Stanford CoreNLP package [87] for syntactic analysis.

2.3 Experimentation

In this section, we wish to address the following questions:

- How accurately does the system perform on the task of fine-grained entity recognition?
- Are the fine grained entity tags useful for the down-stream applications?

2.3.1 Entity Recognition

First we compare FIGER against two state-of-the-art baselines on the Entity Recognition task.

Dataset: From the labeled dataset we generated as decribed in Section 2.2.3, we randomly sample 2 million sentences for training. We further collected 18 up-to-date news reports for testing from various sources, including a student newspaper at a university, a photograph magazine, etc., which covers mostly local news events where most entities do not frequently appear in media or not exist until very recently. We annotate the documents using the tag set T. One entity is allowed to have multiple tags. The annotation is made as complete and precise as possible. In total, 434 sentences are labeled with 562 entities and 771 tags.

Methodology: We use the F1 metric computed from the precision / recall scores in 3 different granualities. All the predictions with wrong segmentation are considered incorrect, which penalizes precision. All labeled entities missed by the system penalize recall. For a correctly segmented entity e, denote the true set of tags as t_e and the prediction set $\hat{t}_e$. The three ways of computing precision / recall are listed as follows:

- **Strict:** If and only if $t_e = \hat{t}_e$, the prediction will be regarded as correct and incorrect otherwise.

Feature	Decription	Example
Tokens	The tokens of the segment.	"Eton"
Word Shape	The word shape of the tokens in the segment.	"Aa" for "Eton" and "A0" for "CS446".
Part-of-Speech tags	The part-of-speech tags of the segment.	"NNP"
Length	The length of the segment.	1
Contextual unigrams	The tokens in a contextual window of the segment.	"victory", "for", "."
Contextual bigrams	The contextual bigrams including the segment.	"victory for", "for Eton" and "Eton ."
Brown clusters	The cluster id of each token in the segment (using the first 4, 8 and 12-bit prefixes).	"4_1110", "8_11100111", etc.
Head of the segment	The head of the segment following the rules by Collins [25].	"HEAD_Eton"
Dependency	The Stanford syntactic dependency [89] involving the head of the segment.	"prep_for:seal:dep"
ReVerb patterns	The frequent lexical patterns as meaningful predicates collected in ReVerb.	"seal_victory_for:dep"

Table 2.1: List of features used in entity tagging. Brown clusters [15, 94] build a partition of words grouped by distributional similarity, which is learned via a probabilistic model from unlabeled text. We used Liang [81]'s implementation to induce the word clusters. ReVerb [41] patterns are mostly multi-word expressions composed of a verb and a noun, with the noun carrying the semantic content of the whole expression. They are complementary to the dependency feature when a single verb is not as meaningful.

- **Loose Macro:** The precision and recall scores are computed for each entity. The overall precision and recall scores are the averages.

• **Loose Micro:** The overall precision is computed as $precision = \frac{\sum_e |t_e \cap \hat{t}_e|}{\sum_e |\hat{t}_e|}$ and the overall recall $recall = \frac{\sum_e |t_e \cap \hat{t}_e|}{\sum_e |t_e|}$.

Systems Compared: We compare against an adaptation from a Named-Entity Linking (NEL) system [113]. From the linked results, we look up their oracle types in Freebase, map them into our tag set and use the mapped results as predictions. Note that the mapping process is deterministic and guaranteed correct. Besides NEL, we also compare to Stanford NER [45] using CoNLL [137]'s 4 classes, i.e. *person*, *organization*, *location* and *miscellaneous*[4]. The training for the perceptron stops after the 20 iterations with the parameter $\alpha = 0.1$. These values are determined using a seperate validation set.

Results: As seen from Table 2.2, NEL is not able to identify most entities. The NEL baseline is quite capable in the various linking experiments [113]. Unfortunately, it has the critical disadvantage that its background knowledge base is incomplete and does not contain many of the entities. For example, people mentioned in everyday news events often are not prominent enough or have not been around long enough to have dedicated Wikipedia entries. One might argue that Wikipedia keeps growing and evolving but a fundamental question is how to conquer the tail of the entity distribution in the world. It is fair to say that Wikipedia will not have one page for each entity. In contrast, our system FIGER will still be able to extract the information of an entity by predicting its types, to a system or a human for comprehension.

Compared to Stanford NER with a coarser-grained tag set, FIGER successfully discovers more information for these entities. One might argue that the numbers are close. Part of the closeness in the results comes from how 217 of the 562 entities were single-labeled as person. For example, one report contained many students' names, and another contained many suspects' names.

We also show the results of FIGER given gold segmentation (the "FIGER (GOLD)" row). The 7%-10% deficit in performance by FIGER with predicted segmentation is partially due to the domain difference from Wikipedia text to newswire.

Another source of errors comes from the noise in the training data. For example, "United States"

[4]for evaluation, each of them is mapped to one of our tag set except *miscellaneous*.

Measure	Strict	Loose Macro	Loose Micro
NEL	0.220	0.327	0.381
Stanford (CoNLL)	0.425	0.585	0.548
FIGER	**0.471**	**0.617**	**0.597**
FIGER (GOLD)	0.532	0.699	0.693

Table 2.2: F1 scores for entity recognition.

in Freebase is annotated with tags including *language* and *cemetary*. However, not all the training sentences automatically labeled as in Section 2.2.3 support these types. In other words, there exist false positives in the training data due to distant supervision. We leave this issue to future work.

2.3.2 Relation Extraction

We now evaluate FIGER's ability to improve performance at the task of Relation Extraction (RE). We adopted a state-of-the-art RE system, MultiR [64] with a publically available implementation. It is trained using distant supervision by heuristically matching relation instances to text. For example, if $r(e_1, e_2) =$`ceoOf(Steve_Ballmer, Microsoft)` is a relation instance[5] and s is a sentence containing mentions of both e_1 and e_2, then s might be an expression of the ground tuple $r(e_1, e_2)$ and therefore can be easily used as a training example. Unfortunately, the heuristic often leads to noisy data and hence poor extraction quality. MultiR tackles this kind of supervision as a form of multi-instance learning, assuming that there is at least one sentence that naturally supports the fact that $r(e_1, e_2)$ holds[6].

Task: We aim at predicting if $r(e_1, e_2)$ holds given a set of relevant sentences. 36 unique relations,

[5]We assume binary relations in this experiment.

[6]For details, we refer interested readers to the original paper.

proposed by NELL [19], are used for testing[7]. Another relation *NA* is included when none of the 36 relations (shown in Table 2.3) holds for a pair of entities.

Dataset: We choose the NYT corpus [121], which has more than 1.8 million news articles from 1987 to 2007, as the textual repository for matching arguments. All entity pairs present in at least one sentence are considered as a candidate relation instance. The data ordered by date is split into 70% and 30% for training and testing. The features are computed following [96]. To get relation labels for these entity pairs, we collect ground tuples for all targeted relations. For each relation r, we start from a set of seed entity pairs which hold for r from NELL database[8]. The set is further enlarged by mapping a relation r_F in Freebase to r and adding all the entity pairs that hold for r_F. These tuples are then used as a gold answer set Δ. The training and test candidates are thus labeled by Δ. If there is no r such that $r(e_1, e_2)$ holds with respect to Δ, the entity pair (e_1, e_2) will then labeled as *NA*.

Methodology: We augment MultiR by allowing it to have FIGER's predictions on the types of the arguments. Binary indicators of FIGER's predicted tags for both arguments (224 in total) are appended to the feature vector of each sentence (whose length is in billions) used in MultiR. We compute precision and recall by comparing the collected set of relation instances in the test data, Δ_{test}, and the predicted relation instances by a relation extractor, Δ_{pred}. A true positive is granted if and only if $r(e_1, e_2)$ exists in both Δ_{test} and Δ_{pred}. The precision / recall curve is drawn by varying the confidence thredhold for each relation prediction. Note that the scores are underestimated in that it is likely that a predicted relation instance holds but does not exist in either NELL or Freebase database.

Results: Figure 2.3 depicts precision / recall curves for two systems, namely the orginal MultiR and the MultiR equipped with FIGER's predictions (MultiR+FIGER). As seen from the curve, FIGER's predictions give MultiR a significant improvement in performance. MultiR+FIGER extended the recall from 15.9% to 32.6% without losing precision. In general, MultiR+FIGER achieved the maximum F1 of 40.0% compared to 20.7% by the original MultiR, showing a 93% increase. Further

[7]We excluded the relations having inadequate ground tuples for training.

[8]`http://rtw.ml.cmu.edu/rtw/resources`

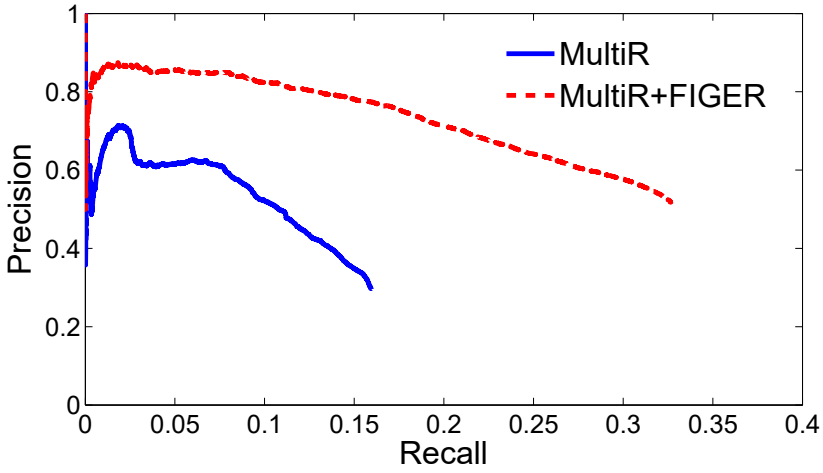

Figure 2.3: Precision / Recall curves for relation extraction.

investigate the precision / recall scores with respect to each relation. MultiR+FIGER has 22 wins, 7 ties and 7 losses against MultiR alone. Take the most improved relation "teamPlaysInLeague" for example (with a F1 increase from 3.6% to 73%). The type signature for this relation by traditional entity types is at best (*ORG*, *ORG*). This does not make distinction between two arguments. In contrast, FIGER provides (*Sports_team*, *Sports_league*), which obviously exposes key information for the relation extractor.

2.4 Related Work

In this section, we discuss the previous work on named entity recognition and methods for multi-class multi-label classification.

2.4.1 Named Entity Recognition (NER)

There has been considerable work on NER [26, 137, 45, 38, 112, 116], but they have several important limitations. Most NER systems only classify into three types: person, location and organization (or miscellaneous). A few systems (*e.g.*, Ontonotes [65]) use more, but still fail to

Relation types	
teamPlaysInLeague (+0.70)	**bookWriter** (+0.04)
stadiumLocatedInCity (+0.53)	**teamPlaysInCity** (+0.02)
coachesInLeague (+0.44)	**acquired** (+0.01)
leagueStadiums (+0.35)	**companyEconomicSector** (0.00)
cityLocatedInCountry (+0.30)	**visualArtistArtMovement** (0.00)
musicianInMusicArtist (+0.24)	**newspaperInCity** (0.00)
cityLocatedInState (+0.23)	**stateHasCapital** (0.00)
teamPlaysAgainstTeam (+0.20)	**musicArtistGenre** (0.00)
coachesTeam (+0.16)	**athletePlaysSport** (0.00)
athletePlaysInLeague (+0.15)	**athleteHomeStadium** (0.00)
athletePlaysForTeam (+0.12)	**musicianPlaysInstrument** (0.00)
televisionStationAffiliatedWith (+0.12)	hasOfficeInCountry (-0.01)
teamHomeStadium (+0.10)	competesWith (-0.03)
athleteCoach (+0.09)	televisionStationInCity (-0.05)
actorStarredInMovie (+0.06)	teammate (-0.05)
stateLocatedInCountry (+0.06)	ceoOf (-0.08)
radioStationInCity (+0.05)	currencyCountry (-0.10)
headquarteredIn (+0.05)	hasOfficeInCity (-0.14)

Table 2.3: The list of relation types used in the experiment. The number in the brackets following each relation type shows the absolute increase in F1 by MultiR+FIGER over MultiR alone. 29 relation types in bold face show non-negative improvement.

make enough distinctions to provide the most useful features for a relation extraction system.

In the past, there has also been some research has been focused on building a tagging system using a large number of fine-grained entity types. Fleischman and Hovy [46] classifies entities into 8 hierarchical person categories. Giuliano and Gliozzo [49] extends the number to 21 and presents the

People Ontology dataset, followed by Ekbal *et al.* [37]'s work of enriching the dataset by making use of appositional title-entity expressions. Identifying fine-grained person categories indeed is a challenging task but person names only occupy a small portion of all the entities in the world. Therefore it lacks the promise of providing essential information to other tasks, *e.g.*identifying relation instances where arguments are not human. Sekine [122] and Lee *et al.* [76] separately proposed a tag set of around 150 types but there is no implementation publically avaiable. [100] presented a semi-supervised NER system for 100 types. However, none of these work allows overlapping entity types. The exclusion largely reduces the coverage of information embedded in an entity.

The way of automatically generating labeled data is inspired by [106]. In contrast, their work is restricted in the traditional tag set while we exploit the Freebase type system and label the Wikipedia text in a much finer-grained tag set $\mathbf{T}$.

Named Entity Linking (NEL) is also closely related (also known as Disambiguation to Wikipedia [17, 30, 95, 43, 113]. Please refer to Section 3.6 for a thorough review.). NEL is useful for frequent and well-known named entities. It can be seen as extremely fine-grained entity recognition. The downside is that, as shown in our experiment, it is insufficient when entities do not exist in the background database (*e.g.*, Wikipedia).

2.4.2 Multi-class Multi-label Classification

In FIGER, we solved a multi-class multi-label classification problem for tagging the entities. A comprehensive survey on this topic has been written in [138]. We used a simple adaption from the Perceptron model mainly in consideration of speed. With the growing number of labels, a more sophisticated model, *e.g.*, [146] might achieve a higher accuracy but is highly likely to suffer from unaffordable computational cost.

2.5 Discussion

In this work, we leverage Wikipedia anchor links to heuristically obtain a large number of labeled entity mentions. However, we gloss over a serious issue: all the types that belong to an entity are assigned to every mention of the entity, whether the context indicates every type or not. For instance, John is both a *person* and a *politician*; in a sentence like "John just had breakfast", labeling the mention of "John" with both types will cause the classifier to aggressively predict types even without a sufficient context. There has been some effort in reducing such noise [47]. The similar issue also occurs in distant supervision for relation extraction [120].

In addition to improving the quality of the training data, we would also like to design a model more suitable for this multi-label problem. Currently, FIGER consists of independent classifiers for each entity type. Therefore, it is possible, although unlikely, to see a type prediction with both *person* and *location*. A possible solution is to model the correlation between labels and incorporate such information in both training and test time. The correlation between two types t_1 and t_2 can be estimated by the number of entities that share both types. If that number is close to the number of entities with t_1, then t_2 implies t_1 (*e.g.*, all *politician*s are *person*s). It the number is zero, then two types are exclusive. The next step is to build a graph reflecting such correlations and run efficient inference for multi-label classification [33].

Recent work on fine-grained entity types [130, 101] uses a large type hierarchy from YAGO2 [62], derived from WordNet synsets [42]. As the granuality of the type set increases, the entity mentions belonging to the same (leaf) type are getting semantically similar to each other. In the extreme case, all the entity mentions for a type refers to the same entity. This introduces another important task, Entity Linking, which we will discuss in the next chapter.

2.6 Summary

In this chapter, we introduce a large set of entity types, derived from Freebase, which are expectedly useful both to human understanding and other NLP applications. We describe FIGER, a fine-grained entity recognizer, which identifies references to entities in natural language text and labels them

with appropriate tags. We compare FIGER with two state-of-the-art baselines, showing that 1) FIGER has excellent overall accuracy and dominates other approaches for uncommon entities, and 2) when used as features, our fine-grained tags can significantly improve the performance of relation extraction by 93% in F1.

Chapter 3

DESIGN CHALLENGES IN ENTITY LINKING

3.1 Introduction

In the previous chapter, we consider the task of classifying entity mentions into fine-grained entity types. Now we move forward from detecting class membership to linking to entity identities. Entity Linking is a central task in information extraction — given a textual passage, identify entity *mentions* (substrings corresponding to world entities) and link them to the corresponding entry in a given Knowledge Base. A natural choice of the knowledge base is Wikipedia [17, 30, 95]. Other choices are Freebase [124], DBpedia [93], Yago [63], etc. The links not only provide semantic annotations to human readers but also a machine-consumable representation of the most basic semantic knowledge in the text. Many other NLP applications can benefit from such links, such as distantly-supervised relation extraction [28, 114, 64, 71] that uses EL to create training data, and some coreference systems that use EL for disambiguation [55, 148, 36]. Unfortunately, in spite of numerous papers on the topic and several published datasets, there is surprisingly little understanding about state-of-the-art performance.

We argue that there are three reasons for this confusion. First, *there is no standard definition of the problem.* A few variants have been studied in the literature, such as Wikification [95, 113, 21] which aims at linking noun phrases to Wikipedia entities and Named Entity Linking (aka Named Entity Disambiguation) [92, 63] which targets only named entities. Here we use the term *Entity Linking* as a unified name for both problems, and *Named Entity Linking* (NEL) for the subproblem of linking only named entities. But names are just one part of the problem. For many variants there are no annotation guidelines for scoring links. What types of entities are valid targets? When multiple entities are plausible for annotating a mention, which one should be chosen? Are nested mentions allowed? Without agreement on these issues, a fair comparison is elusive.

Secondly, *it is almost impossible to assess approaches, because systems are rarely compared using the same datasets.* For instance, Hoffart *et al.* [63] developed a new dataset (AIDA) based on the CoNLL 2003 Named Entity Recognition dataset but failed to evaluate their system on MSNBC previously created by [30]; Wikifier [21] compared to the authors' previous system [113] using the originally selected datasets but didn't evaluate using AIDA data.

Finally, when two end-to-end systems are compared, *it is rarely clear which aspect of a system makes one better than the other.* This is especially problematic when authors introduce complex mechanisms or nondeterministic methods that involve learning-based reranking or joint inference.

In this chapter, we analyze several significant inconsistencies among the datasets. To have a better understanding of the importance of various techniques, we develop a simple and modular, unsupervised EL system, VINCULUM. We compare VINCULUM to the two leading sophisticated EL systems on a comprehensive set of nine datasets. While our system does not consistently outperform the best EL system, it does come remarkably close and serves as a simple and competitive baseline for future research. Furthermore, we carry out an extensive ablation analysis, whose results illustrate 1) even a near-trivial model using CrossWikis [132] performs surprisingly well, and 2) incorporating a fine-grained set of entity types raises that level even higher.

3.2 No Standard Benchmark

In this section, we describe some of the key differences amongst evaluations reported in existing literature, and propose a candidate benchmark for EL.

3.2.1 Datasets

Nine data sets are in common use for EL evaluation; we partition them into three groups. The UIUC group (ACE and MSNBC datasets) [113], AIDA group (with dev and test sets) [63], and TAC-KBP group (with datasets ranging from the 2009 through 2012 competitions) [92]. Their statistics are summarized in Table 3.1.

Group	Data Set	# of Mentions	Entity Types	KB	# of NILs	Eval. Metric
UIUC	ACE	244	Any Wikipedia Topic	Wikipedia	0	BOC F1
	MSNBC	654	Any Wikipedia Topic	Wikipedia	0	BOC F1
AIDA	AIDA-dev	5917	PER,ORG,LOC,MISC	Yago	1126	Accuracy
	AIDA-test	5616	PER,ORG,LOC,MISC	Yago	1131	Accuracy
TAC KBP	TAC09	3904	PER^T,ORG^T,GPE	TAC $\subset$ Wiki	2229	Accuracy
	TAC10	2250	PER^T,ORG^T,GPE	TAC $\subset$ Wiki	1230	Accuracy
	TAC10T	1500	PER^T,ORG^T,GPE	TAC $\subset$ Wiki	426	Accuracy
	TAC11	2250	PER^T,ORG^T,GPE	TAC $\subset$ Wiki	1126	B^3+ F1
	TAC12	2226	PER^T,ORG^T,GPE	TAC $\subset$ Wiki	1049	B^3+ F1

Table 3.1: Characteristics of the nine NEL data sets. Entity types: The AIDA data sets include named entities in four NER classes, Person (PER), Organization (ORG), Location (LOC) and Misc. In TAC KBP data sets, both Person (PER^T) and Organization entities (ORG^T) are defined differently from their NER counterparts and geo-political entities (GPE), different from LOC, exclude places like `KB:Central California`. KB (Sec. 3.2.2): The knowledge base used when each data was being developed. Evaluation Metric (Sec. 3.2.3): Bag-of-Concept F1 is used as the evaluation metric in [113, 21]. B^3+ F1 used in TAC KBP measures the accuracy in terms of entity clusters, grouped by the mentions linked to the same entity.

UIUC: ACE, a newswire subset of the ACE coreference data set [97], was introduced in [113]. The first nominal mention of each gold coreference chain is annotated by Amazon's Mechanical Turk workers. MSNBC was developed by [30], which consists of 20 MSNBC news articles on different topics.

AIDA: Based on CoNLL 2003 Named Entity Recognition data, Hoffart *et al.* [63] hand-annotated all these proper nouns with corresponding entities in YAGO2. Both the dev set (AIDA-dev) and the test set (AIDA-test) are included in the benchmark.

Our set of nine is not exhaustive, but most other datasets, *e.g.* CSAW [73] and AQUAINT [95], annotate common concepts in addition to named entities. As we argue in Sec. 3.3.1, it is extremely difficult to define annotation guidelines for common concepts, and therefore they aren't suitable for evaluation. For clarity, this work focuses on linking named entities. Similarly, we exclude

datasets comprising Tweets and other short-length documents, since radically different techniques are needed for the specialized corpora.

Table 3.2 presents a list of recent EL publications showing the data sets that they use for evaluation. The sparsity of this table is striking — apparently no system has reported the performance data from all three of the major evaluation groups.

UIUC: ACE, a newswire subset of the ACE coreference data set [97], was introduced in [113]. The first nominal mention of each gold coreference chain is annotated by Amazon's Mechanical Turk workers. MSNBC was developed by [30], which consists of 20 MSNBC news articles on different topics.

AIDA: Based on CoNLL 2003 Named Entity Recognition data, Hoffart *et al.* [63] hand-annotated all these proper nouns with corresponding entities in YAGO2. Both the dev set (AIDA-dev) and the test set (AIDA-test) are included in the benchmark.

TAC-KBP: From the annual TAC-KBP competitions[1], the evaluation sets from 2009 to 2012 are included (as well as a training set from 2010, TAC10T). Each data set consists of a series of linking queries for named entities. A query provides the surface form of the mention and the source document id. The source documents mainly come from newswire and web documents.

3.2.2 Knowledge Base

Existing benchmarks have also varied considerably in the knowledge base used for link targets. Wikipedia has been most commonly used [95, 113, 21], however datasets were annotated using different snapshots and subsets. Other KBs include Yago [63], Freebase [125], DBpedia [93] and a subset of Wikipedia [91]. Given that almost all KBs are descendants of Wikipedia, we use Wikipedia as the base KB in this work.[2]

NIL entities: In spite of Wikipedia's size, there are many real-world entities that are absent from the KB. When such a target is missing for a mention, it is said to link to a *NIL entity* [92] (aka out-of-KB or unlinkable entity [61]). In the TAC KBP, in addition to determining if a mention

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

[2] Since the knowledge bases for all the data sets were around 2011, we use Wikipedia dump 20110513.

Data Set	ACE	MSNBC	AIDA-test	TAC09	TAC10	TAC11	TAC12	AQUAINT	CSAW
Cucerzan [30]		x							
Milne and Witten [95]								x	
Kulkarni et al. [73]		x							x
Ratinov et al. [113]	x	x						x	
Hoffart et al. [63]			x						
Han and Sun [56]				x					x
He et al. [58]			x		x				
He et al. [59]	x	x						x	
Cheng and Roth [21]	x	x				x		x	
Sil and Yates [125]	x	x	x						
Li et al. [80]			x	x					
Cornolti et al. [27]		x	x						x
TAC-KBP participants				x	x	x	x		

Table 3.2: A sample of papers on entity linking with the data sets used in each paper (ordered chronologically). TAC-KBP proceedings comprise additional papers [92, 68, 68, 91]. Our intention is not to exhaust related work but to illustrate how sparse evaluation impedes comparison.

has no entity in the KB to link, all the mentions that represent the same real world entities must be clustered together. Since our focus is not to create new entities for the KB, NIL clustering is beyond the scope of this work. We only evaluate whether a mention with no suitable entity in the KB is predicted as NIL. The AIDA data sets similarly contain such NIL annotations whereas ACE and MSNBC omit these mentions altogether.

3.2.3 Evaluation Metrics

While a variety of metrics have been used for evaluation, there is little agreement on which one to use. However, this detail is quite important, since the choice of metric strongly biases the results. We describe the most common metrics below.

Bag-of-Concept F1 (ACE, MSNBC): For each document, a gold bag of Wikipedia entities is evaluated against a bag of system output entities requiring exact segmentation match. This metric may have its historical reason for comparison but is in fact flawed since it will obtain 100% F1 for an annotation in which every mention is linked to the wrong entity, but the bag of entities is the same as the gold bag.

Micro Accuracy (TAC09, TAC10, TAC10T): For a list of given mentions, the metric simply measures the percentage of correctly predicted links.

TAC-KBP B^3+ F1 (TAC11, TAC12): The mentions that are predicted as NIL entities are required to be clustered according to their identities (NIL clustering). The overall data set is evaluated using a entity cluster-based B^3+ F1.

NER-style F1 (AIDA): Similar to official CoNLL NER F1 evaluation, a link is considered correct only if the mention matches the gold boundary *and* the linked entity is also correct. A wrong link with the correct boundary penalizes both precision and recall.

We note that Bag-of-Concept F1 is equivalent to the measure for Concept-to-Wikipedia task proposed in [27] and NER-style F1 is the same as strong annotation match. In the experiments, we use the corresponding dataset-specific measures for comparison.

3.3 No Annotation Guidelines

Not only do we lack a common data set for evaluation, but most prior researchers fail to even *define* the problem under study, before developing algorithms. Often an overly general statement such as annotating the mentions to "referent Wikipedia pages" or "corresponding entities" is used to describe which entity link is appropriate. This section shows that failure to have a detailed annotation guideline causes a number of key inconsistencies between data sets. A few assumptions are subtly made in different papers, which makes direct comparisons unfair and hard to comprehend.

3.3.1 Entity Mentions: Common or Named?

Which entities deserve links? Some argue for restricting to *named* entities. Others argue that any phrase that *can* be linked to a Wikipedia entity adds value. Without a clear answer to this issue, any data set created will be problematic. It's not fair to penalize a NEL system for skipping a common noun phrases; nor would it be fair to lower the precision of a system that "incorrectly" links a common concept. However, we note that including mentions of common concepts is actually quite problematic, since the choice is highly subjective.

Example 1 In December 2008, Hoke was hired as the head football coach at San Diego State University. (Wikipedia)

At first glance, `KB:American football` seems the gold-standard link. However, there is another entity `KB:College football`, which is clearly also, if not more, appropriate. If one argues that `KB:College football` should be the right choice given the context, what if `KB:College football` does not exist in the KB? Should NIL be returned in this case? The question is unanswered.[3]

For the rest of this chapter, we focus on the (better defined) problem of solely linking named entities.[4] AQUAINT and CSAW are therefore not used for evaluation due to an disproportionate number of common concept annotations.

3.3.2 How Specific Should Linked Entities Be?

It is important to resolve disagreement when more than one annotation is plausible. The TAC-KBP annotation guidelines [1] specify that different iterations of the same organization (e.g. the `KB:111th U.S. Congress` and the `KB:112th U.S. Congress`) should not be considered as distinct entities. Unfortunately, this is not a common standard shared across the data sets, where often the most specific possible entity is preferred.

---

[3] Note that linking common noun phrases is closely related to Word Sense Disambiguation [99].

[4] We define *named entity mention* extensionally: any name uniquely referring to one entity of a predefined class, e.g. a specific person or location.

Example 2 Adams and Platt are both injured and will miss England's opening World Cup qualifier against Moldova on Sunday. (AIDA)

Here the mention "World Cup" is labeled as `KB:1998 FIFA World Cup`, a specific occurrence of the event `KB:FIFA World Cup`.

It is indeed difficult to decide how specific the gold link should be. Given a static knowledge base, which is often incomplete, one cannot always find the most specific entity. For instance, there is no Wikipedia page for the `KB:116th U.S. Congress` because the Congress has not been elected yet. On the other hand, using general concepts can cause troubles for machine reading. Consider *president-of* relation extraction on the following sentence.

Example 3 Joe Biden is the Senate President in the 113th United States Congress.

Failure to distinguish different Congress iterations would cause an information extraction system to falsely extracting the fact that `KB:Joe Biden` is the Senate President of the `KB:United States Congress` at all times!

3.3.3 Metonymy

Another situation in which more than one annotation is plausible is metonymy, which is a way of referring to an entity not by its own name but rather a name of some other entity it is associated with. A common example is to refer to a country's government using its capital city.

Example 4 Moscow's as yet undisclosed proposals on Chechnya's political future have , meanwhile, been sent back to do the rounds of various government departments. (AIDA)

The mention here, "Moscow", is labeled as `KB:Government of Russia` in AIDA. If this sentence were annotated in TAC-KBP, it would have been labeled as `KB:Moscow` (the city) instead. Even the country `KB:Russia` seems to be a valid label. However, neither the city nor the country can actually *make a proposal*. The real entity in play is `KB:Government of Russia`.

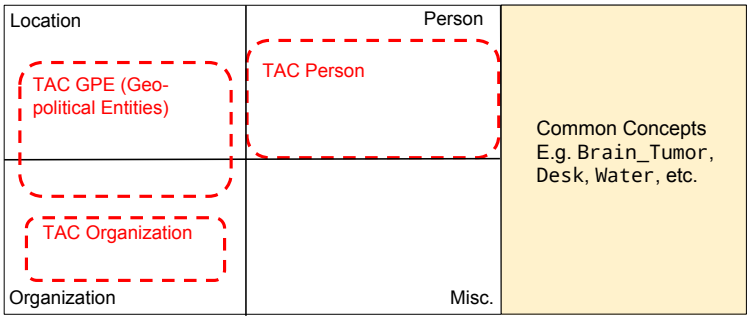

Figure 3.1: Entities divided by their types. For named entities, the solid squares represent 4 CoNLL(AIDA) classes; the red dashed squares display 3 TAC classes; the shaded rectangle depicts common concepts.

3.3.4 Named Entities, But of What Types?

Even in the data sets consisting of solely named entities, the types of the entities vary and therefore the data distribution differs. TAC-KBP has a clear definition of what types of entities require links, namely Person, Organization and Geo-political entities. AIDA, which adopted the NER data set from the CoNLL shared task, includes entities from 4 classes, Person, Organization, Location and Misc.[5] Compared to the AIDA entity types, it is obvious that TAC-KBP is more restrictive, since it does not have Misc. entities (e.g. `KB:FIFA World Cup`). Moreover, TAC entities don't include fictional characters or organizations, such as `KB:Sherlock Holmes`. TAC GPEs include some geographical regions, such as `KB:France`, but exclude those without governments, such as `KB:Central California` or locations such as `KB:Murrayfield Stadium`.[6] Figure 3.1 summarizes the substantial differences between the two type sets.

3.3.5 Can Mention Boundaries Overlap?

We often see one entity mention nested in another. For instance, a U.S. city is often followed by its state, such as "Portland, Oregon". One can split the whole mention to individual ones,

[5] `http://www.cnts.ua.ac.be/conll2003/ner/annotation.txt`

[6] `http://nlp.cs.rpi.edu/kbp/2014/elquery.pdf`

"Portland" for the city and "Oregon" for the city's state. AIDA adopts this segmentation. However, annotations in an early TAC-KBP dataset (2009) select the whole span as the mention. We argue that all three mentions make sense. In fact, knowing the structure of the mention would facilitate the disambiguation (i.e. the state name provides enough context to uniquely identify the city entity). Besides the mention segmentation, the links for the nested entities may also be ambiguous.

Example 5 Dorothy Byrne, a state coordinator for the Florida Green Party, said she had been inundated with angry phone calls and e-mails from Democrats, but has yet to receive one regretful note from a Nader voter.

The gold annotation from ACE is `KB:Green Party of Florida` even though the mention doesn't contain "Florida" and can arguably be linked to `KB:US Green Party`.

3.4 A Simple & Modular Linking Method

In this section, we present VINCULUM, a simple, unsupervised EL system that performs comparably to the state of the art. As input, VINCULUM takes a plain-text document d and outputs a set of segmented mentions with their associated entities $A_d = \{(m_i, l_i)\}$. VINCULUM begins with mention extraction. For each identified mention m, candidate entities $C_m = \{c_j\}$ are generated for linking. VINCULUM assigns each candidate a linking score $s(c_j|m, d)$ based on the entity type compatibility, its coreference mentions, and other entity links around this mention. The candidate entity with the maximum score, *i.e.* $l = \arg\max_{c \in C_m} s(c|m, d)$, is picked as the predicted link of m.

Figure 3.2 illustrates the linking pipeline that follows mention extraction. For each mention, VINCULUM ranks the candidates at each stage based on an ever widening context. For example, candidate generation (Section 3.4.2) merely uses the mention string, entity typing (Section 3.4.3) uses the sentence, while coreference (Section 3.4.4) and coherence (Section 3.4.5) use the full document and Web respectively. Our pipeline mimics the sieve structure introduced in [77], but instead of merging coreference clusters, we adjust the probability of candidate entities at each stage. The modularity of VINCULUM enables us to study the relative impact of its subcomponents.

Subroutines: A mention extractor E, a candidate generator D, an entity type predictor TP, a coreference system R and a coherence function ϕ.

Input: Document d.

Output: Entity Link Annotations $\{(m, l_m)\}$

1) Extract mentions $M = E(d)$. 2) Run coreference resolution $R(d)$ and obtain coreference clusters of mentions. Denote the cluster containing a mention m as $r(m)$ and the representative mention of a cluster r as $rep(r)$.

for $m \in M$ **do**

  if $m = rep(r(m))$ **then**

    Generate candidates $C_m = D(m)$ (Sec. 3.4.2) ;

    use TP to predict the entity types (Sec. 3.4.3);

    for $c \in C_m$ **do**

      Compute the prob. of each candidate $p(c|m, s_m)$ based on the predicted types.

  else

    use the representative mention $rep(r(m))$ for linking (Sec. 3.4.4).

  Set $p_m = \arg\max_{c \in C_m} p(c|m, s_m)$;

Let $P_d = \cup_{m_i \in M} \{p_{m_i}\}$ (Sec. 3.4.5) ;

for $m \in M$ **do**

  for $c \in C_m$ **do**

    Compute $p_\phi(c|P_d)$ using the given coherence function ϕ and the final score

    $s(c|m, d) = p(c|m, s_m) + p_\phi(c|P_d)$;

  Set the final link $l_m = \arg\max_{c \in C_m} s(c|m, d)$

return $\{(m, l_m) : m \in M\}$

Algorithm 1: VINCULUM Algorithm.

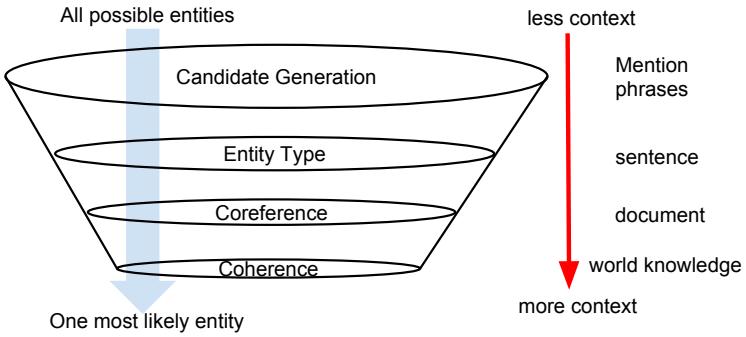

Figure 3.2: The process of finding the best entity for a mention. All possible entities are sifted through as VINCULUM proceeds at each stage with a widening range of context in consideration. Details of the approach are presented in Algorithm 1.

3.4.1 Mention Extraction

The first step of EL extracts potential mentions from the document. Since VINCULUM restricts attention to named entities, we use a Named Entity Recognition (NER) system [45]. Alternatively, an NP chunker may be used to identify the mentions.

3.4.2 Dictionary-based Candidate Generation

While in theory a mention could link to any entity in the KB, in practice one sacrifices little by restricting attention to a subset (dozens) precompiled using a dictionary. A common way to build such a dictionary D is by crawling Web pages and aggregating anchor links that point to Wikipedia pages. The frequency with which a mention (anchor text), m, links to a particular entity (anchor link), c, allows one to estimate the conditional probability $p(c|m)$. We adopt the CrossWikis dictionary, which was computed from a Google crawl of the Web [132]. The dictionary contains more than 175 million unique strings with the entities they may represent. In the literature, the dictionary is often built from the anchor links within the Wikipedia website (e.g., [113, 63]).

In addition, we employ two small but precise dictionaries for U.S. state abbreviations and demonyms when the mention satisfies certain conditions. For U.S. state abbreviations, a comma before the mention is required. For demonyms, we ensure that the mention is either an adjective or

a plural noun.

3.4.3 Incorporating Entity Types

For an ambiguous mention such as "Washington", knowing that the mention denotes a person allows an EL system to promote `KB:George Washington` while lowering the rank of the capital city in the candidate list. We incorporate this intuition by combining it probabilistically with the CrossWikis prior.

$$p(c|m,s) = \sum_{t \in T} p(c,t|m,s) = \sum_{t \in T} p(c|m,t,s)p(t|m,s) \ ,$$

where s denotes the sentence containing this mention m and T represents the set of all possible types. We assume the candidate c and the sentential context s are conditionally independent if both the mention m and its type t are given. In other words, $p(c|m,t,s) = p(c|m,t)$, the RHS of which can be estimated by renormalizing $p(c|m)$ w.r.t. type t:

$$p(c|m,t) = \frac{p(c|m)}{\sum_{c \mapsto t} p(c|m)} \ ,$$

where $c \mapsto t$ indicates that t is one of c's entity types.[7] The other part of the equation, $p(t|m,s)$, can be estimated by any off-the-shelf Named Entity Recognition system, *e.g.* [45] and [86].

3.4.4 Coreference

It is common for entities to be mentioned more than once in a document. Since some mentions are less ambiguous than others, it makes sense to use the most representative mention for linking. To this end, VINCULUM applies a coreference resolution system (*e.g.* [77]) to cluster coreferent mentions. The representative mention of a cluster is chosen for linking.Note that the representative mention in coreference resolution is not always the best mention for linking. When the representative mention contains a relative clause, we use the submention without the clause, which is favorable

[7] We notice that an entity often has multiple appropriate types, e.g. a school can be either an organization or a location depending on the context. We use Freebase to provide the entity types and map them appropriately to the target type set.

for candidate generation. When the representative mention represents a location, a longer, non-conjunctive mention is preferred. We also apply some heuristic to find organization acronyms, etc. While there are more sophisticated ways to integrate EL and coreference [55], VINCULUM's pipeline is simple and modular.

3.4.5 Coherence

When `KB:Barack Obama` appears in a document, it is more likely that the mention "Washington" represents the capital `KB:Washington, D.C.` as the two entities are semantically related, and hence the joint assignment is *coherent*. A number of researchers found inclusion of some version of coherence is beneficial for EL [30, 95, 113, 63, 21]. For incorporating it in VINCULUM, we seek a document-wise assignment of entity links that maximizes the sum of the coherence scores between each pair of entity links predicted in the document d,

$$\sum_{1 \le i < j \le |M_d|} \phi(l_{m_i}, l_{m_j}) \,,$$

where ϕ is a function that measures the coherence between two entities, M_d denotes the set of all the mentions detected in d and l_{m_i} (l_{m_j}) is one of the candidates of $m_i (m_j)$. Instead of searching for the exact solution in a brute-force manner $O(|C|^{|M_d|})$ where $|C| = \max_{m \in M_d} |C_m|$, we isolate each mention and greedily look for the best candidate by fixing the predictions of other mentions, allowing linear time search $O(|C| \cdot |M_d|)$.

Specifically, for a mention m and each of its candidates, we compute a score,

$$coh(c) = \frac{1}{|P_d| - 1} \sum_{p \in P_d \setminus \{p_m\}} \phi(p, c), c \in C_m \,,$$

where P_d is the union of all intermediate links $\{p_m\}$ in the document. Since both measures take values between 0 and 1, we denote the coherence score $coh(c)$ as $p_\phi(c|P_d)$, the conditional probability of an entity given other entities in the document. The final score of a candidate is the sum of coherence $p_\phi(c|P_d)$ and type compatibility $p(c|m, s)$.

Two coherence measures have been found to be useful: Normalized Google Distance (NGD) [95, 113] and relational score [21]. NGD between two entities c_i and c_j is defined based on the link

structure between Wikipedia articles as follows:

$$\phi_{NGD}(c_i, c_j) = 1 - \frac{\log(\max(|L_i|, |L_i|)) - \log(|L_i \cap L_j|)}{\log(W) - \log(\min(|L_i|, |L_i|))}$$

where L_i and L_j are the incoming (or outgoing) links in the Wikipedia articles for c_i and c_j respectively and W is the total number of entities in Wikipedia. The relational score between two entities is a binary indicator whether a relation exists between them. We use Freebase [8] as the source of the relation triples $F = \{(sub, rel, obj)\}$. Relational coherence ϕ_{REL} is thus defined as

$$\phi_{REL}(e_i, e_j) = \begin{cases} 1 & \exists r, (e_i, r, e_j) \text{ or } (e_j, r, e_i) \in F \\ 0 & \text{otherwise.} \end{cases}$$

3.5 Experiments

In this section, we present experiments to address the following questions:

- Is NER sufficient to identify mentions? (Sec. 3.5.1)
- How much does candidate generation affect final EL performance? (Sec. 3.5.2)
- How much does entity type prediction help EL? What type set is most appropriate? (Sec. 3.5.3)
- How much does coherence improve the EL results? (Sec. 3.5.4)
- How well does VINCULUM perform compared to the state-of-the-art? (Sec. 3.5.5)
- Finally, which of VINCULUM's components contribute the most to its performance? (Sec. 3.5.6)

3.5.1 Mention Extraction

We start by using Stanford NER for mention extraction and measure its efficacy by the recall of correct mentions shown in Table 3.3. TAC data sets are not included because the mention strings are given in that competition. The results indicate that at least 10% of the gold-standard mentions are left out when NER, alone, is used to detect mentions. Some of the missing mentions are noun

[8] The mapping between Freebase and Wikipedia is provided at `https://developers.google.com/freebase`.

	ACE		MSNBC		AIDA-dev		AIDA-test	
	R	P	R	P	R	P	R	P
NER	89.7	10.9	77.7	65.5	89.0	75.6	87.1	74.0
+NP	96.0	2.4	90.2	12.4	94.7	21.2	92.2	21.8
+DP	96.8	1.8	90.8	9.3	95.8	14.0	93.8	13.5
+NP+DP	98.0	1.2	92.0	5.8	95.9	9.4	94.1	9.4

Table 3.3: Performance(%, R: Recall; P: Precision) of the correct mentions using different **mention extraction** strategies. ACE and MSNBC only annotate a subset of all the mentions and therefore the absolute values of precision are less useful than the relative order between methods.

Approach	TAC09	TAC10	TAC10T	TAC11	TAC12	AIDA-dev	AIDA-test	ACE	MSNBC
CrossWikis only	80.4	85.6	86.9	78.5	62.4	62.6	60.4	87.6	82.6
+NER	79.2	83.3	85.1	76.6	61.1	66.4	**66.2**	76.8	77.9
+FIGER	**81.0**	**86.1**	**86.9**	**78.8**	**63.5**	**66.7**	64.6	**87.8**	**83.6**
+NER(GOLD)	**85.7**	87.4	88.0	80.1	**66.7**	72.6	72.0	89.2	87.1
+FIGER(GOLD)	84.1	**88.8**	**89.0**	**81.6**	66.1	**76.2**	**76.5**	**91.7**	**89.5**

Table 3.4: Performance (%) after **incorporating entity types**, comparing two sets of entity types (NER and FIGER). Using a set of fine-grained entity types (FIGER) generally achieves better results.

phrases without capitalization, a well-known limitation of automated extractors. To recover them, we experiment with an NP chunker (NP) [9] and a deterministic noun phrase extractor based on parse trees (DP). Although we expect them to introduce spurious mentions, the purpose is to estimate an upper bound for mention recall. The results confirm the intuition: both methods improve recall, but the effect on precision is prohibitive. Therefore, we only use NER in subsequent experiments. Note that the recall of mention extraction is an upper bound of the recall of end-to-end predictions.

[9]OpenNLP NP Chunker: `opennlp.apache.org`

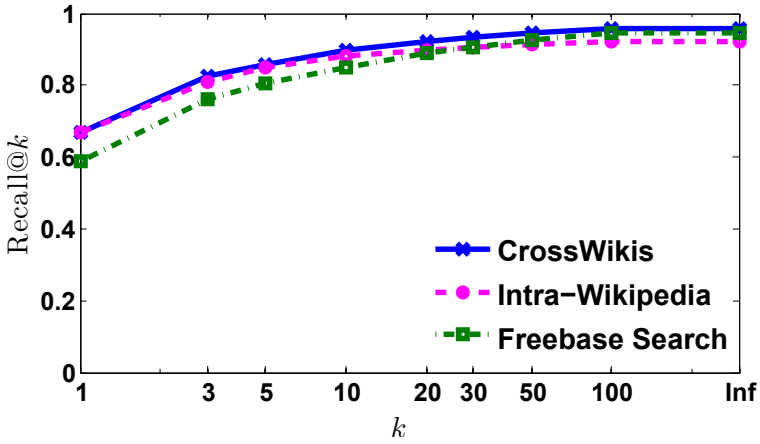

Figure 3.3: Recall@k on an aggregate of nine data sets, comparing three **candidate generation** methods.

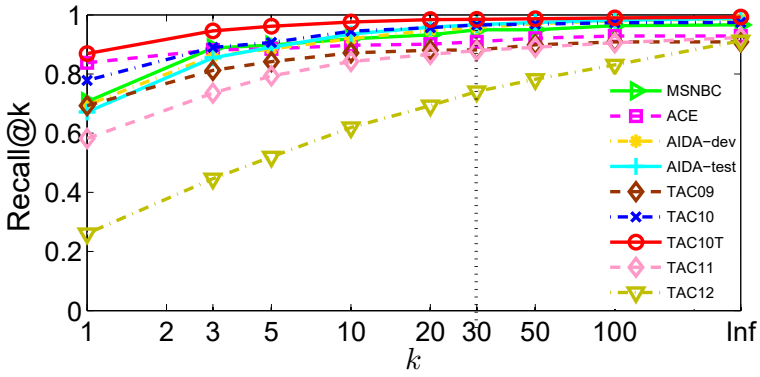

Figure 3.4: Recall@k using CrossWikis for candidate generation, split by data set. 30 is chosen to be the cut-off value in consideration of both efficiency and accuracy.

3.5.2 Candidate Generation

In this section, we inspect the performance of candidate generation. We compare CrossWikis with an intra-Wikipedia dictionary adopted from AIDA [10] and Freebase Search API [11]. Each candidate generation component takes a mention string as input and returns an ordered list of candidate entities representing the mention. The candidates produced by Crosswikis and the intra-Wikipedia dictionary

[10] `https://github.com/yago-naga/aida`

[11] `https://www.googleapis.com/freebase/v1/search`, restricted to no more than 220 candidates per query.

are ordered by their conditional probabilities given the mention string. Freebase API provides scores for the entities using a combination of text similarity and an in-house entity relevance score. We compute candidates for the union of all the non-NIL mentions from all 9 data sets and measure their efficacy by recall@k. From Figure 3.3, it is clear that CrossWikis outperforms both the intra-Wikipedia dictionary and Freebase Search API for almost all k. The intra-Wikipedia dictionary is on a par with CrossWikis at $k = 1$ but in general has a lower coverage of the gold candidates compared to CrossWikis [12]. Freebase API offers a better coverage than the intra-Wikipedia dictionary but is less efficient than CrossWikis. In other words, Freebase API needs a larger cut-off value to include the gold entity in the candidate set.

Using CrossWikis for candidate generation, we plot the recall@k curves per data set (Figure 3.4). To our surprise, in most data sets, CrossWikis alone can achieve more than 70% recall@1. The only exceptions are TAC11 and TAC12 because the organizers intentionally selected the mentions that are highly ambiguous such as "ABC" and/or incomplete such as "Brown". For efficiency, we set a cut-off threshold at 30 ($> 80\%$ recall for all but one data set). Note that Crosswikis itself can be used a context-insensitive EL system by looking up the mention string and predicting the entity with the highest conditional probability. The second row in Table 3.4 presents the results using this simple baseline. Crosswikis alone, using only the mention string, has a fairly reasonable performance.

3.5.3 Incorporating Entity Types

Here we investigate the impact of the entity types on the linking performance. The most obvious choice is the traditional NER types ($T_{\text{NER}} = \{\text{PER}, \text{ORG}, \text{LOC}, \text{MISC}\}$). To predict the types of the mentions, we run Stanford NER [45] and set the predicted type t_m of each mention m to have probability 1 (*i.e.* $p(t_m|m, s) = 1$). As to the types of the entities, we map their Freebase types to

[12] We also compared to another intra-Wikipedia dictionary (Table 3 in [113]). A recall of 86.85% and 88.67% is reported for ACE and MSNBC, respectively, at a cut-off level of 20. CrossWikis has a recall of 90.1% and 93.3% at the same cut-off.

the four NER types[13].

A more appropriate choice is 112 fine-grained entity types introduced by [86] in FIGER, a publicly available package [14]. These fine-grained types are not disjoint, i.e. each mention is allowed to have more than one type. For each mention, FIGER returns a set of types, each of which is accompanied by a score, $t_{\text{FIGER}}(m) = \{(t_j, g_j) : t_j \in T_{\text{FIGER}}\}$. A softmax function is used to probabilistically interpret the results as follows:

$$p(t_j|m,s) = \begin{cases} \frac{1}{Z}\exp(g_j) & \text{if } (t_j, g_j) \in t_{\text{FIGER}}(m), \\ 0 & \text{otherwise} \end{cases}$$

where $Z = \sum_{(t_k,g_k)\in t_{\text{FIGER}}(m)} \exp(g_k)$.

We evaluate the utility of entity types in Table 3.4, which shows that using NER typically worsens the performance. This drop may be attributed to the rigid binary values for type incorporation; it is hard to output the probabilities of the entity types for a mention given the chain model adopted in Stanford NER. We also notice that FIGER types consistently improve the results across the data sets, indicating that a finer-grained type set may be more suitable for the entity linking task.

To further confirm this assertion, we simulate the scenario where the gold types are provided for each mention (the oracle types of its gold entity). The performance is significantly boosted with the assistance from the gold types, which suggests that a better performing NER/FIGER system can further improve performance. Similarly, we notice that the results using FIGER types almost consistently outperform the ones using NER types. This observation endorses our previous recommendation of using fine-grained types for EL tasks.

3.5.4 Coherence

Two coherence measures suggested in Section 3.4.5 are tested in isolation to better understand their effects in terms of the linking performance (Table 3.5). In general, the link-based NGD works

[13]The Freebase types "/person/*" are mapped to PER, "/location/*" to LOC, "/organization/*" plus a few others like "/sports/sports_team" to ORG, and the rest to MISC.

[14]`http://github.com/xiaoling/figer`

Approach	TAC09	TAC10	TAC10T	TAC11	TAC12	AIDA-dev	AIDA-test	ACE	MSNBC
no COH	80.9	86.2	87.0	78.6	59.9	68.9	66.3	87.8	86.2
+NGD	**81.8**	85.7	86.8	79.7	63.2	**69.5**	**67.7**	88.0	**86.7**
+REL	81.2	86.3	87.0	79.3	63.1	69.1	66.4	**88.4**	86.6
+BOTH	81.4	**86.8**	**87.0**	**79.9**	**63.7**	69.4	67.5	**88.4**	**86.7**

Table 3.5: Performance (%) after re-ranking candidates using coherence scores, comparing two **coherence measures** (NGD and REL). "no COH": no coherence based re-ranking is used. "+BOTH": an average of two scores is used for re-ranking. Coherence in general helps: a combination of both measures often achieves the best effect and NGD has a slight advantage over REL.

slightly better than the relational facts in 6 out of 9 data sets (comparing row "+NGD" with row "+REL"). We hypothesize that the inferior results of REL may be due to the incompleteness of Freebase triples, which makes it less robust than NGD. We also combine the two by taking the average score, which in most data set performs the best ("+BOTH"), indicating that two measures provide complementary source of information.

3.5.5 Overall Performance

To answer the last question of how well does VINCULUM perform overall, we conduct an end-to-end comparison against two publicly available systems with leading performance:[15]

AIDA [63]: We use the recommended GRAPH variant of the AIDA package and are able to replicate their results when gold-standard mentions are given.

WIKIFIER [21]: We are able to reproduce the reported results on ACE and MSNBC and obtain a close enough B^3+ F1 number on TAC11 (82.4% vs 83.7%). Since WIKIFIER overgenerates mentions and produce links for common concepts, we restrict its output on the AIDA data to the mentions that Stanford NER predicts.

[15]We are also aware of other systems such as TagMe-2 [44], DBpedia Spotlight [93] and WikipediaMiner [95]. A trial test on the AIDA data set shows that both Wikifier and AIDA tops the performance of other systems reported in [27] and therefore it is sufficient to compare with these two systems in the evaluation.

Approach	TAC09	TAC10	TAC10T	TAC11	TAC12	AIDA-dev	AIDA-test	ACE	MSNBC	Overall
CrossWikis	80.4	85.6	86.9	78.5	62.4	62.6	62.4	87.6	82.6	76.3
+FIGER	81.0	86.1	86.9	78.8	63.5	66.7	64.5	87.8	83.6	77.7
+Coref	80.9	86.2	87.0	78.6	59.9	68.9	66.3	87.8	86.2	78.0
+Coherence =VINCULUM	**81.4**	**86.8**	**87.0**	79.9	63.7	69.4	67.5	**88.4**	86.7	79.0
AIDA	73.2	78.6	77.5	68.4	52.0	71.9	**74.8**	77.8	75.4	72.2
WIKIFIER	79.7	86.2	86.3	**82.4**	**64.7**	**72.1**	69.8	85.3	**88.2**	**79.4**

Table 3.6: **End-to-end performance**: We compare VINCULUM in different stages with two state-of-the-art systems, AIDA and WIKIFIER (%). The column "Overall" lists the average performance of nine data sets for each approach. CrossWikis appears to be a strong baseline. VINCULUM is 0.4% shy from WIKIFIER, each winning in four data sets; AIDA tops both VINCULUM and WIKIFIER on AIDA-test.

Table 3.6 shows the performance of VINCULUM after each stage of candidate generation (CrossWikis), entity type prediction (+FIGER), coreference (+Coref) and coherence (+Coherence). The column "Overall" displays the average of the performance numbers for nine data sets for each approach. WIKIFIER achieves the highest in the overall performance. VINCULUM performs quite comparably, only 0.4% shy from WIKIFIER, despite its simplicity and unsupervised nature. Looking at the performance per data set, VINCULUM and WIKIFIER each is superior in 4 out of 9 data sets while AIDA tops the performance only on AIDA-test. The performance of all the systems on TAC12 is generally lower than on the other dataset, mainly because of a low recall in the candidate generation stage.

We notice that even using CrossWikis alone works pretty well, indicating a strong baseline for future comparisons. The entity type prediction provides the highest boost on performance, an absolute 1.4% increase, among other subcomponents. The coherence stage also gives a reasonable lift.

In terms of running time, VINCULUM runs reasonably fast. For a document with 20-40 entity

	VINCULUM	AIDA	WIKIFIER
Mention Extraction	NER	NER	NER, noun phrases
Candidate Generation	**CrossWikis**	an intra-Wikipedia dictionary	an intra-Wikipedia dictionary
Entity Types	**FIGER**	NER	NER
Coreference	find the representative mention	-	re-rank the candidates
Coherence	link-based similarity, relation triples	link-based similarity	link-based similarity, relation triples
Learning	unsupervised	trained on AIDA	trained on a Wikipedia sample

Table 3.7: Comparison of entity linking pipeline architectures. VINCULUM components are described in detail in Section 3.4, and correspond to Figure 3.2. Components found to be most useful for VINCULUM are highlighted.

mentions on average, VINCULUM takes only a few seconds to finish the linking process on one single thread.

3.5.6 System Analysis

We outline the differences between the three system architectures in Table 3.7. For identifying mentions to link, both VINCULUM and AIDA rely solely on NER detected mentions, while WIKIFIER additionally includes common noun phrases, and trains a classifier to determine whether a mention should be linked. For candidate generation, CrossWikis provides better coverage of entity mentions. For example, in Figure 3.3, we observe a recall of 93.2% at a cut-off of 30 by CrossWikis, outperforming 90.7% by AIDA's dictionary. Further, [63] report a precision of 65.84% using gold mentions on AIDA-test, while CrossWikis achieves a higher precision at 69.24%. Both AIDA and WIKIFIER use coarse NER types as features, while VINCULUM incorporates fine-grained types that lead to dramatically improved performance, as shown in Section 3.5.3. The differences in

Category	Example	Gold Label	Prediction
Metonymy	South Africa managed to avoid a fifth successive defeat in 1996 at the hands of the All Blacks ...	South Africa national rugby union team	South Africa
Wrong Entity Types	Instead of Los Angeles International, for example, consider flying into Burbank or John Wayne Airport ...	Bob Hope Airport	Burbank, California
Coreference	It is about his mysterious father, Barack Hussein Obama, an imperious if alluring voice gone distant and then missing.	Barack Obama Sr.	Barack Obama
Context	Scott Walker removed himself from the race, but Green never really stirred the passions of former Walker supporters, nor did he garner outsized support “outstate”.	Scott Walker (politician)	Scott Walker (singer)
Specific Labels	What we like would be Seles , (Olympic champion Lindsay) Davenport and Mary Joe Fernandez .	1996 Summer Olympics	Olympic Games
Misc	NEW YORK 1996-12-07	New York City	New York

Table 3.8: We divide linking errors into **six error categories** and provide an example for each class.

Coreference and Coherence are not crucial to performance, as they each provide relatively small gains. Finally, VINCULUM is an unsupervised system whereas AIDA and WIKIFIER are trained on labeled data. Reliance on labeled data can often hurt performance in the form of overfitting and/or inconsistent annotation guidelines; AIDA's lower performance on TAC datasets, for instance, may be caused by the different data/label distribution of its training data from other datasets (*e.g.* CoNLL-2003 contains many scoreboard reports without complete sentences, and the more specific entities as annotations for metonymic mentions).

Error Category	TAC09	TAC10	TAC10T	TAC11	TAC12	AIDA-dev	AIDA-test	ACE	MSNBC
Metonymy	16.7%	0.0%	3.3%	0.0%	0.0%	60.0%	60.0%	5.3%	20.0%
Wrong Entity Types	13.3%	23.3%	20.0%	6.7%	10.0%	6.7%	10.0%	31.6%	5.0%
Coreference	30.0%	6.7%	20.0%	6.7%	3.3%	0.0%	0.0%	0.0%	20.0%
Context	30.0%	26.7%	26.7%	70.0%	70.0%	13.3%	16.7%	15.8%	15.0%
Specific Labels	6.7%	36.7%	16.7%	10.0%	3.3%	3.3%	3.3%	36.9%	25.0%
Misc	3.3%	6.7%	13.3%	6.7%	13.3%	16.7%	10.0%	10.5%	15.0%
# of examined errors	30	30	30	30	30	30	30	19	20

Table 3.9: **Error analysis:** We analyze a random sample of 250 of VINCULUM's errors, categorize the errors into six classes, and display the frequencies of each type across the nine datasets.

We analyze the errors made by VINCULUM and categorize them into six classes (Table 3.8). "Metonymy" consists of the errors where the mention is metonymic but the prediction links to its literal name. The errors in "Wrong Entity Types" are mainly due to the failure to recognize the correct entity type of the mention. In Table 3.8's example, the link would have been right if FIGER had correctly predicted the airport type. The mistakes by the coreference system often propagate and lead to the errors under the "Coreference" category. The "Context" category indicates a failure of the linking system to take into account general contextual information other than the fore-mentioned categories. "Specific Labels" refers to the errors where the gold label is a specific instance of a general entity, includes instances where the prediction is the parent company of the gold entity or where the gold label is the township whereas the prediction is the city that corresponds to the township. "Misc" accounts for the rest of the errors. In the example, usually the location name appearing in the byline of a news article is a city name; and VINCULUM, without knowledge of this convention, mistakenly links to a state with the same name.

The distribution of errors shown in Table 3.9 provides valuable insights into VINCULUM's varying performance across the nine datasets. First, we observe a notably high percentage of metonymy-related errors. Since many of these errors are caused due to incorrect type prediction by

FIGER, improvements in type prediction for metonymic mentions can provide substantial gains in future. The especially high percentage of metonymic mentions in the AIDA datasets thus explains VINCULUM's lower perfomance there (see Table 3.6).

Second, we note that VINCULUM makes quite a number of "Context" errors on the TAC11 and TAC12 datasets. One possible reason is that when highly ambiguous mentions have been intentionally selected, link-based similarity and relational triples are insufficient for capturing the context. For example, in "... while returning from Freeport to Portland. (TAC)", the mention "Freeport"is unbounded by the state, one needs to know that it's more likely to have both "Freeport" and "Portland" in the same state (*i.e.* Maine) to make a correct prediction [16]. Another reason may be TAC's higher percentage of Web documents; since contextual information is more scattered in Web text than in newswire documents, this increases the difficulty of context modeling. We leave a more sophisticated context model for future work [23, 127].

Since "Specific Labels", "Metonymy", and "Wrong Entity Types" correspond to the annotation issues discussed in Sections 3.3.2, 3.3.3, and 3.3.4, the distribution of errors are also useful in studying annotation inconsistencies. The fact that the errors vary considerably across the datasets, for instance, VINCULUM makes many more "Specific Labels" mistakes in ACE and MSNBC, strongly suggests that annotation guidelines have a considerable impact on the final performance. We also observe that annotation inconsistencies also cause reasonable predictions to be treated as a mistake, for example, AIDA predicts `KB:West Virginia Mountaineers football` for "..., Alabama offered the job to Rich Rodriguez, but he decided to stay at West Virginia. (MSNBC)" but the gold label is `KB:West Virginia University`.

3.6 Related Work

Building an end-to-end NEL system, whether a pipeline or a joint model, is often complicated. In this section, we review the critical components that commonly appear in the literature.

Mention Extraction: After sentence splitting and tokenization of the raw text, the first step of

[16] *e.g.* [31] use geo-coordinates as features.

all systems is to detect named entity mentions. It is often an application of an NER system [17, 63]. At the same time, the entity type information is obtained for feature generation in later stages. Besides detecting the boundaries of named entity mentions, some additional preparation is useful, *e.g.*, coreference resolution. All the occurrences of the same entity in the document could provide much more evidence than a single mention [30, 21]. Even in the datasets where the mentions are given, it is sometimes desirable, when the mention is an acronym, to find its full name within the same document. Unfortunately, neither NER or coreference resolution is perfect. The erroneous textual segments and/or entity types provided by NER cause cascaded linking mistakes. To avoid missing correct mentions, noun phrase chunking (NP chunking) is used to enlarge the set of possible mentions [113]. [125] proposed to overgenerate entity mentions by using both NER and NP chunking. To resolve the overlapping of the mentions, they jointly re-rank the choices of both the mention boundaries and the entity links within a small set of mentions that are close to each other. Coreference resolution can similarly provide misleading information. [55] found out that NEL could influence the decision in coreference resolution. For instance, the entity type "US President" for "Barack Obama" from Freebase suggests that a later mention "the president" is likely to refer to "Barack Obama" as well. They incorporate information of entity types and attributes from the linking results and update the links when coreference clusters are changed. The empirical results showed improvements in both NEL and coreference resolution tasks. The issues in mention extraction are usually ignored and considered a separate problem. In many NLP applications, the systems given gold mentions often outperform the counterparts with system mentions by a large margin (*e.g.*, Section 2.3).

Candidate Generation: The number of entities in a knowledge base such as Wikipedia is usually millions. For efficiency, usually less than a hundred more likely entities are pre-selected as candidates for each mention. Common techniques of mining candidates include title exact matching, page redirects and disambiguation pages from Wikipedia. Additionally, the text of Wikipedia articles is heavily annotated with links to other Wikipedia articles by the editors. The massive number of anchor links provide yet another useful resource for mention-entity pairs. An even larger repository of anchor links from the whole Web, called CrossWikis [132], was recently

released to the public. More than just the intra-Wikipedia links, it collects the occurrences of anchor links targeting to Wikipedia articles from normal Web pages. CrossWikis provides the empirical conditional probability of a Wikipedia entity given a textual mention. Moreover, CrossWikis is not limited to English.

Most candidate generation procedures aim at maximal recall so that the correct entity does not slip away from the candidates. However, an unwanted effect is a potentially huge size of candidates, which causes a lot of trouble for a sophisticated ranking model (*e.g.*, a learning-to-rank model whose time complexity is exponential w.r.t the list length [149]) or a joint linking system that considers all possible links for every mention within the same document (*e.g.*, [73]). [52] designed an exhaustive search of candidates and a filtering process to minimize the candidate size.

Candidate Ranking: A few common features used to rank the candidates include the popularity of the candidate entity (*e.g.*, "Bill Gates" is generally more likely to represent the co-founder of Microsoft), a similarity score between the Wikipedia article of the candidate and the mention context [30, 113], Wikipedia categories [17], topic modeling [70, 147, 56] and named entity types [109].

Coherence: Rather than ranking candidates independently for each candidate, Cucerzan [30] found that a mention could be better disambiguated when considering the predicted links of other mentions within the same document. The basic assumption made in [30] is that the entities mentioned in the same document should be closely related. The entity-entity relatedness function can be defined over the overlap of the Wikipedia categories two entities belong to [30] or the in-/out-link structure of the Wikipedia entity graph [95]. The exact collective disambiguation is NP hard [73]. Most systems in pursuit of efficiency make greedy approximation (*e.g.*, [30, 113]). [95] first fixes the entity predictions for the less ambiguous mentions and carries out the joint inference with a smaller set of variables, which makes the computation much less overwhelming. [73] converts the joint inference problem to an Integer Linear Programming and solves it with relaxation and rounding solutions. [57] uses a PageRank-style iterative method to propagate the confidence of the local entity assignment. The propagation matrix is defined based on the connected entity graph.

Other related work includes [58] that employs deep neural networks to learn entity representa-

tions and [107] that leverage sentence parses in Abstrast Meaning Representation [8] to implement an unsupervised entity linking system. Two other papers deserve comparison. [27] present a variety of evaluation measures and experimental results on five systems compared head-to-head. In a similar spirit, [53] provide an easy-to-use evaluation toolkit on the AIDA dataset. In contrast, our analysis focuses on the problem definition and annotations, revealing the lack of consistent evaluation and a clear annotation guideline. We also show an extensive set of experimental results conducted on nine datasets as well as a detailed ablation analysis to assess each subcomponent of a linking system.

3.7 Summary

Despite recent progress in Entity Linking, the community has had little success in reaching an agreement on annotation guidelines or building a standard benchmark for evaluation. When complex EL systems are introduced, there are limited ablation studies for readers to interpret the results. In this chapter, we examine nine EL datasets and discuss the inconsistencies among them. To have a better understanding of an EL system, we implement a simple yet effective, unsupervised system, VINCULUM, and conduct extensive ablation tests to measure the relative impact of each component. From the experimental results, we show that a strong candidate generation component (CrossWikis) leads to a surprisingly good result; using fine-grained entity types helps filter out incorrect links; and finally, a simple unsupervised system like VINCULUM can achieve comparable performance with existing machine-learned linking systems and, therefore, is suitable as a strong baseline for future research.

Chapter 4

ATTRIBUTE EXTRACTION FOR PHYSICAL OBJECT SIZE

4.1 Introduction

Commonsense knowledge plays an essential role in daily activities and allows us to fill in the blanks in the real world — what to expect and what to predict before or after an act. For instance, when we plan to go to some place, we estimate commute time based on our knowledge of the distance to the destination and how fast we will travel. Humans acquire common sense via learning from experiences while it remains an extremely difficult job for machines to acquire such knowledge from massive data [90, 131].

There have been some attempts to learn general rules from textual data [129, 111]. Unfortunately, there is little effort in learning a general knowledge base about numeric values. Specifically, a knowledge base capturing the size of physical objects does not exist. Humans apply previous knowledge about the size of common objects to interpret unfamiliar numbers. For example, to get a better sense of how tall the Space Needle (605 feet) is, one can interpret the number "605 feet" as approximately 60 stories since an average story is 10 feet tall. In addition, many machine understanding tasks can benefit from the knowledge of size. For instance, an automatic question answering system can query the knowledge base for the questions about size. In a language grounding task, where visual perception is connected to linguistic units, if a visual object recognized as "house" is smaller than another object linked to "person," prior knowledge of size will help the system render another interpretation. Similarly, an object recognition system will be warned if two objects in a similar size in the image are recognized as "elephant" and "mouse."

There are three major challenges to building such a KB. First, unlike a knowledge base about specific entities where there are clear answers, it is important to characterize a possible range of values for the object sizes (*e.g.*, it is reasonable to find a desk as wide as any value between 1

and 3 meters). As opposed to KBs about relations between specific entities, our goal is to extract relational knowledge about classes of objects (*e.g.*, the height range of all the vases). To address this issue, we extract from text an independent sample of values for each size attribute of an object and estimate a distribution from this sample, which defines a range of possible values. The second challenge is to develop an extractor for the size values. To extract the relationship between nouns and numeric values, a common approach is to train a classifier from a large set of labeled data, but such a training set does not exist and will be very costly to obtain. Another popular approach is to use distant supervision [28, 64], a heuristic which matches sentences against database records and automatically creates numerous training data. Unfortunately, it is hard to apply this heuristic when one of the arguments is a numerical value because an exact match is rare [136]; even worse, there is no such database to match against. Alternatively, we design a handful of lexico-syntactic pattern templates [60] and effectively extract size attributes from a massive Web crawl. Last but not least, the extraction in string form can be ambiguous. For instance, a door *lock* is around 50 millimeters tall whereas a *lock* for ships is often 50 meters tall. The word *lock* clearly denotes different meanings in two cases. To resolve the ambiguity, we canonicalize the extractions using WordNet senses [42]. Each sense (*synset*) represents a unique concept of physical object.

In this chapter, we present a system called SIZEITALL to address the above challenges. SIZEITALL runs on ClueWeb12[1] and extracts 386,217 values for 8,239 unique objects. In order to validate our results empirically, we ask the following two questions. How reasonable are our dimension estimates? How useful are they in real applications? To answer the first question, we collect three datasets consisting of human-annotated size attribute values for a set of objects and compare our results with two baseline systems: an informed random baseline and a regression model based on features of semantic relations on WordNet [135]. To verify the usefulness of the results, we carry out a prediction task to compare the size of two objects, which can be a common routine for down-stream applications such as object recognition.

[1] `http://lemurproject.org/clueweb12`

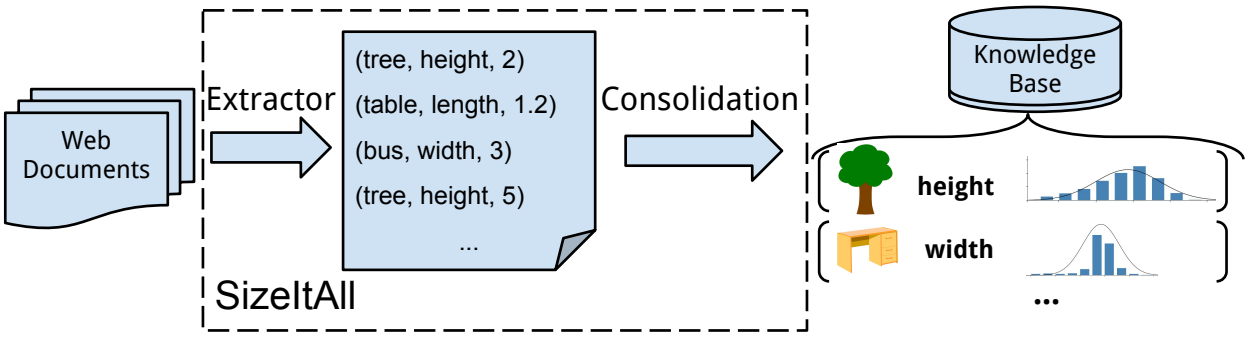

Figure 4.1: Overview of SIZEITALL. The extractor is applied to the source corpus and the resulting extractions are consolidated into a knowledge base.

4.2 Automatically Constructing a Knowledge Base on Object Sizes

In this section, we will describe how to build a knowledge base about size from textual data. Before diving into the details, let us state the problem at hand. The task is to extract numerical values about physical dimensions from a large text corpus and construct a knowledge base about the size of physical objects. We first apply an extractor E to each document d in a text corpus D and obtain a set of extractions in the form $\{t = (\texttt{Object}, \texttt{Attribute}, \texttt{Value})\}$. Then we consolidate all the extractions and construct a knowledge base with entries describing the sizes of canonical objects. An overview of SIZEITALL is depicted in Figure 4.2.

4.2.1 Size Extraction from Text

SIZEITALL starts with extracting individual pieces of evidence about the size of some object. We use ClueWeb12 as the source corpus. ClueWeb12 is a snapshot set of over 733 million English Web pages collected in 2012.

Size information can be stored in many forms, including natural language sentences (*e.g.*, "This table is 5 feet wide."), semi-structured HTML tables (*e.g.*, the product details section in Amazon product pages), labels in images (*e.g.*, blueprints), etc. In this work, we focus on extracting from

natural language sentences.[2] We seek textual segments that express size information about physical objects. There are three major dimensional attributes, namely width, length and height. Essentially, it is a relation extraction task of identifying expressions u and extract the target physical object c, the dimension attribute a and the number n with its size unit t, *i.e.*, $E(u) = (c, a, n, t)$.

Typically, one can train a statistical relation extractor to approximate the function E but this approach needs a significant amount of supervision from human labeling. Another popular approach is *distant supervision* [28, 64], which heuristically creates training examples by matching multiple entities in a sentence to an entry in a database. Unfortunately, it does not work very well when one of the entities is a numerical value because it is very rare to find an exact match with that value.[3] Instead, we design a set of lexico-syntactic patterns [60] to extract the size-related relations (Table 4.1). Each pattern starts with a noun phrase and is allowed to have an optional verb, both of which are omitted in the table due to space limitation. The noun phrase is then extracted as the physical object c. The dimension attribute a is determined by the adjective used to describe the size (*e.g.*, "as wide as 3 feet" represents a width value). The associated number n and its unit ut are also extracted. Although our patterns are fast and efficient for large-scale text processing, they ignore the linguistic structure and sometimes mistakenly extract the adjunct of a prepositional phrase as the object. For instance, in the sentence of "The table built by Joe is three feet tall," *Joe* is extracted as the object instead of *table*. We parse each sentence containing an extraction using the Berkeley parser [108], detect the true head of the prepositional phrase and replace the object of the extraction by the head noun if they differ.

The patterns we use are certainly not exhaustive but they cover a large number of sentences expressing size information. The patterns successfully extract over three hundred thousand extractions from the source corpus without any human labeling effort.

---

[2] Significantly different approaches are needed for other formats (e.g. WebTables [18]) and left for future work.

[3] A relaxed range match is possible but it is generally challenging to set an appropriate universal range.

Pattern	Example Sentence
as ADJ as NUM-UNIT	The table is as wide as 5 feet.
NUM-UNIT ADJ	The table is 5 feet wide.
of size NUM-UNIT ADJ (by NUM-UNIT ADJ){0,2}	The table is of size 5 feet wide by 3 feet tall.
NUM-UNIT ADJ (by NUM-UNIT ADJ)	The table is 5 feet wide by 3 feet tall.
NUM-UNIT in ADJ-NOUN	The table is 5 feet in width.

Table 4.1: Lexico-syntactic patterns for unary extraction. Each pattern starts with a noun phrase and an optional verb, which are omitted in the table. ADJ can be any adjective indicative of a dimension, *e.g.*, *wide, tall, long*. ADJ-NOUN is the nominal version of these adjectives, *e.g.*, *width, height, length*. NUM-UNIT stands for an expression of a combination of a number and a length unit (*e.g.*, *feet, meters, miles, etc.*). The matches are in the form of (Object, Attribute, Number, Unit). For instance, we identify a tuple $(c, a, n, t) =$ (table, width, 5, feet) from the running example in the table above.

4.2.2 Distribution Estimation

The goal is to acquire general knowledge about the size of physical objects. For example, we would like to know how wide a desk typically is, instead of the width of a particular one sold in a furniture store. In other words, we are interested in the distribution on the values of each object attribute, not a single data point. The individual extractions described in the previous subsection can be considered an independent sample from the corresponding distribution.

We assume that each dimensional attribute of a certain type of object follows a log-normal distribution. Although normal distributions are commonly used to model the distribution of an independent sample, it is ill-suited in our setting. The size values are always positive. Using a normal distribution will cause probability leakage to the negative values. The same practice has also been adopted in recent literature [7, 72].

We first normalize all the extractions in the same unit, *meter*. Then we can omit the unit and group the triples $\{(c, a, n)\}$ by the object-attribute pairs. Each object attribute corresponds to a set

of values, from which we estimate the mean $\mu_{c,a}$ and the standard deviation $\sigma_{c,a}$ of a log-normal distribution. We detect and remove outliers using the median-based z-score [67]. In practice, we find the median value $m_{c,a}$ is more stable than the mean value $\mu_{c,a}$ because the mean value can be affected by abnormally large or small values due to extraction noise.

4.2.3 Disambiguation to Word Senses

A word often has different meanings in various contexts. For example, *lock* can represent a device that must be opened with a key (*e.g.*, a door lock), or an enclosure that controls the water level (*e.g.*, a ship lock). Each meaning of the word often has a different size scale. To avoid the ambiguity, we define the set of objects using WordNet [42], a lexical resource of unambiguous semantic units called *synset*s or *word sense*s, each of which corresponds to a unique meaning. Since only physical entities can be measured, we narrow down all the synsets to the descendants of the noun synset `physical-entity` following WordNet's is-a hierarchy, which forms a set of canonical objects P_O. In addition, we exclude mass nouns from this set since they do not have a stable volume (*e.g.*, the size of some milk varies with the container holding it). A list of mass nouns is curated based on Wiktionary (`wiktionary.org`).

A natural next step is to map the extractions, especially the object nouns, into some synset of P_O. This problem is called *Word Sense Disambiguation* (WSD) in the literature [105]. Our preliminary experiments with the existing methods (*e.g.*, [4]) proved to be slow and inaccurate. Instead, we choose an approach that simply assigns each word form all possible synsets it can realize. Every synset s is associated with its lexical word forms or phrases w_s. For instance, the military vehicle synset, `tank`, can be lexically realized by a single word, *tank*, or a phrase, *army tank* (*i.e.*, $w_{\texttt{tank}} = \{\textit{tank}, \textit{army tank}\}$). For each synset, we collect the extractions from all the word forms of this synset and estimate the distribution from the union of these extractions. This can be viewed as an approximation of WSD by assuming that each word form occurrence represents all its possible synsets. Despite a noisy process, it can still have a reasonable estimate if the union of the extractions contains more size values for the correct synset than each individual set of extractions.

4.3 Experiments

In this section, we present the experimental results to answer two main questions:

1. How accurate are the estimated values in the KB?
2. Are the extractions useful for an end task?

4.3.1 Experiment: Size Attribute Extraction

The first experiment we conducted is to evaluate how close the size estimates in the KB are to the real values (or the size distributions of real objects). We compare the results in SIZEITALL against two baseline approaches on three datasets annotated by human subjects.

Experimental Setup

We run SIZEITALL over ClueWeb12 which consists of over 700 million English Web pages. SIZEITALL uses Boilerpipe[4] to remove the boilerplate content in Web pages. We use Stanford CoreNLP [87] to tokenize the content text, split the sentences, and predict the lemmas, Part-of-Speech (POS) tags, and Named Entity Recognition (NER) tags for each token. POS tags (*e.g.*, CD) and NER tags (*e.g.*, NUMBER) are used to detect numerical expressions in both Arabic numerals and text (*e.g.*, 300 and three hundred). We identify the matches of the lexico-syntactic patterns introduced (Section 4.2.1) using TokensRegex [20]. We also implement a unit converter to normalize all the extracted values into a common unit, *meter*. SIZEITALL successfully extract 386,217 extractions about 8,239 unique objects. We consolidate the extractions by grouping the extractions by object-attribute pairs. Each group consists of a set of extracted values, from which a log-normal distribution is estimated.

Dataset	# objects	Attributes	# values
JH	395	height and length	790
TT15U [135]	262	a single measure	262
KO11U [72]	93	height, length and width	196

Table 4.2: A summary of unary datasets. In JH, only height and length attributes are annotated. In TT15U, one single measure was labeled. In KO11U, all three dimensions of the objects are reported if possible (*e.g.*, the length of `tea-bag` is neglected).

Datasets

We collect three datasets to measure the accuracy of the extractions. An overview of the datasets is shown in Table 4.2.

JH: The dataset was originally created to facilitate the development of an interactive reader, which automatically interprets out-of-context numbers into multiples of commonly known values (*e.g.*, 380 meters is twice the height of the Space Needle)[5]. The data was collected via crowdsourcing on Mechanical Turk (MTurk)[6]. Each worker is asked to estimate the height and the length of an object according to their common sense. Each object corresponds to a WordNet synset. The MTurk workers are shown the gloss definition and some automatically extracted images from ImageNet [34] as a visual definition. Each object collects 15 guesses from different MTurk workers.

TT15U: Probably the closest dataset for our evaluation purpose was developed in Takamura and Tsujii [135]. 262 objects are annotated, contrary to our modeling in three dimensions, each of which is assigned one single value as a surrogate for the scale of its size. Also note that the annotation was made with Japanese WordNet. Despite the shared synset Ids, the Japanese version is automatically created via machine translation and therefore contains possible errors.

KO11U: In a psychology experiment where test subjects are asked to group objects in a similar

[4] `https://github.com/kohlschutter/boilerpipe`

[5] `http://visualization.ischool.uw.edu/units_app/project_details.html`

[6] `http://mturk.com`

size [72], 100 unique objects were hand-annotated with their real world measurements (height, length and width if the attribute value can be measured). The objects in the original dataset are in the string form. To facilitate the evaluation with WordNet synsets, we annotate each object with their appropriate synset label for KO11U and removed 7 objects for which no matching synset can be found (*e.g.*, moneyroll).

Evaluation Protocol

We use the *average absolute log difference* between the system estimate and the gold label to measure the accuracy of the system output (both values are normalized in the same unit, meter). It measures how close our estimates are to the gold standard values. Using the log values prevents an imbalanced average towards large difference values. The same metric was also used in [135]. The JH dataset provides multiple annotations for each object attribute. We assume that the annotations follow a log-normal distribution, which allows us to measure the *accuracy* by checking if the estimate is within one standard deviation of the mean of the distribution, where 68% of the probability mass is. In other words, if we asked a MTurk worker for another estimate, the annotation would have been within the same range in 68% of the time. Therefore, an accuracy around 68% is near-human performance.

Despite over eight thousand unique objects found by SIZEITALL, there remain some objects whose size information is missing. For that, we report *coverage* of the test set and compute the accuracy numbers on the subset covered by the system.

We compare SIZEITALL with two baseline systems. The first one is an informed random baseline. A log-normal distribution is estimated from all the gold values in three datasets. For each attribute of an object, a value is randomly sampled. Another system we compare with is a regression model proposed in [135] where TT15U was first introduced. To accommodate TT15U where only one number is labeled for each object, we calculates the geometric mean (g) of the attribute values available for each object (*e.g.*, $g(e) = (m_{e,h} * m_{e,l} * m_{e,w})^{1/3}$) and then compare the geometric mean value to the gold standard. We choose geometric mean as the delegate of volume, which is the common value used for size comparison.

Dataset	Random	TT15	SIZEITALL
TT15U	2.32 / **100%**	3.34 / **100%**	**1.31** / 40.5%
KO11U	2.24 / **100%**	-	**1.11** / 58.7%
JH	1.59 / **100%**	-	**1.22** / 73.0%
JH(±std)	28.8%	-	**49.9%**

Table 4.3: Size extraction performance. In the rows of TT15U, KO11U and JH, *average absolute log difference* and coverage numbers are shown for each method. The last row show the percentage of the estimates falling within one standard deviation from the mean estimated from crowdsourced labels. Best results for each dataset are in bold. The performance of TT15U is from the best number in Figure 1(a) of [135].

Results

Table 4.3 shows the results of SIZEITALL and two compared systems. The random baseline generally has a poor performance except on JH where the gold values are very close to the mean of the informed normal distribution of the random baseline. To better understand the numbers, we can interpret a log difference around 2 as that the estimate is either $e^2 = 7.4$ times or $1/e^2 = 0.13$ of the real value. SIZEITALL significantly improves the performance, especially on TT15U and KO11U. To our surprise, the regression model proposed in [135] underperforms the random baseline, indicating that the WordNet-derived features in the regression model are probably insufficient to predict the size value.

We notice that SIZEITALL has a coverage around 40%-73%. To make a fair comparison with the random baseline, we ran the baseline on the same subset where SIZEITALL has an estimate. The performance is in general similar to its performance on the whole set. We are unable to make the same comparison with TT15 on a subset because the the raw predictions are not available. SIZEITALL has somewhat low coverage on TT15U. It might be attributed to the annotation process of the dataset. TT15U was originally annotated on Japanese WordNet, which is created based on English WordNet via automatic translation. It is likely that the meaning is lost during the translation. It is also possible that the object mention distribution is different across languages.

Dataset	# of labeled pairs	# of unique objects
MTurk	348	251
TT15B [135]	1152	1638
BF15B [7]	486	41

Table 4.4: A summary of binary datasets.

On the last row of Table 4.3, we observe that almost half of the estimates by SIZEITALL on JH are within one standard deviation from the mean of the crowd labels, which is close to human-level performance ($\sim$68%).

4.3.2 Experiment: Size Comparison

Now we evaluate how useful the extracted knowledge base is in an end task, comparing the size of two objects. Humans can easily tell if an elephant is larger than a mouse. We use this task to test if the extracted height, width, and length values can be used to predict if one object is larger than the other.

Method

For a pair of objects (e_1, e_2), we calculate the geometric mean (g) of the attribute values available for each object and compare two geometric mean values $(g(e_1), g(e_2))$ to predict the relation. It is equivalent to that we compare the volume size of both objects assuming that the objects are rectangular. If SIZEITALL has no information about at least one object in comparison, an "unclear" relation is predicted.

Datasets

We created a dataset via crowd-sourcing and adopted another two datasets for evaluation.

MTurk: We created a web interface where a MTurk worker is asked to provide three objects that

are larger than a specified object as well as three objects that are smaller. The workers are also asked to select the corresponding synset for each answer. We selected 23 synsets for annotations and a total number of 348 annotations are collected from 30 workers after the duplicates are removed.

TT15B: With the same setup as in TT15U, a random sample of 1,152 synset pairs are annotated by Takamura and Tsujii [135].

BF15B: In Bagherinezhad et al. [7], the authors introduced a dataset consisting of 486 comparison pairs. A set of 41 objects are selected initially. All possible pairs of objects are annotated. Each pair is given a choice between bigger, smaller, and not obvious. The object pairs which have consistent obvious labels are kept for evaluation. The overall comparisons are further verified to form a directed acyclic graph where the nodes are the objects and the edges indicate the size comparison between two nodes.

Evaluation Protocol

We present the accuracy measure, how accurate the predictions are, as well as the coverage of test object pairs, how many pairs the system can predict (i.e. predicting a relation except "unclear"). The same informed random baseline is used for comparison. On both TT15B and BF15B, we compare with the results reported in the original papers, named TT15 [135] and BF15-Text [7] respectively. We also compare with the full approach described in [7], BF15-Full, which incorporate additional vision knowledge for the predictions.

Results

The results are shown in Table 4.5. It is no surprise that the random baseline shows an even chance for two outcomes on all three datasets. However, it is somewhat surprising to see an 80% accuracy by TT15 despite its imprecise absolute size values in the previous experiment. A possible cause is that the regression model of TT15 also takes into account over a thousand binary orderings of objects [135]. The inaccuracy of absolute values may have been overcome by the more informative orderings of the object size. SIZEITALL is more accurate on TT15B compared against TT15.

Dataset	Random	TT15	BF15-Text	BF15-Full	SIZEITALL
MTurk	52.3% / **100%**	-	-	-	**78.9%** / 58.6%
TT15B	50.0% / **100%**	80.0% / **100%**	-	-	**85.8%** / 37.2%
BF15B	49.4% / **100%**	-	75.3% / **100%**	**83.5%** / **100%**	79.9% / 81.9%

Table 4.5: Size comparison performance. Both accuracy and coverage are reported in each cell. Best results for each dataset are in bold.

Similar to what we have seen in the previous experiment, SIZEITALL's coverage on TT15B is relatively low.

On the BF15B dataset, SIZEITALL outperforms BF15-Text. Although the full approach in [7] has the highest accuracy, the visual cues used in the full approach can similarly assist SIZEITALL to further improve the performance.

4.4 Related Work

In this section, we discuss previous work on commonsense knowledge extraction and word sense disambiguation.

4.4.1 Commonsense Knowledge Extraction

There is a great deal of related work on large scale commonsense knowledge extraction and modeling, resulting in many impactful projects such as OpenCyc [90] and ConceptNet [131]. However, Cyc's manual editing by a small group of experts is expensive and not scalable and ConceptNet's automatic method from natural language lacks formal definitions of extracted relations. The rise of crowd-edited Wikipedia resolved both issues and resulted in many wide-spread knowledge bases such as DBpedia [6], Yago [133] and Freebase [12] along with successful application in numerous problems (*e.g.*, Wikification [95], question answering [74], etc.).

The majority of the above research focuses on named entities and relations between them. Unfortunately, there is very limited work on mining numerical attributes of classes of entities. There

are in fact a few recent attempts. Aramaki *et al.* [5] showed that object size features are useful for semantic relation identification. Davidov and Rappoport [32] showed that the attribute value of a particular can be approximated by the value of the same attribute from similar objects. Narisawa *et al.* [102] proposed a method to determine if a real number is large or small for a particular attribute by identifying the modifier describing a value (*e.g.*, "as expensive as 3,000 dollars" implies 3,000 dollars is a relatively large value). Takamura and Tsujii [135] presented a regression model for object size and showed that ordering information between two objects can be helpful in addition to the absolute values. Most of the above work collects values by issuing queries to a search engine and parse the returned search snippets. Two issues are worth mentioning. First, the number of return results by the search engine is limited (less than a thousand snippets). A small sample can cause inaccurate estimation. Second, the search results may not be perfect and the result snippets are likely incomplete. A cascade of errors can bring noise to the final results. In contrast, we directly extract values from the Web documents, effectively avoiding both problems.

4.4.2 Word Sense Disambiguation

Word Sense Disambiguation (WSD) is the task of finding the correct meaning of each word in some context. The research on this problem has a long history dating back to the late 1940s. A comprehensive survey on this topic has been published by Navigli [105]. In this work, the main focus is to disambiguate the object noun in our extractions. We adopt a simple approach by assuming each word form represents all the possible synsets it can realize. Although this approximate method can make noisy predictions unless the word form is unambiguous, the redundancy of the extracted values alleviates this issue in practice. More sophisticated models [4, 99] are desired but are likely to be slow without a significant improvement in disambiguation accuracy.

4.5 Discussion

4.5.1 Canonical Object Set

In this work, we choose a subset of WordNet (the noun synsets that are descendants of `physical-object`) as the base set of canonical objects. However, this set is not perfect. Some synsets are too subtle to physically characterize a useful class of objects, such as `native` (indigenous plants and animals), or too fine-grained to practically useful, *e.g.*, a synset for the car `ignition-lock`. An alternative is the object set of ImageNet [34], which is also a subset of WordNet but only visually distinguishable synsets are kept.

The downside of both choices, derived from WordNet, is the coverage. Things that appeared after the creation of WordNet, such as `iPhone`, are excluded. Fortunately, the size extraction component of SIZEITALL is not limited to WordNet and can be extended to out-of-WordNet objects.

4.5.2 Ambiguity in Extractions

The lexico-syntactic patterns introduced in Section 4.2.1 are intuitive and effective but the extraction without document-wide context might cause ambiguities and errors. We present three representative categories with examples. The first category is to confuse real objects with man-made artifacts in a disproportionate size (*e.g.*, toys of animals, sculptures of celebrities, etc.). For instance, it is common to see a page from a shopping website where the size of a stuffed animal is described in the same way one describes the real animal, *e.g.*, "The Lion is 1 inch high." Another category of getting unrealistic values is to extract numerical values from sentences in novels or hypothetical statements (*e.g.*, "... we see the gigantic walls, 1500 miles high and 1500 miles wide" or "I tried to hang myself by my own fake rope that was 8320 kilometers long"). The last category is missing references. A typical error is to extract the size of `mine` where the word "mine" is used as a possessive noun. We need to find the actual referred object from the previous context (*e.g.*, what object is "mine" referred to). It is possible to build classifiers to resolve the first two types of errors and identifying coreferent noun phrases can alleviate the third issue.

4.5.3 Bias in Commonsense Knowledge Extraction

When some text is produced, the writer often wants to be truthful as well as concise. In [50], Grice observed this behavior and pointed out four categories of conversation principles — quality, quantity, relation and manner — also known as Grice's Maxims. The second principle "quantity" is interpreted as "say as much as necessary, but no more" [129]. A fact is often omitted in the text when it can be easily inferred from already stated facts and shared commonsense knowledge between the writer and the readers. For example, it is commonly known that a `road` is flat and therefore only length and width are the measurable attributes. The lack of mentioning the height of a `road` may be a good clue for a system to learn its shape. Another interesting case arises when an object can be viewed in the multiple angles and the same dimension is called by different attribute names: the same edge of a flat-screen monitor can be called either width or length; the height of a standing pole is equivalent to the length of the same pole when it is laid down on the ground.

4.5.4 Relative Order of Object Size

In this work, we extract absolute clues about the size of things from text. Another useful source of size information is the relative order of the size of two objects. Similarly, we can extract object pairs from text by pattern matching (*e.g.*, "X is larger than Y" or "X is wider than Y"). Furthermore, we can leverage implicit clues where a statement between two objects implies the relative order of their size (*e.g.*, the sentence "I put the basket in the trunk of my car" implies that the *basket* is smaller than the *trunk* of a car otherwise it will not fit). With such evidence, we can construct a directed graph encoding these pairwise relations. We can then compare a pair of objects in size by checking if there is a path from one object to the other. Furthermore, we can use this information to improve the quality of size attributes by tuning the estimates of the size attributes with respect to the object pairs with their relative order in size. Unfortunately, as mentioned in the previous section, there is also a strong bias on the occurrences of the relative clues [7]. For instance, it is less often to see an explicit statement of the common case, *dogs are larger than cats*, than the "surprise" case, *My cat is bigger than that dog!!!*.

4.6 Summary

In this chapter, we present a system, SIZEITALL, that extracts 386,217 size values from a large Web corpus and construct a knowledge base on the dimensions of physical objects. We evaluate the size extractions against three hand-labeled datasets and the results showed a superior performance of SIZEITALL compared with two baselines. We also evaluate the constructed KB via a prediction task of size comparison. The experimental results show positive evidence in the use of the knowledge base in down-stream applications.

Chapter 5
CONCLUSION

In previous chapters, we proposed three systems for entity analysis. In Chapter 2, we described FIGER that predicted entity types with a set of over a hundred of fine-grained entity types. We showed that a relation extractor can be improved by adding informative features based on the entity types predicted by FIGER. In Chapter 3, we presented a deterministic entity linking system, VINCULUM, connecting entity mentions with their corresponding entries in a knowledge base. VINCULUM consists of multiple components, including candidate generation, entity type prediction, coreference resolution and coherence ranking. We performed a comprehensive analysis of the relative impact by each component and compare VINCULUM with two state-of-the-art machine learned systems. In Chapter 4, we proposed an approach to extract numerical attribute values for physical object size. The SIZEITALL system extracts numerical values for different object dimensions and consolidates the extractions into a knowledge base. We showed that it is promising to apply the knowledge base in an end task of comparing the size of two objects.

5.1 Limitations of This Work

In this section, we discuss several fundamental limitations of this work. Despite the positive results shown in previous chapter, we are still far from a fully automated system for entity analysis.

First, the type set for FIGER was manually created. Although the type set turned out to be useful for downstream applications [71, 145, 84], it still lacks the coverage and the granularity due to somewhat subjective nature of the type selection. Ideally, one would want a system capable of inducing a set of significant entity types from the source data. When a specific task is present and defined, the system creates a mapping from the induced set of entity types to the target types (specified by the user via a definition or some seed entities) or further adapt the induced types to

accommodate the task. A promising recent thread of research related to this issue is Universal Schema [115, 144]. The basic idea is to keep both lexical type predicates and canonical types inherited from an existing KB and the correlations across two sets of types can be learned via matrix factorization. In addition, the FIGER type set is designed for entity typing in the news domain. Entities appearing in other domains such as Twitter or biomedical literature often follow a very different distribution. Consequently, automatic type induction (or type selection) becomes necessary for such domain adaption.

Second, VINCULUM is not capable of discovering new entities. The system predicts if an entity mention is NIL but cannot identify if two NIL mentions refer to the same real-world entity. Existing KBs are often incomplete. For instance, it is not practical for Wikipedia to document every single person in the world; it is unlikely for one to document every emerging entity in a KB. Although we may use FIGER to assign the same types to NIL mentions of the same entity, it would be more useful to cluster them and automatically create new entities to complement existing KBs. In fact, TAC-KBP created a related track, Cold Start KBP[1]. Each participant team is given a document collection about a less known place and required to construct a KB from scratch. The majority of the submissions simply cluster mentions by their text strings. Although this basic approach shows reasonable performance in the competition, apparently more work needs to be done to resolve this issue.

Third, SIZEITALL aims at extracting commonsense knowledge about physical dimensions of objects. However, as discussed in the previous chapter, there is still enormous information not shown in text. For example, it is rare to find an expression about the length (or width or height) of an apple. One of the reasons is that an apple is a round object and the natural measurement is its diameter. A more important reason is that the size of an apple is too visually obvious to human so that it becomes unnecessary to explicitly express it unless the apple is unusually big or small. In that case, the system will end up extracting all the unusual data points, causing a strong bias in the final estimate. Clearly, integrating visual perception and language will be extremely useful for building a

[1] http://www.nist.gov/tac/2015/KBP/ColdStart/

complete commonsense knowledge base.

5.2 Future Directions

Many future directions are worth pursuing, we discuss a few:

5.2.1 Holistic Entity Analysis

From the experimental results in previous chapters, we find that fine-grained type predictions are useful to distinguish the candidates in EL (Section 3.5.3); relations help collectively identify entities (Section 3.5.4); and relation extraction can be improved with fine-grained types (Section 2.3.2). In this work, we design pipeline models mainly because it is easy to assemble the components and we believe that the improvement of each component will be reflected in the final performance of the target task. However, this design choice prevents mutual benefits across tasks. For instance, a joint objective of entity typing and linking can potentially improve both tasks. Despite its complexity, joint models ideally are a better choice of modeling the interdependence between individual models (*e.g.*, [36, 141, 126]).

5.2.2 Interactive Information Extraction and Knowledge Base Construction

Information extraction and knowledge base construction should ideally be an iterative process. It starts with using existing knowledge to teach and build the extractors, applying the extractors to (unlabeled) data, and adding the extracted facts back to the original knowledge base. The whole process then repeats itself. A closely related work to this idea is Never Ending Language Learning [19]. In addition to a continuous loop of the above two steps, periodic human supervision is allowed in the form of 5-minute labeling of the extractions. This input is useful to prevent an unexpected error that could cause collapsing of the whole system. However, the human supervision is unidirectional, which requires that the annotator have the expertise in the system. To close the loop, we need to design a mechism to allow the machine to select the extractions in low confidence and ask relevant questions to improve itself [123, 83] and to harvest the power of crowdsourcing [66, 82, 13].

BIBLIOGRAPHY

[1] Tac kbp entity selection. `http://www.nist.gov/tac/2012/KBP/task_guidelines/TAC_KBP_Entity_Selection_V1.1.pdf`, 2012.

[2] Tac kbp 2013 entity linking track, 2013.

[3] Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94. ACM, 2000.

[4] Eneko Agirre and Aitor Soroa. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–41. Association for Computational Linguistics, 2009.

[5] Eiji Aramaki, Takeshi Imai, Kengo Miyo, and Kazuhiko Ohe. Uth: Svm-based semantic relation classification using physical sizes. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 464–467. Association for Computational Linguistics, 2007.

[6] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference (ISWC/ASWC-2007)*, pages 722–735, 2007.

[7] Hessam Bagherinezhad, Hannaneh Hajishirzi, Yejin Choi, and Ali Farhadi. Are elephants bigger than butterflies? reasoning about sizes of objects. In *AAAI*, 2016.

[8] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In *The 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186. Association for Computational Linguistics, 2013.

[9] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-2007)*, pages 2670–2676, 2007.

[10] Matthew Berland and Eugene Charniak. Finding parts in very large corpora. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 57–64. Association for Computational Linguistics, 1999.

[11] Steven Bethard and James H Martin. Identification of event mentions and their semantic class. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 146–154. Association for Computational Linguistics, 2006.

[12] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.

[13] Jonathan Bragg, Andrey Kolobov, and Daniel S Weld. Parallel task routing for crowdsourcing. In *Second AAAI Conference on Human Computation and Crowdsourcing*, 2014.

[14] Sergey Brin. Extracting patterns and relations from the world wide web. In *The World Wide Web and Databases*, pages 172–183. Springer, 1999.

[15] Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479, 1992.

[16] Razvan Bunescu and Raymond Mooney. Learning to extract relations from the web using minimal supervision. In *Annual meeting-association for Computational Linguistics*, volume 45, page 576, 2007.

[17] Razvan C Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics*, volume 6, pages 9–16, 2006.

[18] Michael J. Cafarella, Alon Y. Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. Webtables: exploring the power of tables on the web. *Proceedings of the International Conference on Very Large Databases (VLDB-2008)*, 1(1):538–549, 2008.

[19] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2010.

[20] Angel X. Chang and Christopher D. Manning. TokensRegex: Defining cascaded regular expressions over tokens. Technical Report CSTR 2014-02, Department of Computer Science, Stanford University, 2014.

[21] Xiao Cheng and Dan Roth. Relational inference for wikification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013.

[22] Nancy Chinchor and Patricia Robinson. Muc-7 named entity task definition. In *Proceedings of the 7th Message Understanding Conference (MUC-7)*, 1997.

[23] Andrew Chisholm and Ben Hachey. Entity disambiguation with web links. *Transactions of the Association for Computational Linguistics*, 3:145–156, 2015.

[24] Eunsol Choi, Tom Kwiatkowski, and Luke Zettlemoyer. Scalable semantic parsing with partial ontologies. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, 2015.

[25] Michael Collins. *Head-driven statistical models for natural language parsing*. PhD thesis, University of Pennsylvania, 1999.

[26] Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 189–196, 1999.

[27] Marco Cornolti, Paolo Ferragina, and Massimiliano Ciaramita. A framework for benchmarking entity-annotation systems. In *Proceedings of the 22nd international conference on World Wide Web*, pages 249–260. International World Wide Web Conferences Steering Committee, 2013.

[28] Mark Craven and Johan Kumlien. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology (ISMB-1999)*, pages 77–86, 1999.

[29] Mark Craven, Andrew McCallum, Dan PiPasquo, Tom Mitchell, and Dayne Freitag. Learning to extract symbolic knowledge from the world wide web. Technical Report CMU-CS-98-122, Carnegie Mellon University, 1998.

[30] Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, volume 2007, pages 708–716, 2007.

[31] Silviu Cucerzan. The msr system for entity linking at tac 2012. In *Text Analysis Conference 2012*, 2012.

[32] Dmitry Davidov and Ari Rappoport. Extraction and approximation of numerical attributes from the web. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1308–1317. Association for Computational Linguistics, 2010.

[33] Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. Large-scale object classification using label relation graphs. In *Computer Vision–ECCV 2014*, pages 48–64, 2014.

[34] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 248–255. IEEE, 2009.

[35] George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie Strassel, and Ralph M Weischedel. The automatic content extraction (ace) program–tasks, data, and evaluation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, volume 4, pages 837–840, 2004.

[36] Greg Durrett and Dan Klein. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics*, 2:477–490, 2014.

[37] Asif Ekbal, Eva Sourjikova, Anette Frank, and Simone Paolo Ponzetto. Assessing the challenge of fine-grained named entity recognition and classification. In *Proceedings of the 2010 Named Entities Workshop*, pages 93–101. Association for Computational Linguistics, 2010.

[38] Micha Elsner, Eugene Charniak, and Mark Johnson. Structured generative models for unsupervised named-entity clustering. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 164–172. Association for Computational Linguistics, 2009.

[39] Oren Etzioni, Michele Banko, and Michael J Cafarella. Machine reading. In *Proceedings of the Twenty-First AAAI Conference on Artificial Intelligence*, volume 6, pages 1517–1519, 2006.

[40] Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. Open information extraction: The second generation. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume One*, pages 3–10. AAAI Press, 2011.

[41] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 2011.

[42] Christiane Fellbaum. *WordNet: An electronic lexical database*. The MIT press, 1998.

[43] Paolo Ferragina and Ugo Scaiella. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1625–1628. ACM, 2010.

[44] Paolo Ferragina and Ugo Scaiella. Fast and accurate annotation of short texts with wikipedia pages. *IEEE Software*, 29(1):70–75, 2012.

[45] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

[46] Michael Fleischman and Eduard Hovy. Fine grained classification of named entities. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics, 2002.

[47] Dan Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. Context-dependent fine-grained entity type tagging. *arXiv preprint arXiv:1412.1820*, 2014.

[48] Roxana Girju, Adriana Badulescu, and Dan Moldovan. Learning semantic constraints for the automatic discovery of part-whole relations. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 1–8. Association for Computational Linguistics, 2003.

[49] Claudio Giuliano and Alfio Gliozzo. Instance-based ontology population exploiting named-entity substitution. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 265–272. Association for Computational Linguistics, 2008.

[50] Herbert P Grice. *Logic and conversation*. 1970.

[51] Ralph Grishman and Beth Sundheim. Message understanding conference-6: A brief history. In *Proceedings of the 16th International Conference on Computational Linguistics*, volume 96, pages 466–471, 1996.

[52] Yuhang Guo, Bing Qin, Yuqin Li, Ting Liu, and Sheng Li. Improving candidate generation for entity linking. In *Natural Language Processing and Information Systems*, pages 225–236. Springer, 2013.

[53] Ben Hachey, Joel Nothman, and Will Radford. Cheap and easy entity evaluation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014.

[54] Aria Haghighi and Dan Klein. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1152–1161. Association for Computational Linguistics, 2009.

[55] Hannaneh Hajishirzi, Leila Zilles, Daniel S. Weld, and Luke Zettlemoyer. Joint Coreference Resolution and Named-Entity Linking with Multi-pass Sieves. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013.

[56] Xianpei Han and Le Sun. An entity-topic model for entity linking. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 105–115. Association for Computational Linguistics, 2012.

[57] Xianpei Han, Le Sun, and Jun Zhao. Collective entity linking in web text: a graph-based method. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 765–774. ACM, 2011.

[58] Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. Learning entity representation for entity disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 2013.

[59] Zhengyan He, Shujie Liu, Yang Song, Mu Li, Ming Zhou, and Houfeng Wang. Efficient collective entity linking with stacking. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 426–435, 2013.

[60] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 15th International Conference on Computational Linguistics*, pages 539–545, 1992.

[61] Johannes Hoffart, Yasemin Altun, and Gerhard Weikum. Discovering emerging entities with ambiguous names. In *Proceedings of the 23rd international conference on World wide web*, pages 385–396, 2014.

[62] Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. Yago2: a spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

[63] Johannes Hoffart, Mohamed A. Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 782–792. Association for Computational Linguistics, 2011.

[64] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 541–550, 2011.

[65] Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. Ontonotes: the 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60. Association for Computational Linguistics, 2006.

[66] Jeff Howe. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4, 2006.

[67] Boris Iglewicz and David Hoaglin. Volume 16: how to detect and handle outliers. *The ASQC Basic Reference in Quality Control: Statistical Technique*, 1993.

[68] Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. Overview of the tac 2010 knowledge base population track. In *Text Analysis Conference (TAC 2010)*, 2010.

[69] Jing Jiang and ChengXiang Zhai. A systematic exploration of the feature space for relation extraction. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 113–120, 2007.

[70] Saurabh Kataria, Krishnan S. Kumar, Rajeev Rastogi, Prithviraj Sen, and Srinivasan H. Sengamedu. Entity disambiguation with hierarchical topic models. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1037–1045, 2011.

[71] Mitchell Koch, John Gilmer, Stephen Soderland, and Daniel S Weld. Type-aware distantly supervised relation extraction with linked arguments. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014.

[72] Talia Konkle and Aude Oliva. Canonical visual size for real-world objects. *Journal of experimental psychology: human perception and performance*, 37(1):23, 2011.

[73] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective annotation of Wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 457–466. ACM, 2009.

[74] Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, and Luke Zettlemoyer. Scaling Semantic Parsers with On-the-fly Ontology Matching. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013.

[75] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289, 2001.

[76] Changki Lee, Yi-Gyu Hwang, Hyo-Jung Oh, Soojong Lim, Jeong Heo, Chung-Hee Lee, Hyeon-Jin Kim, Ji-Hyun Wang, and Myung-Gil Jang. Fine-grained named entity recognition using conditional random fields for question answering. *Information Retrieval Technology*, pages 581–587, 2006.

[77] Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, pages 1–54, 2013.

[78] Kenton Lee, Yoav Artzi, Jesse Dodge, and Luke Zettlemoyer. Context-dependent semantic parsing for time expressions. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[79] Hector J Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning*, 2012.

[80] Yang Li, Chi Wang, Fangqiu Han, Jiawei Han, Dan Roth, and Xifeng Yan. Mining evidences for named entity disambiguation. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1070–1078. ACM, 2013.

[81] Percy Liang. *Semi-supervised learning for natural language*. PhD thesis, Massachusetts Institute of Technology, 2005.

[82] Christopher H Lin, Mausam, and Daniel S Weld. Crowdsourcing control: Moving beyond multiple choice. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence*, pages 491–500, 2012.

[83] Christopher H Lin, Mausam, and Daniel S Weld. Reactive learning: Actively trading off larger noisier training sets against smaller cleaner ones. In *Active Learning Workshop & Workshop on Crowdsourcing and Machine Learning at ICML*, 2015.

[84] Xiao Ling, Sameer Singh, and Daniel S Weld. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 2015.

[85] Xiao Ling and Daniel S Weld. Temporal information extraction. *AAAI*, 10:1385–1390, 2010.

[86] Xiao Ling and Daniel S Weld. Fine-grained entity recognition. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, 2012.

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

[88] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

[89] Marie-Catherine De Marneffe, Bill Maccartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of the fifth international conference on Language Resources and Evaluation (LREC-2006)*, 2006.

[90] Cynthia Matuszek, John Cabral, Michael J Witbrock, and John DeOliveira. An introduction to the syntax and content of cyc. In *AAAI Spring Symposium: Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, pages 44–49, 2006.

[91] James Mayfield, Javier Artiles, and Hoa Trang Dang. Overview of the tac2012 knowledge base population track. *Text Analysis Conference (TAC 2012)*, 2012.

[92] Paul McNamee and Hoa Trang Dang. Overview of the tac 2009 knowledge base population track. In *Text Analysis Conference (TAC)*, 2009.

[93] Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8. ACM, 2011.

[94] Scott Miller, Jethran Guinness, and Alex Zamanian. Name tagging with word clusters and discriminative training. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL-2004)*, 2004.

[95] David Milne and Ian H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518. ACM, 2008.

[96] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL-2009)*, pages 1003–1011, 2009.

[97] Alexis Mitchell, Stephanie Strassel, Shudong Huang, and Ramez Zakhary. Ace 2004 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 2005.

[98] Tom M Mitchell, Justin Betteridge, Andrew Carlson, Estevam Hruschka, and Richard Wang. Populating the semantic web by macro-reading internet text. In *Proceedings of the 8th International Semantic Web Conference*, 2009.

[99] Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity linking meets word sense disambiguation: A unified approach. *Transactions of the Association for Computational Linguistics*, 2, 2014.

[100] David Nadeau. *Semi-Supervised Named Entity Recognition: Learning to Recognize 100 Entity Types with Little Supervision*. PhD thesis, University of Ottawa, November 2007.

[101] Ndapandula Nakashole, Tomasz Tylenda, and Gerhard Weikum. Fine-grained semantic typing of emerging entities. In *ACL (1)*, pages 1488–1497, 2013.

[102] Katsuma Narisawa, Yotaro Watanabe, Junta Mizuno, Naoaki Okazaki, and Kentaro Inui. Is a 204 cm man tall or small? acquisition of numerical common sense from the web. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 382–391, 2013.

[103] Vivi Nastase and Michael Strube. Decoding wikipedia categories for knowledge acquisition. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, volume 8, pages 1219–1224, 2008.

[104] Vivi Nastase, Michael Strube, Benjamin Börschinger, Cäcilia Zirn, and Anas Elghafari. Wikinet: A very large scale multi-lingual concept network. In *Proceedings of the 7th International Conference on Language Resources and Evaluation, Valletta, Malta*, pages 19–21, 2010.

[105] Roberto Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10, 2009.

[106] Joel Nothman, James R Curran, and Tara Murphy. Transforming wikipedia into named entity training data. In *Proceedings of the Australasian Language Technology Association Workshop*, pages 124–132, 2008.

[107] Xiaoman Pan, Taylor Cassidy, Ulf Hermjakob, Heng Ji, and Kevin Knight. Unsupervised entity linking with abstract meaning representation. In *NAACL*, 2015.

[108] Slav Petrov and Dan Klein. Improved inference for unlexicalized parsing. In *HLT-NAACL*, volume 7, pages 404–411, 2007.

[109] Glen Pink, Will Radford, Will Cannings, Andrew Naoum, Joel Nothman, Daniel Tse, and James R Curran. Sydney cmcrc at tac 2013. In *Proc. Text Analysis Conference (TAC2013)*, 2013.

[110] Hoifung Poon, Janara Christensen, Pedro Domingos, Oren Etzioni, Raphael Hoffmann, Chloe Kiddon, Thomas Lin, Xiao Ling, Alan Ritter, Stefan Schoenmackers, et al. Machine reading at the university of washington. In *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*, pages 87–95. Association for Computational Linguistics, 2010.

[111] Sindhu Raghavan, Raymond J Mooney, and Hyeonseo Ku. Learning to read between the lines using bayesian logic programs. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 349–358. Association for Computational Linguistics, 2012.

[112] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. Association for Computational Linguistics, 2009.

[113] Lev-Arie Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, volume 11, pages 1375–1384, 2011.

[114] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Proceedings of the 2010 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery*, pages 148–163, 2010.

[115] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 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*, 2013.

[116] Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. Named entity recognition in tweets: An experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

[117] Alan Ritter, Oren Etzioni, et al. A latent dirichlet allocation method for selectional preferences. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 424–434. Association for Computational Linguistics, 2010.

[118] Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1104–1112. ACM, 2012.

[119] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[120] Benjamin Roth, Tassilo Barth, Michael Wiegand, and Dietrich Klakow. A survey of noise reduction methods for distant supervision. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 73–78. ACM, 2013.

[121] Evan Sandhaus. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 2008.

[122] Satoshi Sekine. Extended named entity ontology with attribute information. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, 2008.

[123] Burr Settles. *Active Learning*, volume 18. Morgan & Claypool Publishers, 2011.

[124] Avirup Sil, Ernest Cronin, Penghai Nie, Yinfei Yang, Ana-Maria Popescu, and Alexander Yates. Linking named entities to any database. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 116–127. Association for Computational Linguistics, 2012.

[125] Avirup Sil and Alexander Yates. Re-ranking for joint named-entity recognition and linking. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2369–2374. ACM, 2013.

[126] Sameer Singh, Sebastian Riedel, Brian Martin, Jiaping Zheng, and Andrew McCallum. Joint inference of entities, relations, and coreference. In *CIKM Workshop on Automated Knowledge Base Construction (AKBC)*, 2013.

[127] Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. Wikilinks: A large-scale cross-document coreference corpus labeled via links to wikipedia. Technical report, Technical Report UM-CS-2012-015, 2012.

[128] Stephen Soderland and Wendy Lehnert. Wrap-up: a trainable discourse module for information extraction. *Journal of Artificial Intelligence Research*, 1994.

[129] Mohammad S Sorower, Janardhan R Doppa, Walker Orr, Prasad Tadepalli, Thomas G Dietterich, and Xiaoli Z Fern. Inverting grice's maxims to learn rules from natural language extractions. In *Advances in neural information processing systems*, pages 1053–1061, 2011.

[130] Marc Spaniol and Gerhard Weikum. Hyena: Hierarchical type classification for entity names. In *24th International Conference on Computational Linguistics*, page 1361, 2012.

[131] Robert Speer and Catherine Havasi. Representing general relational knowledge in conceptnet 5. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 3679–3686, 2012.

[132] Valentin I Spitkovsky and Angel X Chang. A cross-lingual dictionary for english wikipedia concepts. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 3168–3175, 2012.

[133] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A large ontology from wikipedia and wordnet. *Elsevier Journal of Web Semantics*, 6(3):203–217, 2008.

[134] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465. Association for Computational Linguistics, 2012.

[135] Hiroya Takamura and Junichi Tsujii. Estimating numerical attributes by bringing together fragmentary clues. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies*, 2015.

[136] Suzanne Tamang and Heng Ji. Relabeling distantly supervised training data for temporal knowledge base population. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 25–30. Association for Computational Linguistics, 2012.

[137] Erik F Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics, 2003.

[138] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. *Data Mining and Knowledge Discovery Handbook*, pages 667–685, 2010.

[139] Peter D Turney. Measuring semantic similarity by latent relational analysis. In *Proceedings of the 19th international joint conference on Artificial intelligence*, pages 1136–1141, 2005.

[140] Ralph Weischedel and Ada Brunstein. Bbn pronoun coreference and entity type corpus. 2005.

[141] Michael Wick, Sameer Singh, Harshal Pandya, and Andrew McCallum. A joint model for discovering and linking entities. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 67–72. ACM, 2013.

[142] Fei Wu and Daniel S. Weld. Autonomously semantifying wikipedia. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM-2007)*, pages 41–50, 2007.

[143] Wei Xu, Raphael Hoffmann, Le Zhao, and Ralph Grishman. Filling knowledge base gaps for distant supervision of relation extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 665–670, 2013.

[144] Limin Yao, Sebastian Riedel, and Andrew McCallum. Universal schema for entity type prediction. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 79–84. ACM, 2013.

[145] Congle Zhang, Stephen Soderland, and Daniel S Weld. Exploiting parallel news streams for unsupervised event extraction. *Transactions of the Association for Computational Linguistics*, 3:117–129, 2015.

[146] Min-Ling Zhang and Zhi-Hua Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.

[147] Wei Zhang, Yan Chuan Sim, Jian Su, and Chew Lim Tan. Entity linking with effective acronym expansion, instance selection and topic modeling. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three*, pages 1909–1914. AAAI Press, 2011.

[148] Jiaping Zheng, Luke Vilnis, Sameer Singh, Jinho D. Choi, and Andrew McCallum. Dynamic knowledge-base alignment for coreference resolution. In *Conference on Computational Natural Language Learning (CoNLL)*, 2013.

[149] Zhicheng Zheng, Fangtao Li, Minlie Huang, and Xiaoyan Zhu. Learning to link entities with knowledge base. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 483–491. Association for Computational Linguistics, 2010.

VITA

Xiao Ling grew up in Shanghai, China. He earned a Bachelor of Science degree in Computer Science from Shanghai Jiao Tong University in 2008, a Master of Science in Computer Science and Engineering from the University of Washington in 2010, and a Doctor of Philosophy in Computer Science and Engineering from the University of Washington in 2015.