Aspects of Coherence for Entity Analysis

Dissertation
zur Erlangung der Doktorwürde
der Neuphilologischen Fakultät
der Ruprecht-Karls-Universität Heidelberg

Vorgelegt von:
Benjamin Heinzerling

Referent: Prof. Dr. Michael Strube

Korreferentin: Prof. Dr. Anette Frank

Einreichung: 8. November 2018

Disputation: 22. Januar 2019

Abstract

Natural language understanding is an important topic in natural language processing. Given a text, a computer program should, at the very least, be able to understand what the text is about, and ideally also situate it in its extra-textual context and understand what purpose it serves. What exactly it means *to understand what a text is about* is an open question, but it is generally accepted that, at a minimum, understanding involves being able to answer questions like "Who did what to whom? Where? When? How? And Why?". Entity analysis, the computational analysis of entities mentioned in a text, aims to support answering the questions "Who?" and "Whom?" by identifying entities mentioned in a text. If the answers to "Where?" and "When?" are specific, named locations and events, entity analysis can also provide these answers. Entity analysis aims to answer these questions by performing entity linking, that is, linking mentions of entities to their corresponding entry in a knowledge base, coreference resolution, that is, identifying all mentions in a text that refer to the same entity, and entity typing, that is, assigning a label such as *Person* to mentions of entities.

In this thesis, we study how different aspects of coherence can be exploited to improve entity analysis. Our main contribution is a method that allows exploiting knowledge-rich, specific aspects of coherence, namely geographic, temporal, and entity type coherence. Geographic coherence expresses the intuition that entities mentioned in a text tend to be geographically close. Similarly, temporal coherence captures the intuition that entities mentioned in a text tend to be close in the temporal dimension. Entity type coherence is based in the observation that in a text about a certain topic, such as sports, the entities mentioned in it tend to have the same or related entity types, such as *sports team* or *athlete*. We show how to integrate features modeling these aspects of coherence into entity linking systems and establish their utility in extensive experiments covering different datasets and systems. Since entity linking often requires computationally expensive joint, global optimization, we propose a simple, but effective rule-based approach that enjoys some of the benefits of joint, global approaches, while avoiding some of their drawbacks. To enable convenient error analysis for system developers, we introduce a tool for visual analysis of entity linking system output. Investigating another aspect of coherence, namely the coherence between a predicate and its arguments, we devise a distributed model of selectional preferences and assess its impact on a neural coreference resolution system. Our final contribution examines how multilingual entity typing can be improved by incorporating subword information. We train and make publicly available subword embeddings in 275 languages and show their utility in a multilingual entity typing task.

Zusammenfassung

Automatisches Sprachverstehen ist ein wichtiger Teilbereich der natürlichen Sprachverarbeitung. Gegeben einen Eingabetext, sollte ein Computerpgrogramm verstehen, worum es in diesem Text geht und idealerweise darüberhinaus sogar warum und in welchem außertextlichen Zusammenhang er verfasst wurde. Auch wenn die Frage, was genau es heißt *einen Text zu verstehen* bisher nicht beantwortet ist, scheint es allgemein als Mindestvoraussetzung akzeptiert, in der Lage zu sein, Fragen wie "Wer hat was mit wem gemacht? Wo? Wann? Wie? Warum?" zu beantworten. *Entity analysis* hat zum Ziel, unter diesen Fragen auf das "Wer?" und das "Wem?" Antworten zu geben. Darüber hinaus kann entity analysis auch die Frage "Wo?" und "Wann?" beantworten, wenn es sich hierbei um konkret benennbare Orte und Ereignisse handelt, wie zum Beispiel Städte oder die erste Mondlandung. Entity analysis beinhaltet drei Unteraufgaben: Entity linking, Koreferenzresolution und entity typing. Ziel im entity linking ist es, Ausdrücke in einem Text, die Entitäten referenzieren zu finden und diese mit dem entsprechenden Eintrag in einer Wissensbasis zu verlinken. Auch die Koreferenzresolution findet solche Ausdrücke und gruppiert alle Ausdrücke, die dieselbe Entität referenzieren. Im entity typing wird jedem Ausdruck, der eine Entität referenziert, eine Klasse, wie zum Beispiel *Person*, zugewiesen.

In der vorliegenden Arbeit untersuchen wir, wie verschiedene Aspekte der Kohärenz verwendet werden können, um entity analysis zu verbessern. Unser Hauptbeitrag ist eine Methode, die es erlaubt mehrere spezifische Aspekte der Kohärenz zu verwenden, obwohl diese rechnerisch sehr aufwändig sind. Konkret führen wir hierfür geographische Kohärenz, temporale Kohärenz, und Entitätsklassenkohärenz ein. Geographische Kohärenz spiegelt die Intuition wieder, nach der Entitäten, die in einem Text erwähnt werden, tendenziell geographisch nahe beieinander liegen. In ähnlicher Weise basiert temporale Kohärenz auf der Beobachtung, dass Entitäten, die ein eimen Text erwähnt werden oft zeitlich nahe beieinander liegen. Entitätsklassenkohärenz drückt aus, dass Entitäten die einem Text über ein bestimmtes Thema wie zum Beispiel *Sport* erwähnt werden, oft denselben oder verwandten Entitätsklassen zugehören, wie zum Beispiel *Verein* oder *Sportler*. Wir demonstrieren,

wie man diese Aspekte der Kohärenz als Features in ein Entity-Linking-System integrieren kann, und dass diese Features zu besseren Ergebnissen in einer ausführlichen Evaluierung führen. Da entity linking oft rechnerisch aufwändige, sogennante *joint* und globale Optimierung benötigt, schlagen wir eine Methode vor, die einige, aber nicht alle Vorteile dieser Art von Optimierung genießt und einige ihrer Nachteile vermeidet. In der Koreferenzresolution untersuchen wir die semantische Übereinstimmung zwischem einem Prädikat und seinen Argumenten als einen weiteren Aspekt der Kohärenz. Hierzu entwicklen wir ein distributionelles Modell von Selektionspräferenzen und messen dessen Auswirkung auf die Qualität eines neuronales Koreferenzresolutionssystems. Schließlich untersuchen wir, wie entity typing durch sogenannte *subword*-Ansätze verbessert werden kann. Hierfür trainieren wir *subword embeddings* in 275 Sprachen und zeigen, dass diese entity typing in mehreren Sprachen verbessern.

Contents

1 Introduction 1

1.1 Thesis Overview . . . 1

1.2 Research Questions . . . 2

1.3 Contributions . . . 3

1.4 Published Work . . . 4

2 Background 5

2.1 Coherence . . . 5

2.2 Entity Analysis . . . 10

2.2.1 Terminology . . . 12

2.3 Entity Linking . . . 16

2.3.1 Entity Disambiguation by Local Inference . . . 19

2.3.2 Entity Disambiguation by Global Inference . . . 20

3 Aspects of Coherence in Entity Linking 23

3.1 Interleaved Multitasking for Entity Linking . . . 23

3.1.1 Interleaved Multitasking . . . 25

3.1.2 Sieves for Entity Linking . . . 28

3.1.3 Evaluation at the TAC 2015 Entity Discovery and Linking Shared Task . . . 34

Task specification . . . 34

Data . . . 35

Evaluation . . . 35

Implementation . . . 37

Results . . . 38

3.1.4 Limitations . . . 41

3.1.5 Summary . . . 43

3.2 Global Coherence in Entity Linking . . . 43

3.2.1 Exploiting Global Coherence without Global Inference . . . 43

3.2.2 Aspects of Global Coherence . . . 48

Geographic Coherence . . . 48

Temporal Coherence 49
Entity Type Coherence 51
Generic Semantic Relatedness 52
3.2.3 Pairwise Coherence Features 53
3.2.4 Local Features 53
3.2.5 Experiments 54
Data 54
Systems 55
3.2.6 Implementation and Experimental Setup 56
3.2.7 Results and Discussion 57
3.2.8 Candidate Reranking 59
3.2.9 Ablation Study 59
3.2.10 Automatic Verification on Noisy Text 60
3.2.11 Comparison of Automatic Verification to Related Methods . . 61
3.2.12 Summary 62
3.3 Visual Entity Explorer: A Tool for Analysis of Entity Linking Errors . 63
3.3.1 Motivation 63
3.3.2 The Visual Entity Explorer 64
3.3.3 Visualizing Entity Linking Errors 66
Partial Mention Detection Errors 66
Full Mention Detection Errors 66
Linking/Clustering Errors 66
3.3.4 Usage examples 67
3.3.5 Insights Gained from Error Analysis 69
3.3.6 Implementation 69
3.3.7 Design Decisions 70
3.3.8 Summary 72

4 Selectional Preferences for Coreference Resolution 73
4.1 Preference and Affordance 73
4.2 Selectional Preferences for Coreference Resolution 75
4.3 Modeling Selectional Preferences 78
4.3.1 Distributed Representation of Selectional Preferences 79
4.3.2 Enhanced++ Universal Dependencies 81
4.3.3 Fine-grained Entity Types. 83
4.4 Implementation 84
4.4.1 Qualitative Evaluation 87
4.5 Do Selectional Preferences Improve Coreference Resolution? 92

4.5.1 Limitations . . . 96
4.6 Summary . . . 96

5 Multilingual Entity Typing with Subword Units 99
5.1 Introduction . . . 99
5.1.1 Type Inventory . . . 100
5.1.2 Feature-based Entity Typing . . . 102
5.1.3 Neural Entity Typing . . . 104
5.1.4 The Unknown Word Problem in Entity Typing . . . 105
5.1.5 The Rare Word Problem in Entity Typing . . . 107
5.2 Subwords as a solution to the unknown and rare word problems . . . 107
5.2.1 Subword units . . . 110
Morphemes . . . 110
Characters . . . 110
Character n-grams . . . 111
FastText . . . 111
Byte Pair Encoding (BPE) . . . 112
5.3 `BPEmb` : Byte Pair Embeddings in 275 Languages . . . 116
5.3.1 Qualitative Evaluation . . . 117
5.4 Multilingual Entity Typing with Subword Units . . . 121
5.4.1 Evaluation: Comparison to `FastText` and Character Embeddings . . . 121
5.4.2 Experimental Setup . . . 121
Fine-grained Entity Typing . . . 121
Data . . . 121
Method . . . 124
Subword Units . . . 125
Composition Functions . . . 126
Hyper-parameter Search . . . 127
Evaluation . . . 127
5.4.3 Results and Discussion . . . 130
Comparison of Subword Units . . . 130
Comparison of Composition Functions . . . 130
Multilingual Results . . . 131
Limitations . . . 133
5.4.4 Summary . . . 134

6 Conclusions **135**
6.1 Outlook 136

A BPE model trained on English Wikipedia with 1000 merge operations **137**

B BPE model trained on German Wikipedia with 1000 merge operations **139**

C Most Similar Subwords for Various BPE Segmentations of *myxomatosis* **141**

D Code and Data Used in this Thesis **145**

Bibliography **147**

List of Figures

2.1 Interplay of different aspects of coherence 11
2.2 The semiotic triangle according to Ogden and Richards (1923) 12
2.3 The semiotic triangle in entity analysis 12
2.4 Relations involving BARACK_OBAMA in YAGO 13
2.6 Variability in entity mentions referring to BARACK_OBAMA 17
2.7 Candidate linking targets for the entity mention Obama obtained via the search function provided by the English edition of Wikipedia. . . 18
2.8 Complex example of global disambiguation in entity linking 22

3.1 Mention detection based on an already-linked mention 24
3.2 Sieve-based architecture for coreference resolution 26
3.3 Multitasking architectures for entity linking 27
3.4 Pivoting from already-linked mentions 32
3.6 Overview of the system implemented for the TAC 2015 shared task . 39
3.7 Comparison of our runs and best runs in the TAC 2015 enitity discovery and linking shared task 42
3.8 Overview of our entity linking verification system 44
3.9 An example of misleading generic coherence 45
3.10 Example showing a geographical outlier 46
3.11 Example showing a temporal outlier 50
3.12 Visualization of results on CoNLL and TAC15 test sets 58
3.13 Feature set ablations for the FL, TAC-1, and PH systems 60
3.14 Screenshot of the main display 65
3.15 Visualization of various mention detection and entity linking error types 67
3.16 Visualization showing a mention detection error and an annotation error 68
3.17 Visualization of an entity via its complete graph, by connecting entities in sequential order, and using an Euclidean minimum spanning tree 71

4.1 Dependency-based embedding model of selectional preferences . . . 79
4.2 Erk and Padó's Structured Vector Space model. Image source: Erk and Padó (2008). 80
4.3 Basic and enhanced++ universal dependency parse of (86) 82
4.4 Basic and enhanced++ universal dependency parse of (87) 82
4.5 The FIGER type inventory 83
4.6 Two instances from the Wikilinks corpus 84
4.7 Empirical entity type distribution for the subject argument of the predicate *sank* 86
4.8 Visualization of selectional preference embeddings 88
4.9 Zoomed view of selectional preference embeddings 89
4.10 Zoomed view of entity type embeddings 90
4.11 Selectional preference similarities of coreferent and non-coreferent mention pairs 92
4.12 The `deep-coref` mention-pair encoder and its position the `deep-coref` architecture 93

5.1 The Google Fine Types inventory 102
5.2 A typical neural entity typing architecture 104
5.3 Relative frequencies of words occurring in entity mentions 108
5.4 Zipfian distribution of words occurring in entity mentions 109
5.5 Text encoded with few, more, and many BPE merge operations . . . 115
5.6 BPE embeddings most similar to the subword *osis* 119
5.7 A Wikidata entry showing multilingual entity labels, which we use as entity mentions 122
5.8 Architecture of the entity typing model used in our experiments . . . 124
5.9 English entity typing performance of subword embeddings across different composition functions 129
5.10 Impact of the `BPEmb` embedding dimension on English entity typing 131
5.11 Impact of the number of BPE merge operations on English entity typing132

List of Tables

2.1 Candidate entities for the mention Jakarta 20
2.2 Simplified example of global disambiguation in entity linking 21

3.1 Trade-offs involved in the choice of a multitasking approach 25
3.2 Prior probabilities for the mention Barack Obama 29
3.3 Prior probabilities for the mention John Smith 29
3.4 Pairwise disambiguation decision 31
3.5 Monolingual English entity linking performance 40
3.6 Configuration of submitted runs 40
3.7 Freebase predicates for querying geo-coordinates 48
3.8 Freebase predicates for querying the *begin* and *end* of an entity's temporal range 51
3.9 Entity type distribution in a document about rugby 52
3.10 Knowledge base coverage of our proposed global coherence features 55
3.11 Results on CoNLL and TAC15 test sets 58
3.12 Comparison of filtering and candidate entity reranking performance on the CoNLL test set 59
3.13 Verification on different text genres 61
3.14 An example of entity linking system output 63
3.15 Example of evaluation script output 63

4.1 Examples of pairs of arguments and selectional preference slots . . . 85
4.2 Most similar terms for the queries *sink@nsubj*, *ship*... 91
4.3 Comparison of two ways of integrating our selectional preference model into `deep-coref` 94
4.4 Results on the CoNLL'12 test set 95
4.5 Changes in recall and precision errors on the CoNLL'12 test set . . . 95
4.6 Recall improvements over the baseline system 95

5.1 Ratio of out-of-knowledge-base entity mentions in common entity linking datasets 100
5.2 OOV words in entity mentions 106

5.3 BPE compression example 112
5.4 BPE segmentation example 113
5.5 BPE merge operations learned on English Wikipedia 113
5.6 Effect of the number of BPE merge operations on the beginning of the Japanese, Chinese, and English Wikipedia article TOYODA_STATION 114
5.7 Learned subword segmentations for different numbers of merge operations 118
5.8 Subwords most similar to *shire*, *ingen*, and *ose* 120
5.9 Randomly selected pairs of English entity mentions and entity types collected from Wikidata and Freebase 123
5.10 Overview of subword units and composition functions used in neural models proposed in the literature 126
5.11 Subword unit and composition function hyper-parameter spaces . . 128
5.12 Entity typing scores 132

C.1 Most similar subwords for each subword in different segmentations of the word *myxomatosis* in English and German 142

Chapter 1

Introduction

1.1 Thesis Overview

In this thesis, we study how different aspects of coherence can be exploited to improve entity analysis. After providing the necessary background on coherence and entity analysis in chapter 2, we turn to entity linking in chapter 3. Since entity linking often requires computationally expensive joint, global optimization, we propose a simple, but effective rule-based approach in section 3.1. This appraoch enjoys some of the benefits of joint, global approaches, while avoiding some of their drawbacks.

section 3.2 introduces our main contribution: A method that allows exploiting knowledge-rich, specific aspects of coherence, namely geographic, temporal, and entity type coherence. Geographic coherence expresses the intuition that entities mentioned in a text tend to be geographically close. Similarly, temporal coherence captures the intuition that entities mentioned in a text tend to be close in the temporal dimension. Entity type coherence is based on the observation that in a text about a certain topic, such as sports, the entities mentioned in it tend to have the same or related entity types, such as *sports team* or *athlete*. We show how to integrate features modeling these aspects of coherence into entity linking systems and establish their utility in extensive experiments covering different datasets and systems.

To enable convenient error analysis for system developers, we introduce a tool for visual analysis of entity linking system output in section 3.3. Investigating another aspect of coherence, namely the coherence between a predicate and its arguments, we devise a distributed model of selectional preferences and assess its impact on a neural coreference resolution system in chapter 4. Our final contribution, presented in chapter 5, examines how multilingual entity typing can be improved by incorporating subword information. We train and make publicly available subword embeddings in 275 languages, demonstrate their utility in multilingual entity typing, and compare them to alternative subword approaches.

1.2 Research Questions

An entity analysis system has to perform multiple interdependent tasks. The most common approaches towards solving many tasks are pipeline architectures in which tasks are performed in fixed sequential order, and joint multitasking, in which multiple tasks are performed simultaneously. Pipeline architectures have the disadvantage that early tasks do not have access to information produced by later stages in the pipeline. Furthermore, it is often not clear what the best task ordering is. Joint multitask approaches do not require task ordering and allow free sharing of information between tasks, but come at the cost of high computational complexity. This trade-off in multitask architectures leads us to our first research question: **How can we exploit interactions in a multitask setting without bearing the computational cost?**

Computational complexity also lies at the core of our second research question. The main problem when incorporating global measures of coherence into an entity linking system is the fact that maximizing such global measures is generally an NP-hard problem, making inference infeasible as the search space grows with the number of entities mentioned in the text to be analyzed. **How can we efficiently use global coherence measures in entity linking?**

Our third research question relates to error analysis of entity linking systems, since the complex structures arising in entity linking pose challenges for visualization tools. However, visualization is essential for developers who want to understand the errors their systems make. **What is a good method for clear and concise visualization of entity linking system output?**

Coherence does not only obtain globally between entities, but also locally, for example between an entity and the local context of its textual mention. Our fourth reaserach question is concerned with a particular type of this local coherence, the semantic agreement between a predicate and its argument. Predicate-argument structures have long been claimed to be important in coreference resolution, but experiments attempting to verify this claim were performed using, by today's standards, small amounts of data. **Does a modern model of predicate-argument structures improve coreference resolution quality?**

Our last research question deals with the third task in entity analysis, namely entity typing. Here coherence is at play in the regularities between entity names and entity types, but these regularities are difficult to capture for word-based approaches. Due to the high variability in names, many names are not contained in the entity typing system's vocabulary and hence are treated as unknown words. Various subword approaches have been proposed to tackle the unknown word problems in tasks such as machine translation. **Which subword approach is best for entity typing?**

1.3 Contributions

With the research presented in this thesis, we make the following contributions:

- We answer the first research question by introducing interleaved multitasking as a trade-off between pipeline architectures and joint multitasking. By treating each task not as monolithic, but instead splitting it into smaller groups of decisions, which are interleaved in free order, we enable a larger degree of information sharing between tasks than in the common pipeline architecture.

- As a solution to the second research question, we propose adding global coherence measures as a post-processing step for entity linking systems. This allows employing computationally expensive, knowledge-rich global coherence features, which would be infeasible in a global disambiguation setting. Furthermore, we propose global coherence features based on notions of entity type coherence, geographic coherence, and temporal coherence and show that they consistently improve entity linking performance.

- To answer the third research question, we develop a visualization tool for entity linking system output. By using minimal spanning trees to display entities, our tool ensures concise visualization of multiple entities mentioned possibly many times in long documents.

- Adressing our fourth research question, we create a selectional preference model with high coverage. To cope with data sparsity arising from the large variability in named entities, we generalize over named entities by using fine-grained entity typing. We then incorporate this model into a neural coreference resolution system to assess the utility of selectional preferences in coreference resolution.

- Answering the last research question, we perform a thorough comparison of different subword units. We find that FastText and Byte-Pair embeddings work best and make publicly available the multilingual subword embeddings trained for our experiments.

1.4 Published Work

The material presented herein is based on and extends published research papers first-authored by the author of this thesis. Unless noted otherwise, the material presented in the chapter or section in question is based on or extends the thesis author's contribution to the published work. The entity linking system presented in section 3.1 was published in Heinzerling, Judea, et al. (2015). The work on using global coherence for entity linking presented in section 3.2 was published in Heinzerling, Strube, and Lin (2017). The Visual Entity Explorer introduced in section 3.3 was published in Heinzerling and Strube (2015). The distributional model of selectional preferences presented in chapter 4 was published in Heinzerling, Moosavi, et al. (2017). The second author of Heinzerling, Moosavi, et al. (2017) incorporated this model into a neural coreference resolution system and performed the experiments and evaluations whose results are reported in section 4.5 in Table 4.3 on page 94, Table 4.4 on page 95, Table 4.5 on page 95, and Table 4.6 on page 95. The subword embeddings and entity typing experiments presented in chapter 5 were published in Heinzerling and Strube (2018).

Chapter 2

Background

2.1 Coherence

This thesis investigates how different aspects of coherence can be exploited to improve the computational analysis of entities mentioned in a text. To start with an example, consider the following sentence:

(1) Ilham Anas, a 40-year-old from Jakarta, Indonesia, works as Obama's doppelgänger. [1]

When a human being with sufficient background knowledge reads this sentence, she will use this knowledge to construct a plausible interpretation of its meaning. It is unlikely that the reader knows a person with the name *Ilham Anas*, but from her knowledge about the properties of names she will likely assume that the two words *Ilham* and *Anas* constitute a person name, even if she has never seen those two words before and not all person names consist of two capitalized words. This assumption is corroborated by the apposition

(2) a 40-year-old from Jakarta, Indonesia

since forty years is a common age in humans and humans are *from* somewhere. The preposition *from* indicates that what follows is likely a location, such as a city or country. Our reader knows that the city of Jakarta is the capital of the country of Indonesia and that it is common to refer to cities together with their containing state or country, such as *Berlin, Germany* or *Paris, Texas*, and infers that what follows *from* is indeed a location. Finally arriving at the predicate

(3) works as Obama's doppelgänger

the assumption that *Ilham Anas* refers to a person appears all but confirmed, since *working* is a typical activity of 40-year-old persons and working as someone's doppelgänger implies that both the look-alike and the – usually famous – original are

[1] Quoted from `https://www.businessinsider.com/barack-obama-doppelganger-indonesia-2015-2` (Accessed: 2018-10-07).

persons. This knowledge about doppelgängers also tells the reader that *Obama* likely refers to a famous person.

With a different predicate, for example

(4) is one of the finest whiskies Obama ever tasted

the reader's assumptions about the meaning of *Ilham Anas* would have turned out to be wrong:

(5) Ilham Anas, a 40-year-old from Jakarta, Indonesia, is one of the finest whiskies Obama has ever tasted.

Or, with a more plausible location:

(6) Ilham Anas, a 40-year-old from Orkney, Scotland, is one of the finest whiskies Obama has ever tasted.

Predicate (4) invalidates the interpretation built up to this point in examples (5) and (6), and prompts the reader to track back. Since it is plausible that a (in this case fictional) whisky has aged for forty years and the predicate asserts that the subject is a whisky, the most plausible interpretation is now that *Ilham Anas* is a whisky enjoyed by someone named *Obama*.

Predicate (4) prompted backtracking, since, in contrast to (3), it does not agree with the reader's experience and knowledge of how the world works. We call this experience and knowledge of the world, as well as the assumptions, inferences, and predictions she makes based on it, the reader's *world model*.

When reading a text, we expect that it is congruent with our world model and call agreement between a text, its interpration, and our world model *coherence*. Coherence is both a property of a text and of its interpretation. The interpretation of example (1) given above is coherent, since there is no violation of our world model. If instead, we claim that *Ilham Anas* refers to a whisky, we are left with an incoherent interpretation, since it does not agree with our world model: Given our experience and knowledge, there is no plausible scenario in which a beverage works as someone's doppelgänger. A text is coherent if it admits a coherent interpretation. Even though (5) and (6) violated the reader's initial assumptions, both sentences can be interpreted coherently and therefore are coherent. If the text does not allow a coherent interpretation it is incoherent:

(7) Ilham Anas, a 40-year-old whisky from Jakarta, Indonesia, works as Obama's doppelgänger.

Here, the apposition asserts that *Ilhman Anas* is a *whisky*, but the predicate asserts that *Ilham Anas* works as doppelgänger, for which no coherent interpretation is possible. The coherence of a text is a matter of degree that reflects the reader's ease or difficulty in finding a coherent interpretation.

(8) Ilham Anas, a 40-year-old from Berlin, Indonesia, works as Obama's doppelgänger.

The combination of *Berlin* and *Indonesia* appears less coherent than *Jakarta, Indonesia* since the reader's world model likely contains a fact like

(9) Berlin is in Germany.

Without detailed knowledge of the country, the reader might accept that there is a place called *Berlin* in Indonesia. Maybe a German ship sank off the coast of Indonesia and the survivors founded a town named after the German city. Assuming for a moment that this fiction is true, (8) can be made more coherent by resolving the incongruency with the reader's world model:

(10) Ilham Anas, a 40-year-old from Berlin, Indonesia, works as Obama's doppelgänger. Born in a village founded by 18th-century Prussian castaways that shares its name with the German capital, the unlikely look-alike developed an early interest in Western culture.

Coherence is not limited to predicates or geographical aspects. The sentence

(11) Ilham Anas, a 14th-century American monk, works as Obama's doppelgänger.

is temporally incoherent in several ways. If Ilham Anas lived in the 14th century he cannot be American in the common reading of the word, since the United States of America did not exist in the 14th century. A similar incoherence arises if we interpret *Obama* as a reference to former U.S. president BARACK_OBAMA. A third inconsistency with the reader's world model lies in the use of the present tense form *works* in combination with a subject that has presumably died several centuries ago.

Even though we appealed to world models in our introduction of coherence, we make no attempt to hypothesize about their precise nature. Instead, we only posit that they are constructed by perceiving objects in the world, finding patterns and connections between those objects, and then forming abstractions of those (perceptions of) objects and (perceptions of) patterns and connections. We call "an object in the world" *entity* and the abstraction of the perception of this object a *concept*. For example former U.S. president BARACK_OBAMA[2] is an entity, and his mental

[2] We denote entities with SMALL CAPS.

representation in the reader's world model is a concept. If a text is coherent, it will contain references to specific instances of concepts and connections between concepts that are known or inferrable by the reader. For example, the reader's world model might contain the following connection between the concept of a *Person*[3] and the concept of *Age*:

(12) A person has an age.

The apposition in example (1) is an instance of this generic connection:

(13) Ilham Anas, a 40-year-old

Or rephrased:

(14) Ilham Anas is 40 years old.

Similarly, the reader's world model might contain connections between the concept *Person* and the concept *Location* such as

(15) a. A person is born somewhere.
b. A person lives somewhere.

which find correspondence in

(16) Ilham Anas, [. . .] from Jakarta

with one or both of the common readings

(17) a. Ilham Anas was born in Jakarta.
b. Ilham Anas lives in Jakarta.

We call such instances of general patterns and connections involving one or more concepts in a world model *semantic relations*. Since semantic relations in her world model help making sense of the world, the reader looks for instances of those semantic relations when interpreting a text, as exemplified in the interpretation of (1). As a shorthand, we say that semantic relations apply to entities, e.g. that a certain semantic relations holds between two entities, instead of saying that the relation holds between the two concepts representing the entities in questions.

So far, we have seen semantic relations that hold between entities such as

(18) Jakarta is the capital of Indonesia.

and relations involving only an entity and a predicate, such as

(19) Ilham Anas works.

[3] We denote concepts using *Italic Small Caps*.

(20) Ilham Anas works as doppelgänger.

(21) Ilham Anas works as Obama's doppelgänger.

We will study relations between entities in more detail in chapter 3 and predications like (19) in chapter 4. A third kind of semantic relation, which we will pursue in chapter 5, is the connection between an entity and the way it is referred to. Regularities in names allow the reader to infer likely properties of the entity a name refers to. In example (1), we infered that *Ilham Anas* probably refers to a person, based on the regularity of person names: Person names often, but not always, consist of a capitalized first and last name. Such regularities are not limited to whether or not words form a person name. For example

(22) Ilham Anas & Sons Ltd

likely refers to a company since both *& Sons* and the abbreviation *Ltd* indicating a legal form both commonly occur in company names.

(23) Ilham Anas City

likely refers to a city since the word occurs in this name,

(24) Ilham Returns

likely to a movie or book, and

(25) Mary Anas

to a female person due to the first word being a common female given name.

That inferences based on such regularities are natural is illustrated by the perceived incoherence of the following sentences:

(26) Ilham Anas & Sons Ltd, a 40-year-old from Jakarta, Indonesia, works as Obama's doppelgänger.

(27) Ilham Anas City, a 40-year-old from Jakarta, Indonesia, works as Obama's doppelgänger.

(28) Ilham Returns, a 40-year-old from Jakarta, Indonesia, works as Obama's doppelgänger.

(29) Mary Anas, a 40-year-old from Jakarta, Indonesia, works as Obama's doppelgänger.

In (26), the company infered as referent of *Ilham Anas & Sons Ltd* is incoherent with being 40 years old and working as doppelgänger. The infered city in (27) and the movie or book apparently mentioned in (24) are similarly incoherent. In (29) an incoherence arises since *Mary Anas* likely refers to a female, while – with the obvious interpration of *Obama* – her doppelgänger is male. Further examples of the rich information contained in names are

(30) Mary VII

which likely refers to a monarch,

(31) Maryville

in which the suffix *-ville* suggests a location, and

(32) St. Mary's

which could plausibly refer to a hospital or a church.

Figure 2.1 on the facing page illustrates the aspects of coherence introduced so far. The existence of semantic relations relating the person ILHAM_ANAS to his age and work, the relation between the country of INDONESIA and its capital city JAKARTA, and the relation between BARACK_OBAMA and his *doppelgänger*, as well as the consistent use of names create coherence. Conversely, the fact that the interpretation is coherent – that under this interpretation all parts "fit well together" and assumptions about the meaning of each part mutually confirm each other – suggests that it is correct.

2.2 Entity Analysis

In this thesis, we will make use of the aspects of coherence introduced in the previous section in order to improve the automatic analysis of entities mentioned in a text. If understanding what a text is about involves answering questions such as "Who did what to whom, where, when, why, and how?" then entity analysis (Durrett and Klein, 2014) gives answers to the questions "Who?" and "Whom?". In case the location and time in question are associated with entities, entity analysis also answers the questions "Where?" and "When?". Entity analysis comprises three interdependent tasks: entity linking (introduced in section 2.3), coreference resolution (chapter 4), and entity typing (chapter 5).

For a computer program processing a text, say a news article, the text is nothing more than a sequence of characters. For example, consider (1) and its subsequent sentence:

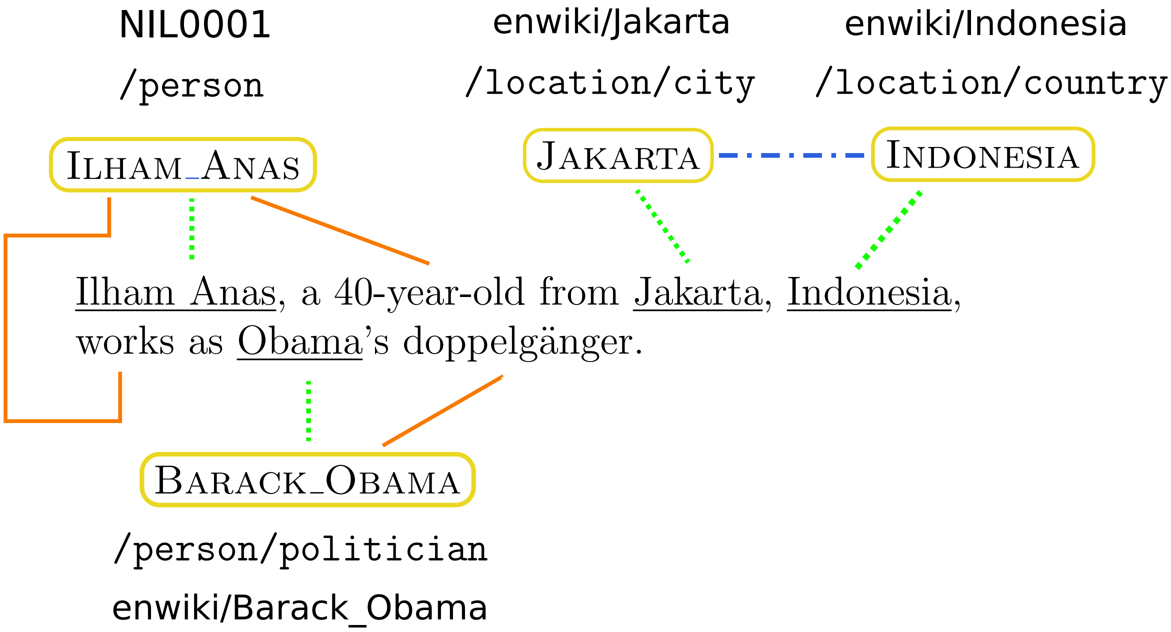


FIGURE 2.1: The interplay of different aspects of coherence gives rise to a particular interpretation of example (1). Entities are framed yellow and annotated with their entity type and knowledge base ID: either a NIL ID (Not In Lexicon) or its article title in the English edition of Wikipedia, enwiki. Entity types are introduced in chapter 5 and knowledge bases in subsection 2.2.1. Relations between entities shown as dot-dashed blue lines, relations between entities and context as solid orange lines, and relations between entities and their mentions as dotted green lines.

(33) Ilham Anas, a 40-year-old from Jakarta, Indonesia, works as Obama's doppelgänger. His uncanny resemblance to the US president allows him to travel the world.[4]

The program does not have any knowledge about the two persons, ILHAM_ANAS and BARACK_OBAMA, mentioned. Neither does it understand what a person is, what it means to be 40 years old, where Jakarta and Indonesia are, how the city and the country are related, or what *works as doppelganger* entails. Nor does it know that *His* and *him* both refer to the previously mentioned ILHAM_ANAS and *US president* to BARACK_OBAMA. A human reader, in contrast, understands the text easily. Part of this understanding involves identifying the targets of referential expressions in the text. Equipped with a world model, the reader is able to divide the sequence of characters into parts and recognize those parts as references to concepts in her world model. That is, the reference from the sequence of characters to the object is

[4] Quoted from `https://www.businessinsider.com/barack-obama-doppelganger-indonesia-2015-2` (Accessed: 2018-10-07).

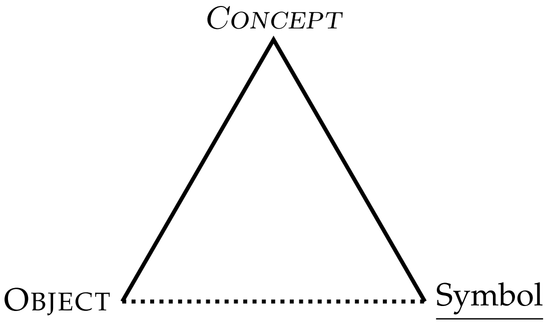


FIGURE 2.2: The semiotic triangle according to Ogden and Richards (1923). Adapted from Ogden and Richards (1923, p. 11), with our term OBJECT replacing the original term *referent* and CONCEPT replacing *Thought or reference*.

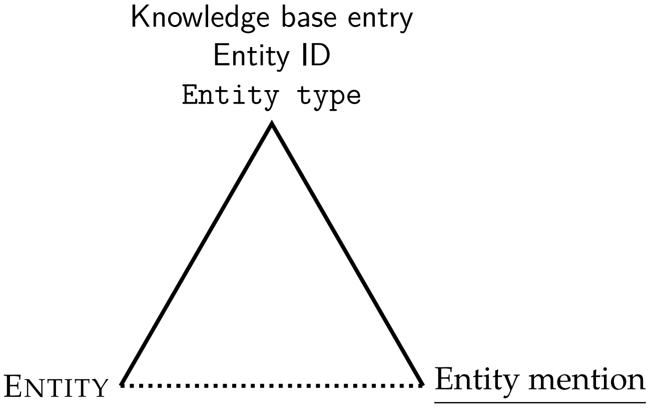


FIGURE 2.3: The semiotic triangle in entity analysis.

mediated by the reader's conception of this object. This triad of object, concept, and symbol is illustrated in the semiotic triangle (Figure 2.2).

A fundamental problem natural language understanding is that computer programs do not possess concepts. Entity analysis offers three surrogates (Figure 2.3): Entity linking provides a knowledge base entry with rich information about the entity in question (see the example entry shown in Figure 2.4 on the facing page). Coreference resolution allows propagating this information to coreferent mentions that could not be linked by the entity linking system, such as pronouns, and entity typing provides type information for entities that are not contained in the knowledge base.

2.2.1 Terminology

Before introducing entity linking, coreference resolution, and entity typing in the following sections, it may be helpful to define – or, in cases where no definition

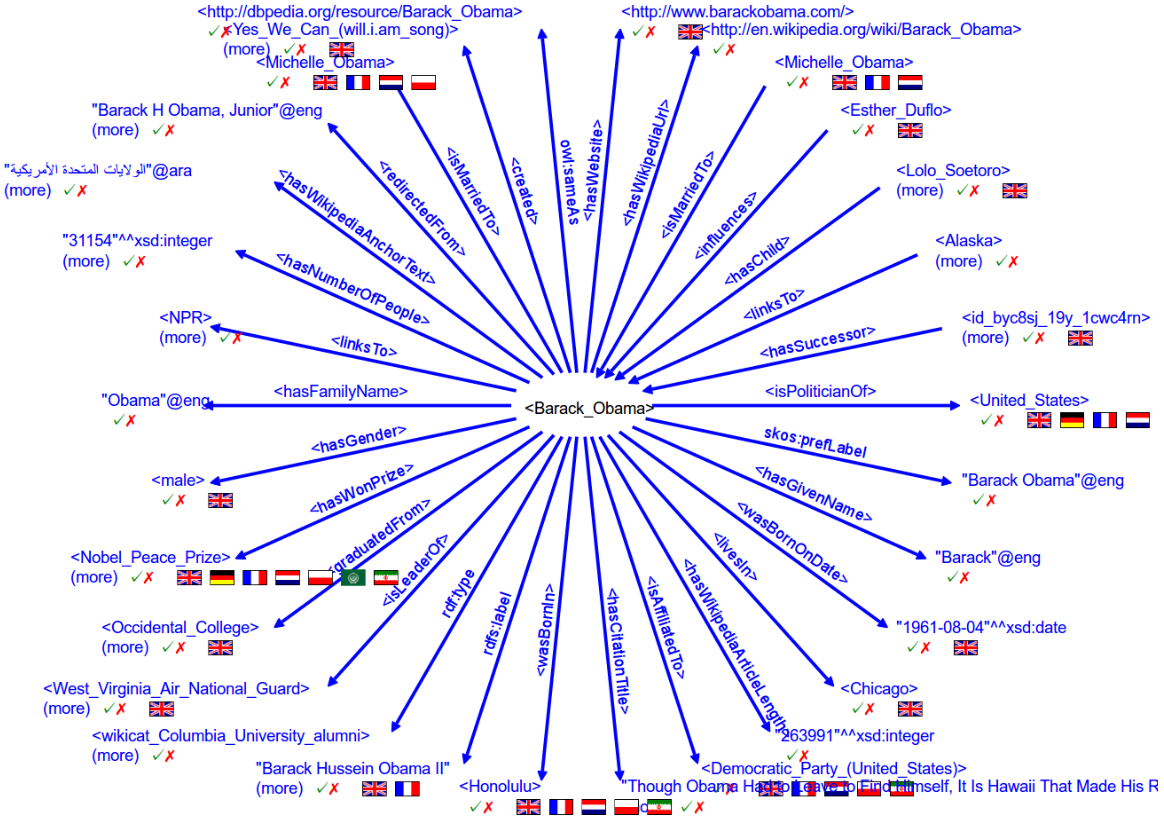


FIGURE 2.4: Relations involving BARACK_OBAMA in YAGO. Image source: Yago3 Browser.`https://gate.d5.mpi-inf.mpg.de/webyago3spotlx/SvgBrowser?entityIn=Barack_Obama`

appears possible, attempt to clarify – terms related to entity analysis.

Entity

Even though entities are the central object of study in several areas of natural language processing research, there exist no generally accepted definitions. When saying *entity* in this thesis, we roughly mean an object that exists in some form in the actual or a possible world. This includes animate and inanimate concrete objects, and, since we make no claims about what it means to *exist*, also abstract objects, concepts, and fictions.

The term *entity* appears with a variety of meanings in the natural language processing literature. In entity linking it refers to an entity in the sense used in this thesis, but also to the knowledge base entry representing a given entity. In coreference resolution, *entity* also denotes a cluster of coreferent mentions. In this thesis, entities are printed in SMALL CAPS.

Entity mention

A sequence of tokens in a text that refers to an entity. For brevity we may also simply say *mention*.

Named entity

An entity with a proper name. For example, EARTH is a named entity with the proper name *Earth*, while WATER in a context like the following is not.

(34) Seventy percent of the Earth's surface is covered by water.

Here, water is a an entity but not a named entity, since what is being referred to is water as a natural kind, and not a specific named instance of it. If, like the *mètre des Archives*, used as standard metre in the 19th century, a liter of water was kept in a vault in Paris and was known as the *litre des Archives*, then this particular volume of water would be a named entity. Canonical examples of named entities are persons, locations, and organizations.

Named entity mention

A rigid designator (Kripke, 1972) referring to a named entity. For example, in the sentence

(35) Barack Obama is the 44th President of the United States.

the proper name *Barack Obama* is a named entity mention, but the definite description *the 44th President of the United States* is not, even though both phrases refer to the same named entity BARACK OBAMA.

Somewhat confusingly, in the literature both the textual mention (*Barack Obama*) and the entity itself (BARACK_OBAMA) are referred to as named entities. The former usage is close in meaning to proper noun phrases and relevant when distinguishing named entity mentions (again, *Barack Obama*) from common noun mentions (e.g., *president*). The latter usage is relevant when distinguishing between entities with a proper name (BARACK_OBAMA) from those without one (WATER). In this thesis, we will mark named entity mentions with an underline:

(36) Seventy percent of the Earth's surface is covered by water.

(37) Barack Obama is the 44th President of the United States.

Knowledge base

A repository of structured information about entities. Throughout this thesis, we assume that an entity analysis system has access to a knowledge base for the domain and language in question. When discussing the use of knowledge bases, for example as a target for entity linking, we will say *the* knowledge base, even though implementations might employ multiple knowledge bases.

Corresponding entry in the knowledge base

The knowledge base entry that represents the entity an entity mention refers to. For instance, the Wikipedia entry enwiki/Barack_Obama is the corresponding knowledge base entry for the entity mention Barack Obama in (35). When the distinction does not matter, we will denote both entities and their corresponding entry in the knowledge base with small caps: BARACK_OBAMA may stand for both the entity as well as the Wikipedia entry with the ID enwiki/Barack_Obama.

Linkable mention

An entity mention that has a corresponding entry in the knowledge base

NIL (Not In Lexicon) entity

An entity that is not represented in the knowledge base.

NIL mention

An entity mention that is not linkable since it refers to a NIL entity.

2.3 Entity Linking

As Wikipedia grew rapidly since its launch in 2001, the online encyclopedia increasingly became subject of scholarly interest. After it was identified as a reliable knowledge source (Giles, 2005), pioneering work established its utility for disambiguating named entities (Bunescu and Paşca, 2006), for computing semantic relatedness between concepts (Strube and Ponzetto, 2006; Milne and Witten, 2008a) and for representing a document in terms of the concepts mentioned in it (Gabrilovich and Markovitch, 2007).

The idea of automatically linking entities mentioned in a text to their corresponding Wikipedia article was found to be particularly useful (Bunescu and Paşca, 2006; Cucerzan, 2007; Mihalcea and Csomai, 2007; Csomai and Mihalcea, 2008; Milne and Witten, 2008b). When expanded from not only linking entity mentions but all keywords in a text, this process is also known as *wikification* (Mihalcea and Csomai, 2007).

Most of the information contained in Wikipedia is unstructured or semistructured, making it difficult to process computationally. Structured knowledge bases, either derived from Wikipedia, such as DBpedia (Bizer et al., 2009) and YAGO (Suchanek et al., 2007; Hoffart, Suchanek, et al., 2011), or created from scratch, such as Freebase (Bollacker et al., 2008) solve this problem.

Entity linking is the task of automatically linking mentions of entities such as persons, locations, or organizations to their corresponding entry in a knowledge base, for example, Wikipedia. Figure 2.5 on the next page illustrates this task using our running example (1). Taking an unannoted, *raw* text like sentence (1) as input, an entity linking system needs to identify mentions of entities. In this example, the entity mentions are *Ilham Anas*, *Jakarta*, *Indonesia*, and *Obama*. This step is known as *mention detection*.

Having identified entity mentions, we are now faced with the fundamental problem in entity linking: the ambiguity of human language. A mention such as Obama is ambiguous, since it can refer to many different entities. In

(38) For Obama, fishing has long been the main industry. Increasingly, the Japanese port town is cashing in on the name it shares with the US president.

Obama refers to a Japanese port town, and in

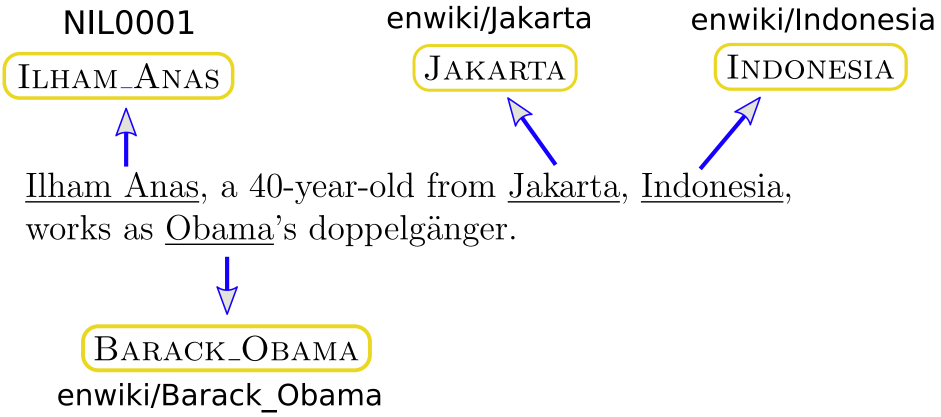


FIGURE 2.5: Entity linking examle showing detected entity mentions (underlined), entities (framed), and entity IDs (sans-serif).

Barack Obama Barack Obama II Barack Obama Jr. Barack Obama Junior Barack Obama, Jr. Barack Obama, Junior Barack Hussein Barack Hussein Obama Barack Hussein Obama II Barack Hussein Obama Jr. Barack Hussein Obama Junior Barack Hussein Obama, Jr. Barack Hussein Obama, Junior Barack Hussein obama Barack H. Obama Barack H. Obama II Barack H. Obama Jr. Barack H. Obama Junior B. H. Obama B. Hussein Obama B. Obama Pres. Obama President Barack H. Obama President Barack Hussein Obama II President Barack Obama President Obama Sen. Obama Senator Barack Obama US President Barack Obama United States President Barack Obama 2008 Democratic Presidential Nominee 44th President of the United States Barack Obana Barack Obbama Barack Ubama Barack OBama Barack obma BarackObama Barak Obamba Barck Obama Barock obama Borack Obama Borrack Obama Brack Obama Brock Obama Burack obama Hussein Obama Obamma 0bama Barack O'Bama O'Bama O'bama

FIGURE 2.6: Example of the variability in entity mentions referring to BARACK_OBAMA. Each phrase represents a redirect on Wikipedia that will send the visitor to the article about BARACK_OBAMA.

(39) Ms Obama

Obama likely refers to MICHELLE_OBAMA. In addition to ambiguity, variability of language poses a further challenge: An entity can be referred to in many different ways. Figure 2.6 shows some of these variations for the entity BARACK_OBAMA. We see variations of his canonical name, inclusions of titles, abbreviations, and different types of spelling mistakes.

For each detected entity mention, a typical entity linking system generates an ordered list of *candidate entities* the mention might refer to. This step is called *candidate generation*. The ordering reflects a prior belief about possible referents, without considering context. For example, we might assume that the word *Obama* in isolation refers to BARACK_OBAMA. In the final *candidate ranking* step, candidate entities are ranked based on both prior belief and contextual information. For instance, the

1. Barack Obama
2. Michelle Obama
3. Barack Obama Sr.
4. Ann Dunham (redirect from Obama's Mama)
5. United States presidential election, 2008 (redirect from Obama vs. McCain)
6. Family of Barack Obama
7. United States presidential election, 2012 (redirect from Obama vs. Romney)
8. Assassination threats against Barack Obama
9. Barack Obama citizenship conspiracy theories
10. Barack Obama presidential campaign, 2008
11. Barack Obama religion conspiracy theories
12. Early life and career of Barack Obama
13. Thanks Obama
14. Presidency of Barack Obama
15. Obama logo
16. United States presidential approval rating (redirect from Obama job approval)
17. Obama (disambiguation)
18. Obama Foundation
19. John McCain
20. Obama (surname)
21. Barack Obama "Hope" poster
22. American Recovery and Reinvestment Act of 2009 (redirect from Obama stimulus plan)
23. Death of Osama bin Laden (redirect from Death of Obama bin Laden)
24. 2009 Nobel Peace Prize (redirect from Award of the 2009 Nobel Peace Prize to Barack Obama)
25. Protests against Barack Obama

FIGURE 2.7: Candidate linking targets for the entity mention Obama obtained via the search function provided by the English edition of Wikipedia.

Japanese port town in the context of Obama in (38) allows an entity linking system to overcome the prior belief that Obama most likely refers to BARACK_OBAMA. Taken together, the candidate generation and candidate ranking steps *disambiguate* an entity mention.

A simple method for candidate generation is searching for matching strings in the knowledge base. Figure 2.7 shows the top 25 results of a search for the mention Obama in Wikipedia article titles and texts. In this case, the top-ranked result is *Barack Obama*, which corresponds to the Wikipedia article enwiki/Barack_Obama, which in turn corresponds to the correct entity BARACK_OBAMA. In other cases, the correct entity may be ranked lower, or not included in the list of top k candidates. Candidate generation has a large impact on the overall quality of an entity linking system (Hachey, Radford, et al., 2013). Considering only few candidates risks the correct entity not being among the top k, while a large number of candidates renders disambiguation more difficult and makes the ranking step slower.

Entity linking is complicated by the fact that the knowledge base generally does not contain corresponding entries for all entities mentioned in a text. The fraction of entities represented in the knowledge base, also known as the *coverage* of the knowledge base, depends on many factors, such as the overall size of the knowledge base, whether text and knowledge base are from the same domain, whether the text is more recent or older, and how often the knowledge base is updated. Entities mentioned in a text but not represented in the knowledge base are called *Not In Lexicon* or *NIL entities*. At this writing, there exists no entry for ILHAM_ANAS in the English edition of Wikipedia. A search returns the following results:

1. Ilham Aliyev
2. Abdul Rachman
3. Ilham Shahmuradov
4. ANAS Central Library of Science
5. Indonesia–Philippines relations

In cases such as this the entity linking system should recognize that none of the candidate entities is a good match and that the entity in question is a NIL entity. This step is known as *NIL classification*. Furthermore, the system may also be required to cluster all entity mentions that refer to the same NIL entity. This step is known as *NIL clustering*.

Approaches differ in whether they rank a mention's candidate entities independently of the candidate entities of other mentions or whether they rank all candidate entities of all mentions simultaneously by choosing an interpration that maximizes the overall coherence between all selected candidate entities. The first type of approach, called *local inference*, and the latter type, known as *global inference*, are introduced in the next sections.

2.3.1 Entity Disambiguation by Local Inference

When performing entity disambiguation by local inference, or *local disambiguation*, an entity linking system scores the candidate entities for a single entity mention according to how coherent they are with the mention's context. Taking our running example (1), suppose the entity linking system is disambiguating the mention Jakarta:

(40) Ilham Anas, a 40-year-old from Jakarta, Indonesia, works as Obama's doppelgänger.

Querying the knowledge base for candidate entities yields a list such as the one shown in Table 2.1 on the following page. In addition to the candidate entities themselves, the knowledge base – in this case Wikipedia – also provides information about these entities, such as glosses, longer descriptions, related entities, and relevant dates (see Figure 2.4 on page 13). Now, the key assumption in local

Entity	Wikipedia gloss
JAKARTA	the capital city of Indonesia
JAKARTA_PROJECT	a software project
JAKARTA_(BAND)	a former Yugoslav rock band
JAKARTA_(DJ)	an electronic music band known for the hit "One Desire"
JAKARTA_(MANGO)	a named mango cultivar from Florida
JAKARTA!	2012 novel by Christophe Dorigné-Thomson

TABLE 2.1: Candidate entities for the mention Jakarta found in the English edition of Wikipedia.

disambiguation is that matches or high similarity between this information in the knowledge base on the one hand and the entity mention's textual context on the other, serve as corroborating evidence for the hypothesis that the candidate entity in question is the one being referred to in the text. By this assumption, the string overlap between the Wikipedia gloss *the capital city of Indonesia* and mention's context, i.e. the string *Indonesia*, gives rise to the interpretation JAKARTA, since none of the glosses of the other candidate entities contains and context matches. In practice, entity linking systems employ more sophisticated methods to measure context overlap, for example measuring textual similarity with convolutional neural networks (Francis-Landau et al., 2016). The most important advantage of local disambiguation is its low compuational complexity. Since one disambiguation decision does not depend on other disambiguation decisions, computation complexity is linear in the number of mentions. The main drawback is that, by definition, local disambiguation cannot exploit coherence between entities, such as the coherence between INDONESIA and its capital JAKARTA in example (40).

2.3.2 Entity Disambiguation by Global Inference

Entity disambiguation by global inference, or *global disambiguation*, rests on the observation that coherence not only obtains between an entity and the (textual) context of its mention in a text, but also between the entity in question and other entities mentioned in the text. This coherence is realized through cohesive ties (Halliday and Hasan, 1976) such as the semantic relation expressed by the second apposition in (40):

(41) Jakarta, Indonesia

Assuming for ease of exposition that only the two mentions Jakarta and Indonesia have been detected, (40) becomes:

Candidate entity 1	Sem. relation	Candidate entity 2
JAKARTA	isCapitalOf	INDONESIA
JAKARTA	-	INDONESIA_(JAZZ ALBUM)
JAKARTA	-	INDONESIA_(BOOK)
JAKARTA_PROJECT	-	INDONESIA
JAKARTA_PROJECT	-	INDONESIA_(JAZZ_ALBUM)
JAKARTA_PROJECT	-	INDONESIA_(BOOK)
JAKARTA_(BAND)	-	INDONESIA
JAKARTA_(BAND)	-	INDONESIA_(JAZZ_ALBUM)
...	...	...

TABLE 2.2: Simplified example of global disambiguation in entity linking.

(42) Ilham Anas, a 40-year-old from Jakarta, Indonesia, works as Obama's doppelgänger.

An entity linking system performing global disambiguation now seeks to find an interpretation that exhibits maximal coherence. In case of only two mentions, this amounts to searching for a pair of candidate entities with maximal relatedness. A simple way of measuring the relatedness between two entities is querying the knowledge base for semantic relations. An example of this method is shown in table 2.2. We see that among the shown candidate entity pairs, only the correct interpretation is related via the predicate isCapitalOf. This example also illustrates a shortcoming of such a binary definition of relatedness: Symbolic relations such as isCapitalOf are sparse. For most pairs of entities, the knowledge base does not contain any relation that holds between them. However, there arguably exists a degree of relatedness between some of the pairs shown. For example, JAKARTA_(BAND) and INDONESIA_(MUSIC) are both related to music and hence to each other. A popular choice of relatedness measure that does not suffer from this shortcoming is the Milne-Witten distance (MWD) (Milne and Witten, 2008a). The MWD quantifies a generic notion of semantic relatedness between two entities in terms of links their corresponding Wikipedia articles share:

$$MWD(a,b) = \frac{log(max(|A|,|B|)) - log(|A \cap B|)}{log(|W|) - log(min(|A|,|B|))}$$

where a, b are Wikipedia articles, A is the set of Wikipedia articles linking to a, B the set of Wikipedia articles linking to b. This distance becomes smaller, i.e., relatedness increases, as the number of shared links $|A \cap B|$ grows larger.

`WikiRelate!` (Strube and Ponzetto, 2006) computes semantic relatedness using

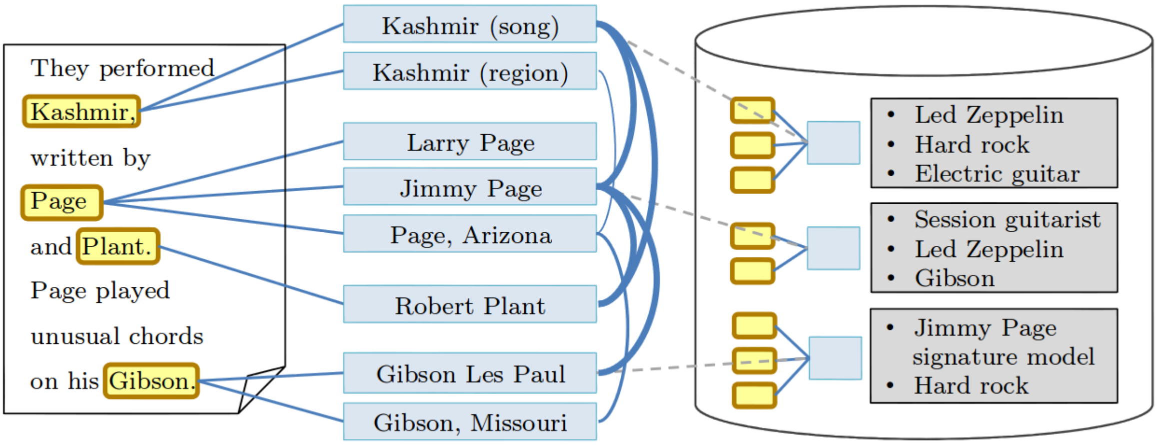


FIGURE 2.8: Complex example of global disambiguation in entity linking. Image source: Hoffart, Yosef, et al. (2011).

Wikipedia pages and the Wikipedia category tree. An alternative that does not rely on links between Wikipedia articles or Wikipedia categories is keyphrase overlap relatedness (KORE) (Hoffart, Seufert, et al., 2012). KORE first collects keyphrases associated with each of the two entities in question and then expresses their relatedness in terms of overlap between between the two sets of keyphrases. (Moro et al., 2014) propose a graph-based relatedness measure which applies not only to pairs of entities, but to all candidate entities in a particular interpretation.

A more complex example of global disambugation, due to Hoffart, Yosef, et al. (2011), is shown in Figure 2.8 . Here, the four entity mentions Kashmir, Page, Plant, and Gibson and their candidate entities are represented as graph with one edge between each mention and its respective candidate entities, and edges between candidate entity nodes, weighted according the relatedness of the respective candidate entity pair. Finding a maximally coherent interpretation is equivalent to finding a dense subgraph in this weighted graph. This is an NP-hard problem, which makes such an approach to global disambiguation infeasible for large graphs, that is, for long documents containing many entity mentions.

We have distinguished entity linking systems employing local disambiguation from those performing global disambiguation. Another distinction is concerned with the fashion in which the subtasks involved in entity linking are performed. Pipeline architectures perform each subtask in a fixed, sequential order, while joint multitasking approaches perform two or more subtasks simultaneously. In section 3.1 we will introduce an approach that combines some of the benefits of both pipeline and joint multitasking architectures, while avoiding some of their drawbacks.

Chapter 3

Aspects of Coherence in Entity Linking

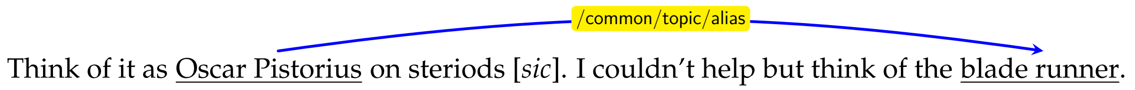
3.1 Interleaved Multitasking for Entity Linking

In this section, we introduce the first contribution of this thesis: an approach to multitasking for entity linking which enjoys some of the benefits of joint multitasking and avoids some of its drawbacks. Recall that entity linking comprises several subtasks:

1. Tokenization, part-of-speech tagging, and other preprocessing;
2. Named entity recognition
3. Additional mention detection, e.g. nominal mentions, depending on the specific task setting
4. Mention linking and NIL classification
5. NIL clustering
6. Post-processing

These subtasks are commonly performed sequentially in a pipeline architecture. An example of a typical entity linking pipeline is shown in the left part of Figure 3.3 on page 27.

Pipeline architectures have well-known advantages and disadvantages (Roth and Yih, 2004; Marciniak and Strube, 2005). While the idea of performing subtasks in order of necessity may appear simple at first sight, it is often not clear what the best order is. For example, the text shown in Figure 3.1 on the following page contains the entity mention blade runner, which is difficult to detect with high confidence by standard mention detection techniques due to its non-standard



Think of it as Oscar Pistorius on steriods [*sic*]. I couldn't help but think of the blade runner.

FIGURE 3.1: Example showing mention detection based on another entity mention that has already been linked to the knowledge base.

lowercase spelling.[1] The preceding sentence in this example contains the entity mention Oscar Pistorius, which is both easy to detect for any named entity recognizer and easy to link with high confidence to the correct knowledge base entry OSCAR_PISTORIUS, due to the low ambiguity of this particular entity mention.[2] Querying the knowledge base for information related to OSCAR_PISTORIUS, a former sprint runner and Paralympian, reveals that he has the alias *Blade Runner*, a nickname alluding to his use of prosthetic running blades. Armed with this knowledge, we are now able to detect the mention blade runner more easily and with higher confidence. In the following, we will call this kind of mutually beneficial relation between tasks or subtasks *task interaction*.

In the example above, we saw an interaction between mention detection and entity linking. In a pipeline, such an interaction between tasks is only possible in one direction, from tasks performed earlier in the pipeline to subsequent *downstream* tasks. For example, in the pipeline architecture outlined above, the entity linking subtask can make use of annotations produced in earlier stages, such as part-of-speech tags, detected entity mentions, or entity type annotations. In contrast, earlier stages cannot exploit information that only becomes available at later stages, such as the knowledge about Oscar Pistorius' alias, which would support detecting *blade runner* as an entity mention.

Recognizing the need to enable free interaction between multiple tasks, Roth and Yih (2004) proposed to jointly optimize multiple tasks by formulating the problem as an Integer Linear Program (Schrijver, 1998). A hypothetical entity linking system performing three key subtasks jointly is displayed in Figure 3.3 on page 27 (right). This system remains hypothetical, as in practice, joint optimization does not scale well in the number of joint tasks or the length of the given text. Furthermore, in joint optimization of multiple tasks the interactions between tasks have to explicitly modeled, for example by introducing pairwise constraints into a Integer Linear Program. As these pairwise interactions are required for all interacting pairs of

[1] It is, of course, possible to detect blade runner as a potential mention by considering all noun phrases or applying any other high-recall method. However, the resulting large number of potential mentions hurts entity linking precision.

[2] We say that a mention has *low ambiguity* if it refers to the same entity in almost all cases. Also see the discussion of almost unambiguous mentions in subsection 3.1.2.

	Pipeline	Joint multitasking
task ordering	difficult	not necessary
task interaction	weak	strong
speed	fast	slow
extensibility	easy	difficult

TABLE 3.1: Trade-offs involved in the choice of a multitasking approach.

tasks, extending a joint optimization architecture with an additional tasks becomes more difficult as the number of tasks increases. These difficulties explain why joint subtask approaches in entity linking have been limited to two or three tasks. Fahrni and Strube (2012) perform joint linking and NIL clustering using Markov Logic Networks (Domingos and Lowd, 2009), while G. Luo et al. (2015) perform joint mention detection and linking.

The trade-offs involved in choosing between pipeline and joint multitasking architectures are summarized in Table 3.1 . We now introduce our approach, interleaved multitasking, which avoids some of these trade-offs.

3.1.1 Interleaved Multitasking

The starting point for our proposed multitasking architecture is the sieve-based approach to coreference resolution (Raghunathan et al., 2010; H. Lee, Peirsman, et al., 2011; H. Lee, A. Chang, et al., 2013), shown in Figure 3.2 on the next page. This approach implements a series of deterministic rules, called *sieves*, which operate on clusters of coreferent mentions and are applied in order of decreasing precision. A sieve merges two clusters of mentions that have been determined to refer to the same entities into a new cluster if the criteria of the sieve are met. For example, the *Speaker Identification* sieve merges first-person pronoun mentions in direct speech with the speaker mention. The precision of a sieve expresses how often applying this sieve gives a correct coreference result. For example, merging mentions with matching strings is often correct, but will be wrong if there are, two different persons named “John” mentioned in a text.

After a high-recall mention detection stage, the sieves are applied in order of decreasing precision, i.e. the sieves with highest precision first, and those with lower precision later. This order aims to minimize the risk of making costly early mistakes. Since each sieve bases its decisions on the mention clusters created by earlier sieves, early mistakes have a disproportionate impact as they propagate through

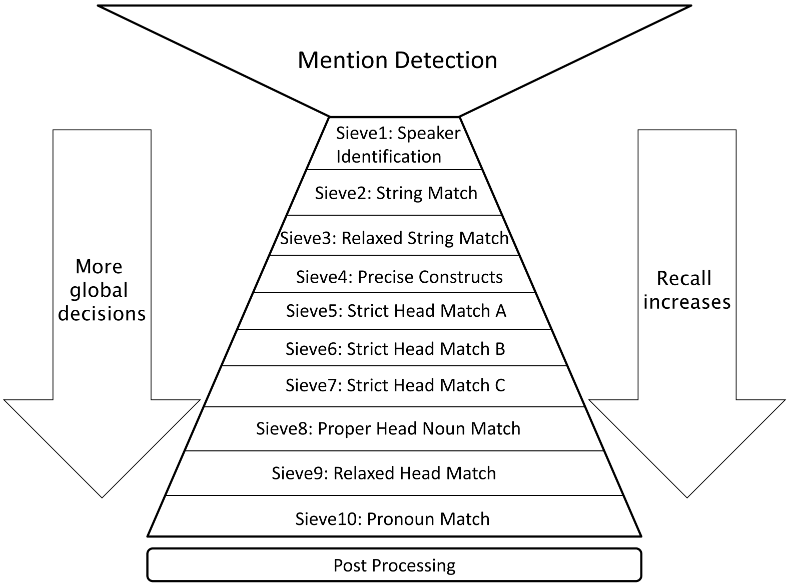

Mention Detection
Sieve1: Speaker Identification
Sieve2: String Match
Sieve3: Relaxed String Match
Sieve4: Precise Constructs
Sieve5: Strict Head Match A
Sieve6: Strict Head Match B
Sieve7: Strict Head Match C
Sieve8: Proper Head Noun Match
Sieve9: Relaxed Head Match
Sieve10: Pronoun Match
Post Processing
More global decisions
Recall increases

FIGURE 3.2: Sieve-based architecture for coreference resolution. Image source: H. Lee, A. Chang, et al. (2013).

more stages of the of the pipeline than later mistakes. Conversely, mistakes committed in later stages of the pipelines tend to have a smaller impact on the final result, since there are fewer stages left during which errors can propagate. This observation underlies the choice to apply high-precision, low-recall sieves first, and low-precision, high-recall sieves towards the end of the pipeline.

One limitation of H. Lee, A. Chang, et al.'s approach is the fact that sieves are applied only for a single task, namely, merging coreferent mention clusters. As we have seen above, entity linking comprises different subtasks. Aiming to exploit the interactions between these subtasks, we extend the sieve approach to perform sieve-based joint multitasking. Instead of applying sieves only to one task, we formulate sieves for many tasks and, like in the original approach, apply them in order of decreasing precision, but, crucially, switch freely between tasks as necessary, as outlined in Figure 3.3 on the facing page (middle). By doing so, we aim to strike a balance between a pipeline architecture, in which each task is performed sequentially, and joint multitasking, in which all tasks are optimized jointly.

Our approach, which we call *interleaved multitasking*, can also be thought of as splitting up monolithic tasks in a pipeline into smaller groups of task-specific decisions, ordering those groups by precision, and then interleaving them back into a pipeline.

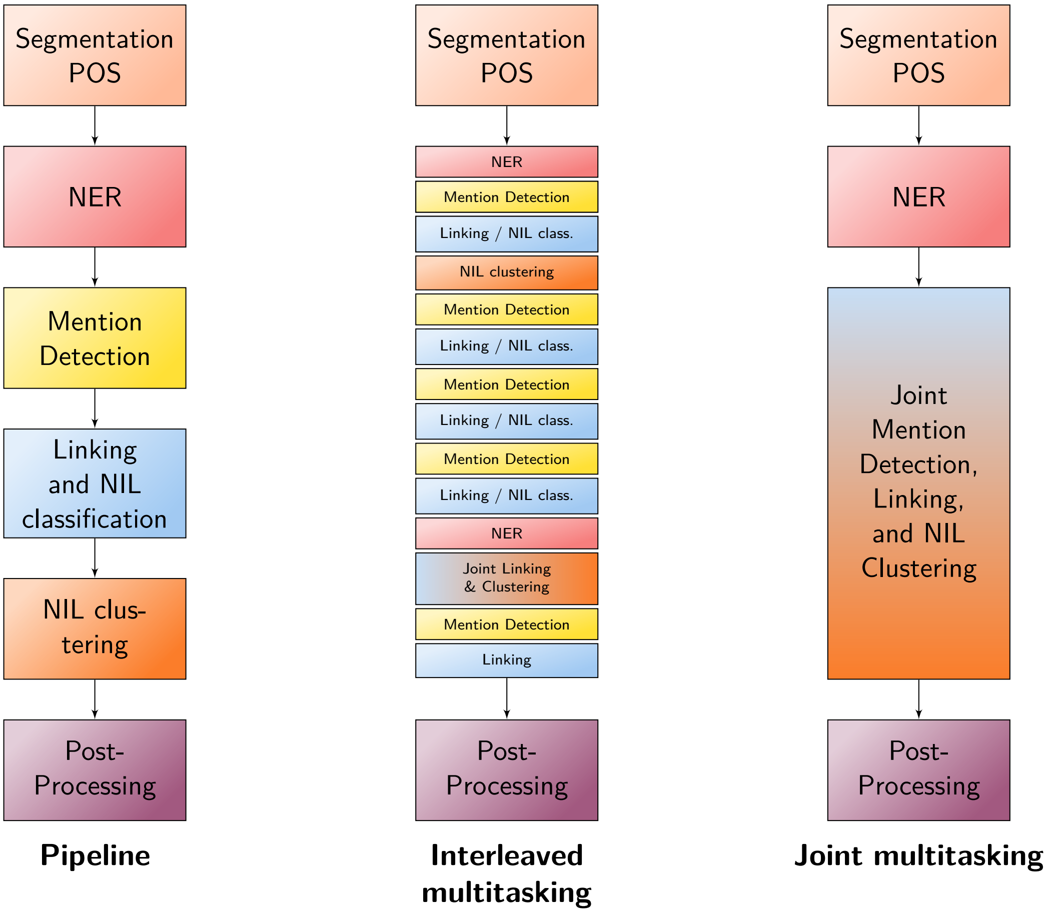


FIGURE 3.3: Multitasking architectures for entity linking.

3.1.2 Sieves for Entity Linking

We now describe the sieves we propose for entity linking, in order of decreasing precision.

High-precision mention detection and linking

Aiming to minimize the risk of committing early mistakes, the design goal of the first sieves is to detect and link "easy" entity mentions that can be linked to entries in the knowledge base with high precision. Querying the knowledge base for information related to these entries will then provide contextual for downstream sieves.

Concretely, we first run a standard preprocessing pipeline consisting of a tokenizer, part-of-speech tagger, and named entity recognition annotation. Then we apply the following three sieves to perform high-precision mention detection and linking:

Almost unambiguous mention sieve. Among the recognized named entities, we identify *almost unambiguous* mentions, that is, mentions that almost always refer to the same entity. For example, according to an estimate from a large corpus, the mention <u>Barack Obama</u> refers to BARACK_OBAMA with a probability of over 98 percent (Table 3.2 on the next page). Unfortunately, not all mentions are almost unambiguous – some mentions may exhibit much higher ambiguity (see Table 3.3 on the facing page for an example). For this sieve, we employ CrossWikis (Spitkovsky and A. X. Chang, 2012), a resource providing probabilities that a given string links to a certain Wikipedia article. Since these probabilities are estimated on the basis of a large background corpus, they express a default or *prior* belief about the likely referent of a given mention in the absence of context, which is updated as contextual information becomes available.[3]

Based on experiments on a development set, we set the threshold for *almost unambiguous* as having a prior probability of 95 percent or higher, as estimated by CrossWikis. This first sieve, of course, is not perfect. Aiming to undo some of the mistakes made by this sieve, we apply two filters next.

Entity type mismatch filter. This sieve first queries the entity types of all entities linked by the previous sieve. These entity types, obtained from entity linking, are then compared to the entity types annotated by the named entity recognizer during preprocessing. Viewing agreement as positive evidence that both entity linking and named entity recognition decisions were correct, and disagreement as evidence

[3] CrossWikis provides empirical prior probabilities estimated on Wikipedia and a Google-internal web crawl.

Mention	Prior (%)	Wikipedia article
Barack Obama	98.73	Barack_Obama
Barack Obama	0.25	2009_Nobel_Peace_Prize
Barack Obama	0.21	List_of_characters_in_the_Mortal_Kombat_series
Barack Obama	0.17	Barack_Obama_presidential_campaign,_2008
Barack Obama	0.10	Barack_Obama,_Sr.
Barack Obama	0.06	The_Audacity_of_Hope
Barack Obama	0.05	Political_positions_of_Barack_Obama
Barack Obama	0.03	List_of_judicial_appointments_made_by_Barack_Obama
Barack Obama	0.03	Barack_Obama_citizenship_conspiracy_theories
Barack Obama	0.03	Barney_Frank
Barack Obama	0.02	Madelyn_and_Stanley_Dunham
Barack Obama	0.02	Barack_Obama_(comic_character)
Barack Obama	0.02	United_States_Senate_career_of_Barack_Obama
Barack Obama	0.02	Electoral_history_of_Barack_Obama
Barack Obama	0.02	Native_born_Americans

Table 3.2: Prior probabilities for the mention Barack Obama listed in CrossWikis. We say that this mention is almost unambiguous, since almost all occurrences in the CrossWikis background corpus refer to the Wikipedia article Barack_Obama.

Mention	Prior (%)	Wikipedia article
John Smith	26.84	John_Smith_of_Jamestown
John Smith	19.38	John_Smith_(UK_politician)
John Smith	14.46	John_Smith
John Smith	07.58	John_Smith_(comics)
John Smith	03.03	John_Smith_(name)
John Smith	02.83	John_Smith_(wrestler)
John Smith	02.82	John_Smith_(uncle_of_Joseph_Smith,_Jr.)
John Smith	01.77	Pocahontas_(1995_film)
John Smith	01.39	John_Smith_(New_York)
John Smith	01.18	John_Smith_(Chancellor_of_the_Exchequer)
John Smith	01.05	John_Smith_(actor)
John Smith	00.86	John_Smith_(athlete)
John Smith	00.84	John_Smith_(Wendover_MP)
John Smith	00.75	John_Smith_(Welsh_politician)
John Smith	00.72	John_Smith_(nephew_of_Joseph_Smith,_Jr.)

Table 3.3: Prior probabilities for the mention John Smith. Unlike Barack Obama, this mention is not almost unambiguous since the probability mass is distributed more equally among several entities.

to the contrary, we remove all entity links created by the previous sieve whose corresponding entity types do not match. This filtering is applied to linked named entities of type `person`, `location`, and `organization`.

Mention-surface mismatch filter. This sieve queries all known surfaces forms for all entities linked by previous sieves. Known surface forms are canonical names, corresponding to the Freebase predicate rdfs:label, as well as name variations, nicknames, and aliases, corresponding to the Freebase predicate /common/topic/alias. Next, the sieve compares each linked entity mention with the known surface forms of its referent entity and removes the entity link if there is no matching surface form.

The two filters remove erroneous links introduced by previous sieves, at the cost of also removing some correct links. Verifying that this first group of three sieves indeed yields high-precision entity links, we observe a linking precision of 90 percent on our development set (see subsection 3.1.3 for dataset and metrics). The linked entities found by this group of high-precision sieves provide context which will enable better linking decisions by the following sieves.

Path-based mention detection and linking

Recall that in entity linking, the notion of coherence is operationalized in the form of relatedness scores, which quantify how semantically related two entities are. Under the assumption that a more coherent interpretation is more likely to be correct than a less coherent one, ambiguous entity mentions in a text can be disambiguated by taking the interpretation with the maximum overall relatedness score. The overall relatedness score, in turn, can be maximized by choosing candidate entities that are maximally related to all other entities in the text (Kulkarni et al., 2009; Hoffart, Yosef, et al., 2011; Moro et al., 2014).

As a simple example, consider the following text:

(43) Nigeria became Africa's largest economy earlier this year. But farmers in the town of Daura have yet to [...].

Let us further simplify by assuming only two candidate entities for the mention Nigeria:

- NIGERIA, a country in Africa; and
- NIGERIA_(JAZZ_ALBUM), an album by jazz guitarist Grant Green released in 1980;

and two candidate entities for the mention Daura:

- PIERRE_DAURA, a Catalan artist; and

Subject candidate entity	Predicate	Object candidate entity
NIGERIA	-	PIERRE_DAURA
NIGERIA_(JAZZ_ALBUM)	-	PIERRE_DAURA
NIGERIA	/location/contains	DAURA_(NIGERIA)
NIGERIA_(JAZZ_ALBUM)	-	DAURA_(NIGERIA)

TABLE 3.4: A pairwise disambiguation decision supported by the existence of a semantic relation between two candidate entities.

- DAURA_(NIGERIA), a town in Nigeria.

This gives $2 \times 2 = 4$ possible interpretations which are shown in Table 3.4 . The maximum relatedness approach scores the relatedness of each interpretation, in this simplified case by checking each pair of candidate entities whether a relation exists. Then, it chooses the interpretation with the highest score, in this case NIGERIA and DAURA_(NIGERIA).

A drawback of this approach is that querying the knowledge base for information regarding the relatedness of two candidate entities, e.g. whether there exists a path connecting one candidate entity to the other, is computationally expensive.

Our sieve-based approach allows an alternative way of exploiting semantic relatedness between entities. As described above, the first sieves in our pipeline perform high-precision entity linking. Using those already-linked mentions, we query the knowledge base for related entities, and then check if these related entities are mentioned in the text (Figure 3.4 on the following page). In other words, instead of querying the knowledge base for paths connecting a subject candidate entity and an object candidate entity, we locate the subject entity in the knowledge base graph, follow specific paths to arrive at related object candidate entities, and then check if any object candidate entity is mentioned in the text by string-matching all known names contained in the knowledge base.

Previous work has made extensive use of semantic relations for re-ranking candidate entities of given mentions (Hoffart, Yosef, et al., 2011; Moro et al., 2014). In contrast to these approaches, we use mentions linked by upstream sieves as a pivot for detecting and link semantically related mentions. More concretely, this is done by compiling a list of paths in the knowledge base graph that connect two entity types of interest, following these paths for each linked mention of matching entity type to find related entities, and then checking if known surface forms of any of the related entities occur in the text.

To find related entities, we use the following list of paths:

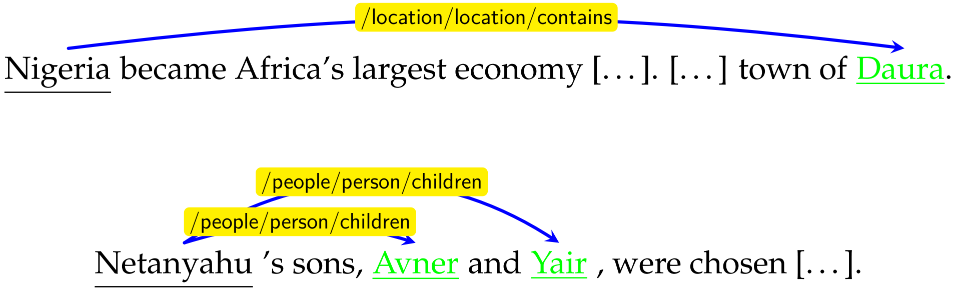

/location/location/contains
Nigeria became Africa's largest economy [. . .]. [. . .] town of Daura.
/people/person/children
/people/person/children
Netanyahu 's sons, Avner and Yair , were chosen [. . .].

FIGURE 3.4: Pivoting from entity mentions that have already been linked to mentions of related entities. In the top example, *Nigeria* has been correctly linked by an earlier high-precision sieve. Querying the knowledge base for related entities, we find that *Daura* matches the object of the relation (NIGERIA, /location/location/contains, DAURA) and link this mention to DAURA. In the bottom example, the knowledge base contains entries for AVNER_NETANYAHU and YAIR_NETANYAHU, who are children of the already linked entity BENJAMIN_NETANYAHU and whose names partially match the entity mentions *Avner* and *Yair*.

- **Children.** The given entity's children. For example, AVNER_NETANYAHU and YAIR_NETANYAHU are listed as children of BENJAMIN_NETANYAHU in Freebase. This relation corresponds to the Freebase predicate /people/person/children.

- **Geographical containment.** The locations contained by a location. For example NIGERIA contains the town of DAURU. In Freebase and YAGO, this relations corresponds to taking the transitive closure of the predicate /location/location/contains and <isLocatedIn > respectively.[4]

- **Party affiliation.** The political party to which a politician belongs to. For example, BARACK_OBAMA's party affiliation is DEMOCRATIC_PARTY_(U.S.). This path applies to all known politicians, that is, all already-linked mentions who have a corresponding /people/person/profession value. Having identified a mention of a politician in the text, we query related entities in Freebase using the path government/politician/party → /government/political_party_tenure/party, which connects politicians to political parties via a compound value type.[5]

[4] We use YAGO in addition to Freebase for this sieve, since we found that YAGO offers a richer inventory of geographical information than Freebase, presumably due to its inclusion of the `https://www.geonames.org` database. In Freebase, the transitive closure for a given predicate can be conveniently queried via SPARQL property paths. In YAGO, a recursive SQL query is required.

[5] Compound value types allow representing arbitrarily complex n-ary relations in a schema of 3-ary relations (see Pellissier Tanon et al. (2016), section 2.1 for an explanation of CVTs).

- **Known surface forms.** This path aims to find known aliases or nicknames of already-linked entities, for example *blade runner* for OSCAR_PISTORIUS. This corresponds to the Freebase predicate /common/topic/alias.

Due to the limited domain in our evaluation (see subsection 3.1.3), i.e. news articles and forum discussions about politics, this list of paths was compiled manually. In a different domain, e.g. sports, different paths, e.g. ones corresponding to team membership, would be more relevant. A principled method of obtaining relevant paths would be to use annotated data to collect statistics on paths that frequently connect entities, which is left for future work. We are now ready to add the next two sieves to the pipeline:

Related entity finder. This sieve traces paths through the knowledge base graph as described above, starting from an already linked entity, and arriving at related entities. The surface forms of related entities are then searched in the text and, if found, linked to their corresponding entity.

Known surface form finder. This sieve queries known surface forms of linked entities as described above and links all matches in the text to the corresponding entry in the knowledge base.

Low-precision, high-recall sieves

The final group of sieves aims to boost recall at the expense of lower precision. This group is run at a late stage in the pipeline in order to minimize error propagation.

Possible genders. This sieve annotates each person mention with its compatible semantic genders.[6] For person mentions already linked, there is only one possible gender, which can easily be queried from the knowledge base. For all other person mentions, we employ the CoreNLP gender annotator, as well as a simple heuristic based on gender markers such as *Mr*, *Ms*, or *Lady*, which are not taken into account by CoreNLP. All remaining person mentions not covered by any of the applied methods are marked as compatible with both female and male gender. The purpose of this sieve is to prevent gender inconsistencies in the coreference clusters created by subsequent sieves.

Person name unification. This sieve applies heuristics in order to perform simple, string-based coreference resolution of person name mentions, such as *Obama* and *Barack Obama*. It unifies, that is, links to the same entry in the knowledge base, all first-name or surname-only person name mentions with their unambiguous full name antecedent, taking gender compatibility into account.

[6] We use the term *semantic* gender to avoid possible confusion with *grammatical* gender.

Most frequent sense fallback. For all mentions that have not been linked yet, this sieve assigns the entity with the highest prior probability for the given mention string. To prevent erroneous links caused by strings with many different possible low-probability referents, this sieve only applies to mentions if the mention string has a dominant sense. We say that a mention has a dominant sense if there exists an entity with a prior probability larger than a tunable threshold, which we set to 40 percent based on experiments on development sets.

First token unification. For all multi-token mentions whose first token is not a frequent word, this sieve unifies all matching tokens that have not been linked by an upstream sieve.

Abbreviation unification. For all multi-token entity mentions, such as *United States*, this sieves generate abbreviation strings, e.g. *US* and *U.S.*, and unify all matching occurrences in the text.

Country adjectivals mapping. This sieve maps all as yet unlinked occurrences of country adjectivals, such as *English*, *American*, or *Bhutanese*, to their corresponding country.[7]

Having designed the main components of our entity linking system's architecture, we now turn to its implementation and evaluation.

3.1.3 Evaluation at the TAC 2015 Entity Discovery and Linking Shared Task

We evaluated our interleaved multitasking approach by participating in the Entity Discovery and Linking shared task held at Text Analysis Conference (TAC) 2015 (Ji, Nothman, Hachey, and Florian, 2015). The conference organizers provided a task specification, training and test data, as well as evaluation metrics, which we describe in the following.

Task specification

In the TAC 2015 entity discovery and linking shared task, systems were required to perform the following tasks:

- **Mention detection.** Given raw text in English, Spanish, or Chinese, the systems finds mentions of specific entities, for example Barack Obama, but not *the Obamas*. A mention can either be a named entity, taking the same example

[7] This mapping is derived from: `https://en.wikipedia.org/wiki/List_of_adjectival_and_demonymic_forms_for_countries_and_nations`

again, Barack Obama, or nominal, e.g. president. Nominal mentions are limited to persons. Mentions are detected correctly only if their character offsets in document exactly match the gold standard offsets. There is no partial credit given for partial matches.

- **Entity typing.** The systems assigns one of the following entity type to each mention: Person (PER), Geo-political Entity (GPE), Organization (ORG), Location (LOC), Facility (FAC).
- **Entity linking.** The system links each entity mention to its corresponding entry in the knowledge base or classifies as NIL those mentions without a matching entry.
- **Cross-document NIL clustering.** The system clusters all NIL mentions that refer to the same entity. The clustering is performed not only within one document, but across all documents in all three languages.

Data

The training data consists of raw text documents and standoff annotations of entity mentions, entity links, entity types, and clusters of NIL mentions. The documents are from the news domain and are divided into two genres, newswire and discussion forums. The English subset of the data consists of 85 newswire and 83 discussion forum documents. Compared to newswire, discussion forum documents pose several challenges, as the excerpt in Figure 3.5 on the next page shows:

- Spelling and grammar mistakes, such as the missing apostrophe in *Patience Jonathans Whereabouts*;
- inconsistent capitalization, such as *aso rock* instead of *Aso Rock*; and
- frequent use of nicknames and pejoratives, such as *Mama P*, *P baby*, *mama piss*, *Mama Peace*, and *lady shrek*, which all refer to the former First Lady of Nigeria PATIENCE_JONATHAN.

Evaluation

Participants were allowed to submit up to five runs with different configurations. The submissions were evaluated by the organizers using three main metrics:

- **`strong_typed_mention_match`.** This set-based metric jointly evaluates mention detection and mention typing.

```
<?xml version="1.0" encoding="utf-8"?>
<doc id="ENG_DF_001230_20150407_F0000008P">
<headline>
Presidency Silent On Patience Jonathans Whereabouts
</headline>
<post id="p1" author="personal59" datetime="2015-04-07T01:21:00">
http://www.punchng.com/news/presidency-silent-on-patience-jonathans-whereabouts/
</post>

<post id="p2" author="QuotaSystem" datetime="2015-04-07T01:30:00">
She is too ashamed to show her face after being disgraced out of aso rock by the same northern
children she insulted. Those "born troway" pikins dem for Kano have done Mama P a strong thing ooo
... chai! grin
</post>

<post id="p3" author="mypals" datetime="2015-04-07T01:34:00">
Learn to be civil bro. Thanks.
http://www.nairaland.com/2215960/warri-south-constituency-ii-chief
</post>

<post id="p4" author="ddaammyy" datetime="2015-04-07T01:53:00">
P baby is recounting her losses. Feel so sori fr her though. This one na from fame to shame.
</post>

<post id="p5" author="bettercreature" datetime="2015-04-07T01:56:00">
Are you people missing mama piss or why are you looking for her after refusing to vote for her husband
</post>

<post id="p6" author="QuotaSystem" datetime="2015-04-07T02:00:00">
Though she is the most uncivil and uncouth first woman Nigeria ever had, I will be the bigger person
and edit appropriately.
</post>

<post id="p7" author="mypals" datetime="2015-04-07T00:00:00">
Learn to be civil bro. Thanks.
Was she civil when she made those hateful utterances against northern ppl and stoning APC supporters
?
She deserves what's coming to her; I have no pity for either her or her husband - the thousands of
innocent lives lost, maimed, destroyed and families brutalised cannot be easily forgotten or forgiven.
</post>

<post id="p8" author="davit" datetime="2015-04-07T04:15:00">
Mama Peace don run for her live! smiley she no wan go jail.
</post>

<post id="p9" author="egift" datetime="2015-04-07T04:26:00">
"I don't want to take food to my husband in jail" - Patience.
</post>

<post id="p10" author="Caseless" datetime="2015-04-07T04:34:00">
I'm happy that lady shrek is off my tv screen.
</post>
```

FIGURE 3.5: Excerpt of a forum discussion from the TAC 2015 training data.

- `strong_all_match`. This set-based metric evaluates mention detection, entity links, and NIL classification.

- `mention_ceaf`. This metric evaluates the clusters formed by NIL clustering and entity linking.

For a set of gold annotations G and a set of system annotations S, the two set-based metrics are calculated as precision P, recall R and their harmonic mean F_1:

$$P = \frac{|G \cap S|}{|S|} \qquad R = \frac{|G \cap S|}{|G|} \qquad F_1 = \frac{2PR}{P+R}$$

The clustering metric is a variant of `CEAF` (X. Luo, 2005). It first finds an optimal alignment between gold and system clusters and then calculates precision, recall, and F_1 of aligned clusters.

Implementation

We implemented all sieves in the UIMA framework (Ferrucci and Lally, 2004), using the Stanford CoreNLP (Manning et al., 2014) UIMA components provided by DKPro (Gurevych et al., 2007) for text segmentation, POS tagging, and named entity recognition, and DKPro WSD (Miller et al., 2013) for representing entity mentions and entity links.

In addition to the reference knowledge base provided by the task organizers, namely, a specific version of Freebase, our system makes use of the following resources:

- CrossWikis (Spitkovsky and A. X. Chang, 2012), as an inventory of unambiguous surface forms;

- YAGO (Suchanek et al., 2007), as a gazetteer;

- Wikipedia (2013 dump), for most-frequent-sense statistics collected from inter-article links; and

- Word lists such as demonyms and gender-specific salutations.

In addition to the sieves described in subsection 3.1.2, we implement the following post-processing steps specific to the shared task:

Family reference removal. This sieve removes collective entity mentions that tend to be incorrectly linked or treated as NIL entities by previous sieves, such as plural forms of known surnames, e.g. *the Steenkamps*, and family references such as *the Steenkamp family*.

Post authors. In texts taken from discussion forums, the shared task specification required the disambiguation of mentions of participants in a discussion, such as the author of a post. An initial list of post authors can be trivially obtained from metadata. This sieve then performs naive coreference resolution by looking for strings matching known post authors in the discussion text, if the post author's name is not one of the most frequent English words.

Media and news organization removal. According to the shared task specification, generic mentions of media or news organization such as *CNN* in

(44) CNN reported that [. . .]

should not be linked. This sieve removes any linked mention whose most salient entity types, corresponding to the Freebase predicate /common/topic/notable_types, include any of /tv/tv_network, /broadcast/radio_network, or /book/newspaper. Since the notable_types in the knowledge base are incomplete and do not cover all media and news organizations, this sieve is also applied if the first sentence of the entity's description in the knowledge base, corresponding to the predicate /common/topic/description, contains a noun phrase such as *news website* or *television network*.

Finally, we experiment with a combination of our sieve-based system and the system by Fahrni, Heinzerling, et al. (2014), which performs joint global disambiguation and NIL clustering. The system is based on a Markov Logic Network (MLN) whose weights were trained on 500 Wikipedia articles, i.e. not on training data from the shared task.[8] Mentions linked by high-precision upstream sieves are provided to the MLN as ground truth. These already-linked mentions have a dual benefit by reducing the MLN's search space and by providing additional contextual information. Figure 3.6 on the facing page gives an overview of our implementation.

Results

Table 3.5 on page 40 shows the results of the four runs we submitted. The configurations for each run are described in Table 3.6 on page 40 Since we only attempted the English subset of the trilingual queries, we report only monolingual English results. We also did not attempt to resolve nominal coreference, which lowers our recall scores. For entity typing we submitted only a baseline combining the output of the CoreNLP 3-class named entity recognizer and 3-class type information from the knowledge base. This explains our low ranking in terms of the

[8] For a detailed description see Fahrni, Heinzerling, et al. (2014).

1. Preprocessing:
   - Tokenization
   - Part-of-speech tagging
   - Named entity recognition
2. High-precision mention detection and linking:
   - Almost unambiguous mention sieve
   - Entity type mismatch filter
   - Mention-surface mismatch filter
3. Path-based mention detection and linking:
   - Related entity finder
   - Known surface form finder
4. Joint global disambiguation and NIL clustering (Fahrni, Heinzerling, et al., 2014)
5. Low-precision, high-recall sieves:
   - Possible genders
   - Person name unification
   - Most frequent sense fallback
   - First token unification
   - Abbreviation unification
   - Country adjectivals mapping
6. Post-processing:
   - Family reference removal
   - Post authors
   - Media and news organization filter

FIGURE 3.6: Overview of the system implemented for the TAC 2015 entity discovery and linking shared task.

	NER			Linking			Clustering		
Run	P	R	F1	P	R	F1	P	R	F1
HITS1	0.629	0.514	0.566	**0.707**	0.578	0.636	0.747	0.610	0.671
HITS2	**0.627**	0.525	0.571	0.703	0.588	**0.640**	**0.748**	0.626	**0.682**
HITS3	0.614	0.540	**0.574**	0.673	**0.592**	0.630	0.719	**0.633**	0.673
HITS4	0.617	0.536	**0.574**	0.670	0.582	0.623	0.709	0.616	0.659
Best	0.791	0.673	0.727	0.703	0.588	0.640	0.765	0.619	0.684

TABLE 3.5: Monolingual English entity linking performance. NER is performance in terms of `strong_typed_mention_match`, Linking is `strong_mention_match`, and Clustering `mention_ceaf`. Best shows the overall best result for each metric among all participants.

Run	**Setting**
HITS1	Sieves only, remove low-confidence NILs
HITS2	Sieves + MLN, remove low-confidence NILs
HITS3	Sieves only, retain low-confidence NILs
HITS4	Sieves + MLN, retain low-confidence NILs

TABLE 3.6: Configuration of the four runs we submitted. *Sieves only* means that only sieves were applied, *Sieves + MLN* that the MLN system by Fahrni, Heinzerling, et al. was run as well, and removal or retention of *low-confidence NILs* specifies whether mentions that were classified as NIL entities with only very low confidence were retained in an attempt to boost recall, or removed in order to increase precision.

`strong_typed_mention_match` metric. For linking, our system is ranked first, and for clustering second, with only 0.002 F_1 points difference to the first-ranked system.

To study the effect of global inference in comparison to our sieve-based approach which performs only pairwise or local inference, we submitted two runs which use only sieves, and two runs which use both sieves and the MLN system by Fahrni, Heinzerling, et al. This comparison is shown in Figure 3.7 on the next page.

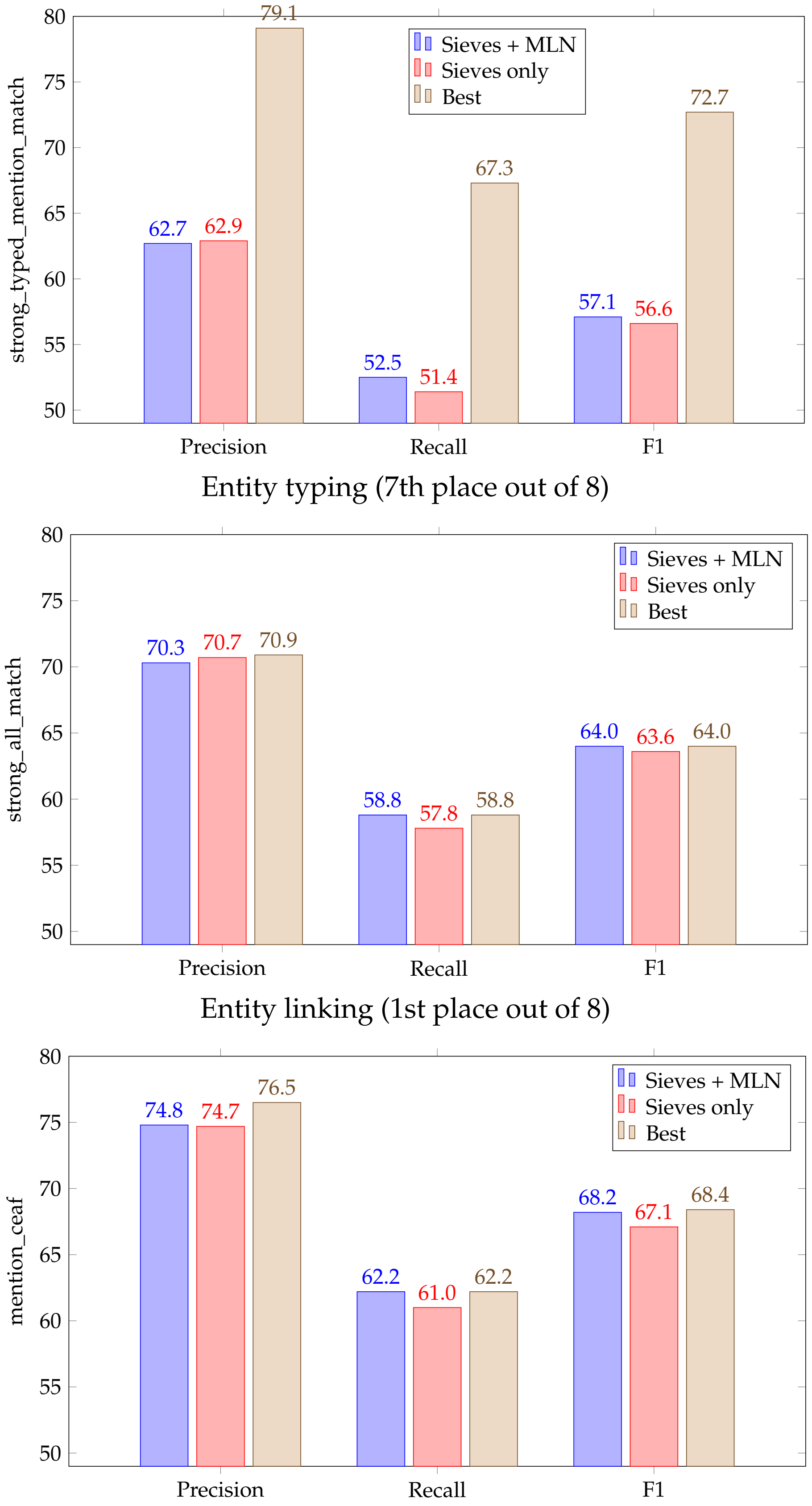
The comparison between the sieves-only and sieves+MLN runs shows that the sieves our system applies before and after the MLN-based global joint disambiguation and clustering step, account for almost all of the linking and clustering performance. A possible explanation is the fact that the MLN-based system was trained on a different corpus, while the sieves are tailored to the shared task training data, namely newswire texts and discussions of news-related topics. However, even in comparison to the systems of other shared task participants, the sieves-only setting (HITS1), achieves the same ranking in terms of `strong_link_match` and `mention_ceaf` as our best run (HITS2), which combines sieves and the MLN-based system. While our approach performed well in two out of three metrics, it has several limitations.

3.1.4 Limitations

Our proposed approach and its implementation for the TAC 2015 entity discovery and linking shared task have several limitations. Since many sieves require on information from the knowledge base to find related entities, they are not suited for NIL classification and clustering. For this, we rely on the system by Fahrni, Heinzerling, et al., which improves our entity clustering score by 1.1F_1 points, as shown at the bottom of Figure 3.7 on the following page. The sieve-only approach is, given preprocessing, a deterministic sequence of knowledge base queries and string matching operations. Hence, it is fast and interpretable, but the design of the sieves requires manual effort and some of the sieves, e.g. the ones involving political affiliations, were tailored specifically to the domain of the training and test data. Furthermore, recall that the shared task specification required resolving nominal coreference, such as in this example:

(45) Netanyahu will be busy being a leader, unlike Obama the golfer!

Here, according to the shared task annotation guidelines, both Obama and golfer refer to the BARACK_OBAMA. However, any attempt to resolve nominal coreference degraded performance as recall gains were outweighed by precision errors.



Entity typing (7th place out of 8)

Entity linking (1st place out of 8)

Entity clustering (2nd place out of 8)

FIGURE 3.7: Comparison of our best sieves-only run, the combination of sieves and global inference (sieves + MLN), and the best system for each metric in the English-only evaluation.

3.1.5 Summary

In summary, the core of our proposed approach is a pipeline of deterministic rules, called sieves. These sieves perform the subtasks necessary for entity linking, i.e., mention detection, entity linking, NIL classification, and NIL clustering. The sieves are ordered in descending order of precision and according to dependency on results of previous sieves, This architecture follows similar approaches in coreference resolution but, crucially, performs several subtasks in free order via interleaved multitasking. Our participation in the TAC 2015 Entity Discovery and Linking shared task showed that this approach is simple but effective, achieving first and second place in two of the three main metrics in the English-only evaluation. For computational speed all sieves operate only on a single mention, i.e. performing local disambiguation, or on pairs of mentions, performing pairwise disambiguation, but do not globally disambiguate all mentions in a document. In the next section, we address this drawback and introduce an efficient method for exploiting global coherence in entity linking.

3.2 Global Coherence in Entity Linking

In the previous section we introduced interleaved multitasking as a method for exploiting coherence between pairs of entities. In this section, we propose an efficient method for exploiting diverse aspects of *global* coherence between all entities mentioned in a text.

3.2.1 Exploiting Global Coherence without Global Inference

While linguistically well-founded in the concept of cohesion (Halliday and Hasan, 1976), global inference approaches (Kulkarni et al., 2009; Hoffart, Yosef, et al., 2011) do not scale well in the number of mentions and in the number of candidate entities considered for each mention. Inference becomes prohibitively slow when processing long texts with many mentions or when disambiguating many highly ambiguous entity mentions. In contrast, local approaches do not suffer from scalability issues, since they only optimize the similarity between a single mention's context and a given candidate entity's knowledge base entry, without considering other entities mentioned in the document (Bunescu and Paşca, 2006; Cucerzan, 2007). Recent local inference approaches achieve state-of-the-art results by using convolutional neural networks to capture similarity at multiple context sizes (Francis-Landau et

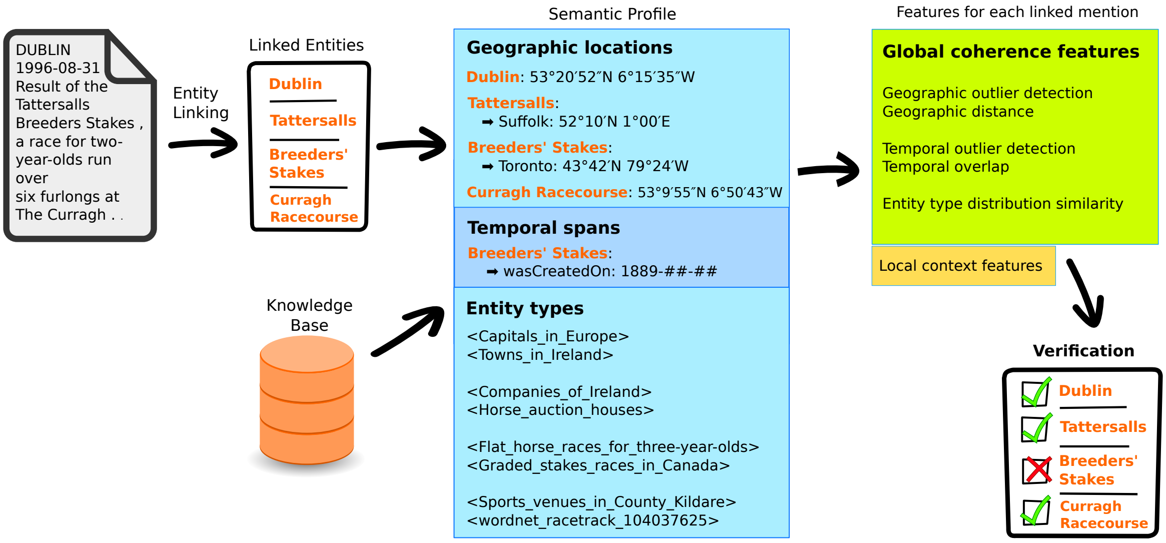

DUBLIN
1996-08-31
Result of the Tattersalls Breeders Stakes , a race for two-year-olds run over six furlongs at The Curragh . .
Entity Linking
Linked Entities
Dublin
Tattersalls
Breeders' Stakes
Curragh Racecourse
Knowledge Base
Semantic Profile
Geographic locations
Dublin: 53°20′52″N 6°15′35″W
Tattersalls:
➡ Suffolk: 52°10′N 1°00′E
Breeders' Stakes:
➡ Toronto: 43°42′N 79°24′W
Curragh Racecourse: 53°9′55″N 6°50′43″W
Temporal spans
Breeders' Stakes:
➡ wasCreatedOn: 1889-##-##
Entity types
<Capitals_in_Europe>
<Towns_in_Ireland>
<Companies_of_Ireland>
<Horse_auction_houses>
<Flat_horse_races_for_three-year-olds>
<Graded_stakes_races_in_Canada>
<Sports_venues_in_County_Kildare>
<wordnet_racetrack_104037625>
Features for each linked mention
Global coherence features
Geographic outlier detection
Geographic distance
Temporal outlier detection
Temporal overlap
Entity type distribution similarity
Local context features
Verification
Dublin
Tattersalls
Breeders' Stakes
Curragh Racecourse

FIGURE 3.8: Overview of our entity linking verification system.

al., 2016). However, local approaches fail, by definition, to take global coherence among entities into account.

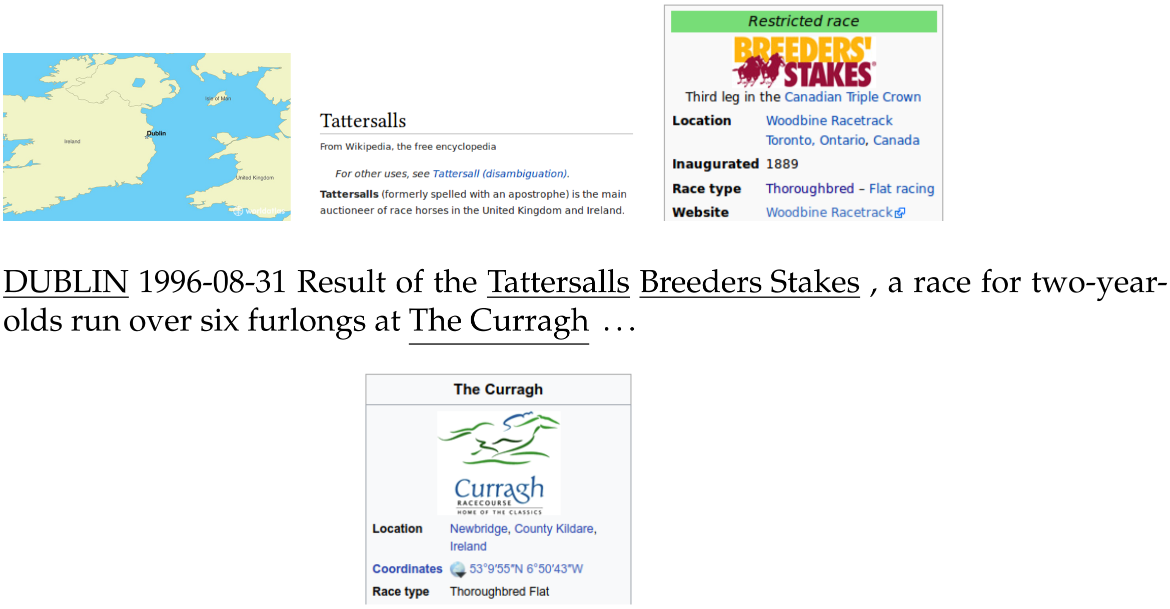
To avoid the trade-off between the efficiency of local inference on the one hand and the coherence benefits of global inference on the other, we propose a two-stage approach: In the first stage, candidate entities are ranked by a fast, local inference-based entity linking system. In the second stage these results are used to create a semantic profile of the given text, derived from rich data the knowledge base contains about the top-ranked candidates. Since the linking precision of current entity linking systems is relatively high, we assume that this profile is reasonably accurate and leverage it to measure the cohesive strength between a given candidate entity and the other, already-linked entities mentioned in the text. We then automatically *verify* the results from the first stage by classifying entity links as correct if they exhibit high coherence, and as wrong if there are only weak or no cohesive ties to the semantic profile. Figure 3.8 gives an overview of this process.

The verification results obtained in the second stage can then be used in at least three ways:

1. to increase linking precision by filtering out all entity links classified as wrong;

2. to rerank candidate entities by the class probability estimated by the verifier, i.e., prefer candidates that were predicted as correct with higher probability; or

3. to employ a more sophisticated entity linking system to re-link all entity links classified as wrong, using the entity links deemed correct as additional context.



Tattersalls

From Wikipedia, the free encyclopedia

For other uses, see Tattersall (disambiguation).

Tattersalls (formerly spelled with an apostrophe) is the main auctioneer of race horses in the United Kingdom and Ireland.

Restricted race

Third leg in the Canadian Triple Crown

Location Woodbine Racetrack Toronto, Ontario, Canada

Inaugurated 1889

Race type Thoroughbred – Flat racing

Website Woodbine Racetrack

DUBLIN 1996-08-31 Result of the Tattersalls Breeders Stakes , a race for two-year-olds run over six furlongs at The Curragh ...

The Curragh

Location Newbridge, County Kildare, Ireland

Coordinates 53°9′55″N 6°50′43″W

Race type Thoroughbred Flat

FIGURE 3.9: An example of misleading generic coherence. Text source: document 1112testa from the CoNLL development set.

In the following, we investigate options 1. and 2., leaving option 3. to future work.

As a motivating example for our approach, consider the sentence shown in Figure 3.9 , in which an entity linking system correctly linked the following mentions:

- DUBLIN → DUBLIN, the capital of Ireland;
- Tattersalls → TATTERSALLS, a race horse auctioneer based in the UK and Ireland; and
- The Curragh → CURRAGH_RACECOURSE, a course for horse races in Ireland.

These entities clearly situate the text in Ireland. However, several current entity linking systems compared in our experiments link the mention Breeders Stakes to the Wikipedia article about a Canadian horse race of the same name. This mistake was likely made because the actual referent, a horse race sponsored by Tattersalls and held in Ireland, does not have a Wikipedia article. The system is then misled by other evidence: high similarity between mention context and Wikipedia article due to the appositive *race*, as well as an almost perfect string match between mention string and the article title. This mistake results in an interpretation in which all entities except one are located in Ireland while one entity is isolated in Canada (Figure 3.10 on the next page).

We aim to prevent these kinds of mistakes by verifying entity linking results, using aspects of coherence that have not been employed for entity linking so far,

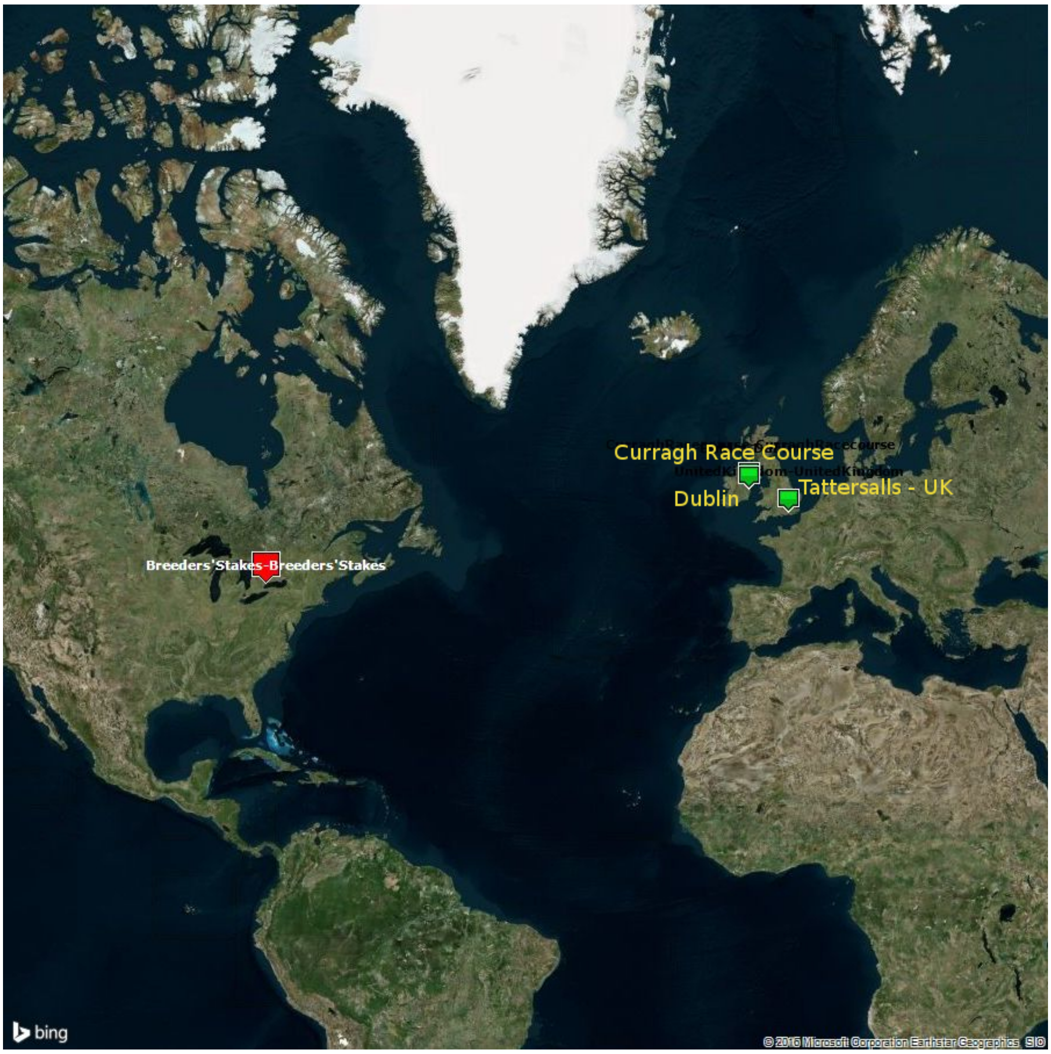

CurraghRacecourse-CurraghRacecourse
Curragh Race Course
UnitedKingdom-UnitedKingdom
Dublin
Tattersalls - UK
Breeders'Stakes-Breeders'Stakes
bing
© 2016 Microsoft Corporation Earthstar Geographics SIO

FIGURE 3.10: Example showing a geographical outlier: Breeders' Stakes (red, in Canada) and contextual entities located in Ireland and the UK (green). Image source: `https://www.bing.com/maps`.

such as geographical coherence in the example above. To do so, we assume that an existing entity linking system has linked all entity mentions found in a document to an entry in the knowledge base. Due to entity linking mistakes, some of these entities may, in fact, not be referred to in the document. However, we can also expect that some of these entities have been correctly linked by the entity linking system.[9] We now query the knowledge base for information about the entities that, according to the entity linking system, are mentioned in the document, regardless of whether this is actually the case or not. This information includes geographic data such as locations of the entities mentioned in it, temporal data such as years of birth or death, and the semantic types of all mentioned entities. We call this information the *semantic profile* of the document. The semantic profile of our motivating example is visualized as the blue box in Figure 3.8 on page 44. The main idea in our approach is that the semantic profile allows judging whether a given linked entity is coherent with the document, that is, whether it is highly related to other entities mentioned in the document or not. We cast the comparison of a linked entity to the document's semantic profile as a supervised classification task.

The *input* for our classifier is the *output* of an entity linking system, which consists of links to entries in the knowledge base for all entity mentions found in a given set of documents. Next, we extract a rich set of global, pairwise, and local features for each linked mention. Using the gold annotations, which provide the correct knowledge base entry for all mentions in the document set, we then train a classifier to predict whether a given mention was linked correctly by the system or not.

Recall that in global disambiguation approaches to entity linking (Kulkarni et al., 2009; Hoffart, Yosef, et al., 2011; Fahrni and Strube, 2012; Moro et al., 2014), global inference is an NP-hard problem, since all combinations of all candidate entities of all mentions are considered simultaneously. In our proposed automatic verification setting, inference scales linearly in the number of mentions since we only need to compare the top candidate entity for each mention to the document's semantic profile. This allows employing knowledge-rich, global coherence features that otherwise would have prohibitively high computational cost. Our features are designed to exploit several aspects of global coherence, such as geographic or temporal coherence.

[9] Since entity linking precision ranges between 60 and 90 percent on common datasets, this is not an unrealistic expectation. See, for example, the baseline precisions on the CoNLL and TAC15 datasets in Figure 3.12 on page 58.

Predicate
/location/location/geolocation
/organization/organization/geographic_scope
/time/event/locations
/sports/sports_team/location
/organization/organization/headquarters

TABLE 3.7: Freebase predicates for querying geo-coordinates of locations, geo-political entities, events, and organizations.

3.2.2 Aspects of Global Coherence

Global coherence captures how well a candidate entity fits into the overall semantic profile of a text. Current global inference approaches optimize a single coherence measure, most commonly a measure of general semantic relatedness such as the Milne-Witten distance (Milne and Witten, 2008a), or keyphrase overlap relatedness (KORE) (Hoffart, Seufert, et al., 2012).

In contrast, verification allows employing many global coherence features, which we categorize according to four aspects of coherence: geographical coherence and temporal coherence, which to our knowledge have not been used before in entity linking, as well as entity type coherence and the general semantic relatedness mentioned above.

Geographic Coherence

Entities mentioned in a text tend to be geographically close or clustered around very few locations. We use this observation to identify geographic outliers as potential entity linking mistakes. We aim to identify these kinds of mistakes by first querying the geographic locations (Table 3.7) of all linked mentions in the document. The result of this query is a set geo-coordinates, i.e. latitudes and longitudes, of all locatable entities. We then use an ensemble of geographic outlier detection algorithms implemented in by off-the-shelf software.[10] This process yields a binary feature indicating whether a linked entity is a geographic outlier or not.

Since geographic outliers are rare and hence the resulting features sparse, we also also add a feature for the average geographic distance $\bar{d}(d, E)$ of an entity e to all other entities in document D:

$$\bar{d}(e, D) = \frac{\sum_{e' \in D \setminus e} d(e, e')}{|D| - 1}$$

[10] We use the DBSCAN and OPTICS implementations in the ELKI clustering toolkit (Achtert et al., 2011).

where $d(e, e')$ is the geographic distance between entities e and e', and $|D|$ is the number of entities mentioned in D. This feature is based on the intuition that a candidate entity which is geographically closer to other entities is more likely to be correct than a distant one.

A complication not considered by the above feature is that fact that geographic scope varies between documents. For example, entities mentioned in a text about world politics will be geographically more distant than entities in a text about a local business). As a scale-invariant distance measure $s(e, D)$, we divide the average distance $\bar{d}(e, D)$ by the average distance between all other entities:

$$s(e, D) = \bar{d}(e, D) / \frac{\sum_{e', e'' \in D \setminus e} d(e', e'')}{|e', e'' \in D \setminus e|}$$

Having captured intuitions about geographic coherence, we now carry over these intuitions to temporal aspects of coherence.

Temporal Coherence

As a motivating example for exploiting temporal coherence, consider the following excerpt of a document from the CoNLL development set:[11]:

(46) BONN 1996-08-30 [...] German Foreign Ministry spokesman Martin Erdmann...

One of the entity linking systems used in the experiments for this thesis produced the following entity links:

- Martin Erdmann → MARTIN_ERDMANN, a German diplomat born on 25 January **1955**; and
- German → NAZI_GERMANY, a name for Germany from 1933 to **1945**.

It is highly unlikely that Martin Erdmann worked as spokesman for the government of a country that ceased to exist in 1945 before he was born in 1955. In addition, the dateline reveals that this news article was written in 1996. This temporal incoherence suggests an entity linking mistake.

Applying the notion of coherence to the temporal dimension, we observe that entities mentioned in a text tend to be temporally close or clustered around a few points in time. Entities are associated with temporal ranges with a *begin*, that is the point in time at which the entity comes into existence, and an *end*, that is the point

[11] Document ID: `1017testa`

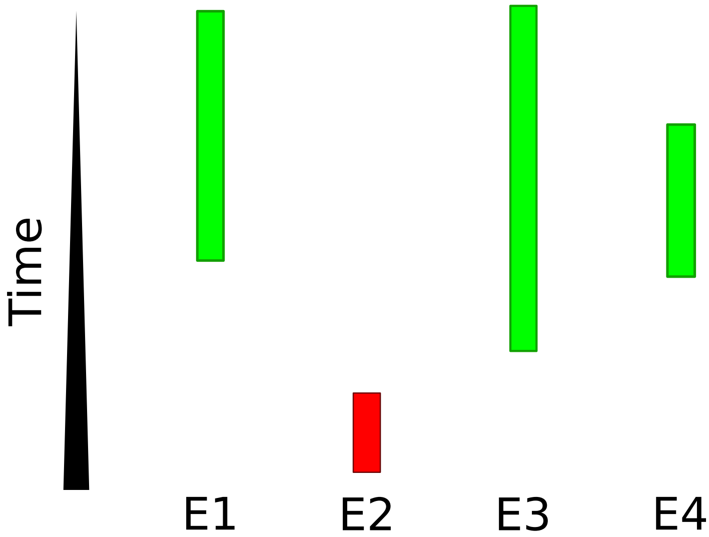


FIGURE 3.11: Example showing a temporal outlier (red, entity E2) whose temporal range does not overlap with the temporal ranges of other entities in a document (green, entities E1, E3, E4).

in time at which the entity ceases to exists. Using the same method as in geographical outlier detection, we perform temporal outlier detection on all *begin* and *end* times associated with linked entities in the given text. A difference to geographic locations is that entities have an associated temporal *range*. We identify a temporal range as outlier if both its *begin* and *end* were found to be outlier, an example of which is shown in Figure 3.11 .

Since temporal outliers are rare, we also add a feature aiming to capture temporal proximity and distance in a softer fashion with higher coverage. This is done by calculating the total overlap $T(e, D)$ between the temporal range $t(e)$ of a candidate entity e, and the known temporal ranges of all other linked entities in the document D:

$$T(e, D) = \sum_{e' \in D \setminus e} \left| t(e) \cap t(e') \right|$$

where $|t(e) \cap t(e')|$ is the length of the overlap between the temporal ranges of entities e and e'.[12]

Analogously to the geographic distance feature, we take temporal proximity, i.e. a large overlap with other temporal ranges, as evidence for a correctly linked entity, and temporal distance, i.e. only small or no overlap with other temporal ranges, as

[12] $T(e, D)$ increases non-strictly monotonically as the number of entities mentioned in a document grows larger. Consequently, we also tried extracting a version of this feature that is normalized by the number of entity mentions in the document, but did not see any effect. This is likely due to little variation in the number of entities per document for which the knowledge base contains temporal information.

Predicate
/people/person/date_of_birth
/organization/organization/date_founded
/sports/sports_team/founded
/location/dated_location/date_founded
/time/event/start_date
/film/film/initial_release_date
/music/album/release_date
/music/release/release_date
/architecture/structure/construction_started
/architecture/structure/opened
/people/deceased_person/date_of_death
/location/dated_location/date_dissolved
/time/event/end_date
/business/defunct_company/ceased_operations
/architecture/structure/closed

TABLE 3.8: Freebase predicates for querying the *begin* (top) and *end* (bottom) of an entity's temporal range.

evidence for a linking mistake. Temporal ranges are queried from the knowledge base using the predicates shown in Table 3.8 .

The final feature using temporal information checks whether an entity's temporal ranges contains the document's creation date. This feature is based on the intuition that, especially in the news genre, an existing entity is more likely to be mentioned than an entity that has already ceased to exist or did not exist yet at the time of writing. The document creation date is either trivially obtained if metadata is present, or heuristically by using the first date found in the document text by the HeidelTime temporal tagger (Strötgen and Gertz, 2010). Having covered geographic and temporal coherence, we now turn to coherence of entity types.

Entity Type Coherence

Frequency statistics of the types of entities mentioned in a text are an indicator of what the text is about. For example, looking at the entity type distribution, shown in Table 3.9 on the next page, we can tell that the text from which these statistics were collected appears to be about sports in general and rugby in particular. Unlike other methods for representing the "aboutness" of a text, such as topic models (Blei et al., 2003), entity type statistics are grounded in the knowledge base, thus offering a simple method of measuring the relatedness between entities in terms of their types via the similarity of their type distributions.

tf-idf	Count	Type
1115.67	2	/base/rugby/rugby_club
243.62	3	/organization/organization
231.76	2	/base/schemastaging/sports_team_extra
183.49	2	/sports/sports_team
56.34	2	/base/tagit/concept

TABLE 3.9: Entity type distribution in a document about rugby, sorted by tf-idf scores.

Specifically, we model entity type coherence between a given candidate entity e and all other linked entities in document D as the cosine similarity of the respective type distributions. Type frequencies are weighted by their tf-idf scores (Spärck Jones, 1972). This is done to discount frequent types, for example an unspecific entity type such as /base/tagit/concept, and give more importance to salient types occurring in the document, e.g. an informative entity type like /base/rugby/rugby_club:

$$coh_{type}(e, D) = sim(types(e), tfidf(types(D)))$$

where sim is the cosine similarity, $types(e)$ a binary vector indicating the types of entity e, and $types(D)$ a vector whose entries are occurrence counts of entity types in document D, which are weighted by $tfidf$.

Generic Semantic Relatedness

To capture other aspects of coherence between entities that are not covered by the three specific aspects of coherence introduced above, we include measures of generic semantic relatedness which are a standard feature in global inference systems. Specifically, we add features for the average and maximum semantic relatedness $SemRel(e, D)$ of a candidate entity e with respect to all other entities e' mentioned in document D, using two semantic relatedness measures:

$$SemRel_{max}(e, D) = max_{e' \in D \setminus e} SemDist(e, e')$$

$$SemRel_{avg}(e, D) = avg_{e' \in D \setminus e} SemDist(e, e')$$

where *max* and *avg* are the maximum and average operators. *SemDist* denotes either the Milne-Witten Distance (Milne and Witten, 2008a), which defines relatedness of Wikipedia entries in terms of shared incoming article links, or the Normalized Freebase Distance (Godin et al., 2014), an adaptation of the Milne-Witten Distance to Freebase entities.

3.2.3 Pairwise Coherence Features

We incorporate two sets of features that apply to pairs of entity mentions.

Semantic relation. Given a pair consisting of a candidate entity and an entity mention in its context, we add a feature encoding whether a (and if yes which) semantic relation exists between the two entities. We add different features depending on the type of context in which the entity pair occurs: in the same sentence, within a fixed token window, and within the same noun phrase. For example, in the noun phrase *German Chancellor Angela Merkel*, we find a wasBornIn and a isLeaderOf relation between the entities GERMANY and ANGELA_MERKEL. We expect this feature to be sparse, but strong evidence for both arguments of the identified relation being linked correctly. We record the relation predicate, as some relations tend to be more informative than others. For example, a playsFor relation, which holds between players and sports teams, should provide stronger evidence than the less specific isCitizenOf relation, which holds between citizens and countries.

Person name consistency. Having observed that some local inference systems tend to make the mistake of linking a full name mention (e.g. "John Smith") to one entity, and a coreferent surname-only mention ("Smith") to a different one, we add a binary feature that indicates whether a candidate entity assigned to a partial person name mention agrees with its unambiguous full name antecedent.

3.2.4 Local Features

Since the global and pairwise features do not have a coverage that is high enough to provide evidence for all linked entities, we employ local features that are devised to capture the similarity between a candidate entity and its textual context. As these features are commonly used in entity linking systems, we only give brief descriptions for completeness.

Popularity prior. The prior probability of the candidate entity given its mention, obtained from the CrossWikis dictionary (Spitkovsky and A. X. Chang, 2012). This feature aims to cover unambiguous and almost unambiguous mentions.

Entity type agreement. A binary feature indicating whether the candidate entity type, as found in the knowledge base agrees with the named entity type, as determined by the named entity recognition system applied during preprocessing.
Keyphrase match. Knowledge bases contain various sources of key phrases that are strongly associated with a given entity. Keyphrases include surface realizations of semantic types, e.g. *politician* for the entity type `/person/politician`, or salient noun phrases in entity descriptions, e.g. noun phrases occurring in the first sentence of a Wikipedia article. We add a binary feature indicating whether a known keyphrase occurs in the context of a given candidate entity.
Demonym match. This binary feature indicates whether a mention is a demonym of its linked entity. For example, the mention string *French* is a demonym match for the entity FRANCE.
Mention-entity string match. Finally, we extract features encoding the string similarity between a mention and the known surface forms of a candidate entity. The similarity measures include exact match, case-insensitive match, head match, match with stop words filtered, fuzzy string match, Levenshtein distance, and abbreviation pattern matches, as well as different combinations of these.

3.2.5 Experiments

We evaluate our automatic verification method by applying it to the entity linking results produced by seven systems on two standard datasets.

Data

The datasets used in our experiments are: CoNLL, which consists of 1393 Reuters news articles annotated with Wikipedia links by (Hoffart, Yosef, et al., 2011); and TAC15, which contains news articles and discussion forum texts annotated with Freebase links for the TAC 2015 Entity Discovery and Linking shared task (Ji, Nothman, Hachey, and Florian, 2015).

The knowledge base coverage for each of our proposed global coherence features on these two datasets is shown in Table 3.10 on the facing page. YAGO and Freebase contain entity type information for almost all linkable entities mentioned in the two datasets. Geographic data is available for 62.9 percent on CoNLL, but only for 41.5 percent of entities mentioned in TAC15. This difference is likely due to the large fraction of documents from the sports genre in CoNLL, which include match results tables mentioning a large number of cities, sports teams, and other

Dataset	CoNLL	TAC15
Entity Type	99.2	98.5
Geographic	62.9	41.5
Temporal	87.6	79.4

TABLE 3.10: Knowledge base coverage of our proposed global coherence features. Shown are the percentages of linkable mentions in each dataset for which the knowledge base (YAGO or Freebase) contains the required information for each coherence feature set.

popular entities that are well-represented Freebase and YAGO. Temporal information is available for most entities in both datasets.

Systems

We evaluate our proposed approach by automatically verifying the results produced by several off-the-shelf entity linking systems:

- **AIDA** (Hoffart, Yosef, et al., 2011). This graph-based system globally optimizes three factors: a popularity prior, the context similarity of mention and candidate entity, and the global coherence between entities quantified via generic semantic relatedness measures. We use the AIDA system output on the CoNLL dataset as provided by the Wikilinks project.[13]
- **SPOTL** (Daiber et al., 2013). DBpedia Spotlight is a local inference system. We use results obtained from the Spotlight webservice.[14]
- **FL** (Francis-Landau et al., 2016). This local inference system models mention and entity context with a convolutional neural network (CNN). The CNN encodes a given mention's context at different granularities, namely, a small context window, the containing paragraph, and the document. It then encodes the Wikipedia articles of candidate entities in a similar fashion and provides similarities between context and candidate entity encodings as features to an existing entity linking system.
- **PH** (Pershina et al., 2015). This global inference system applies Personal PageRank to a graph whose nodes represent candidate entities and whose edges indicate if a link between the corresponding Wikipedia articles exists. PH achieves the best published result on the CoNLL dataset among the systems compared in our evaluation.

[13] `https://github.com/wikilinks/conll03_nel_eval`

[14] `https://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki/Web-service`

- **TAC-1** (Heinzerling, Judea, et al., 2015). This system uses local and pairwise inference and is described in section 3.1.

- **TAC-2** (Sil, Dinu, et al., 2015). This system employs a global inference approach which partitions a document into sets of mentions that appear near each other. The partitioning reduces the search space during global optimization and is motivated by the intuition that a given mention's immediate context provides the most salient information for disambiguation.

- **TAC-3** (Dai et al., 2015). This local inference system models mentions and entity context with a CNN and word embeddings.

  The systems were chosen for their popularity (AIDA, SL), state-of-the-art performance on CoNLL (FL, PH), and top performance on TAC15 (TAC systems). Unless stated otherwise, we use system output provided by authors for CoNLL systems, and provided by the workshop organizers for TAC15 systems.[15] Our evaluation does not include (Globerson et al., 2016) and (Yamada et al., 2016), who report better performance on CoNLL than PH, but were unable to make system output available.

3.2.6 Implementation and Experimental Setup

Feature extraction is implemented as a UIMA pipeline (Ferrucci and Lally, 2004); using the Stanford CoreNLP (Manning et al., 2014) UIMA components provided by DKPro (Eckart de Castilho and Gurevych, 2014) for text segmentation, POS tagging, and named entity recognition; DKPro WSD (Miller et al., 2013) for representing entity mentions and links, and using Freebase (Bollacker et al., 2008) and YAGO (Hoffart, Suchanek, et al., 2011) as knowledge bases.

After feature extraction, we train a random forest classifier using the implementation in `scikit-learn` (Pedregosa et al., 2011). Various other classifiers we tried, for example neural networks, showed no better performance during cross-validation on development sets. We train a classifier for each dataset, one using FL system results for 216 documents in the CoNLL development set and one using TAC-1 results for the 168 documents in the English subset of the TAC15 training set.

For evaluation, we apply the verifier trained on FL CoNLL development results to the test set results of the FL and AIDA systems, and a verifier trained on PH

[15] `http://www.nist.gov/tac/2015/KBP/data.html`

training data to the PH test set results. For the test set output of TAC systems 1-3 we apply the verifier trained on the TAC15 training set output of TAC-1.

Our evaluation metric is `strong_link_match` as implemented by the Wikilinks project for the CoNLL dataset, and the official NIST scorer (Hachey, Nothman, et al., 2014) for TAC15. As in the TAC 2015 entity discovery and linking shared task (see subsection 3.1.3), this metric measures precision, recall, and $F1$ of matching entity links and mention spans.

3.2.7 Results and Discussion

Evaluation results are visualized in Figure 3.12 on the next page and reported in more detail in Table 3.11 on the following page. The results show that automatic verification yields F_1 score improvements across all evaluated entity linking systems. The impact is most noticeable for the systems that only use local and pairwise inference, namely FL with a 1.9 F_1 increase, TAC-1 with a 2.4 F_1 increase, and TAC-3 with a 1.1 F_1 increase. The improved TAC-1 result (68.1 F_1) is the best published linking score on the English-only TAC15 subset.

Improvements are smaller for the global inference systems, AIDA, HP, and TAC-2. In contrast to (Ratinov et al., 2011), who report only a very small increase in linking performance when incorporating global features into a local inference-based system, our results indicate that global features are useful and lead to considerable improvements.

As expected, improvements are caused by increased precision, due to filtering out likely linking mistakes. The fact that this increase is not accompanied by a commensurate decrease in recall shows that our method predicts wrong linking decisions with high accuracy.

On TAC15 data, we observe large improvements in linking precision of up to 10.4 percent. With the CoNLL dataset, the precision increase is less pronounced, arguably owing to the already higher baseline precision, which leaves less room for improvement. Since entity linking is usually performed as part of a larger task, such as knowledge base completion, search, or as part of a more comprehensive entity analysis system (Durrett and Klein, 2014), good precision is highly desirable in order to minimize error propagation to other system components and downstream applications.

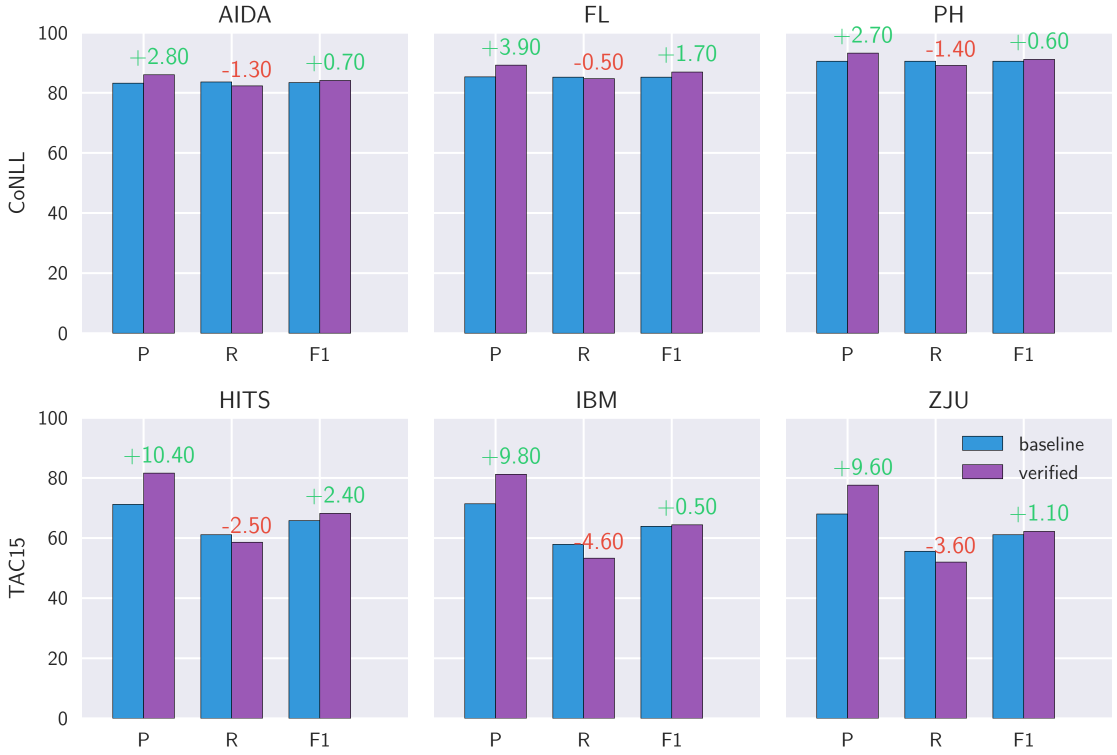


FIGURE 3.12: Visualization showing results on CoNLL and TAC15 test sets. *baseline* shows performance of the original systems, *verified* shows performance after application of our automatic verification method. Verification yields considerable precision improvements across all systems.

		Baseline			Verified			Δ		
Dataset	System	Prec.	Rec.	*F1*	Prec.	Rec	*F1*	Prec.	Rec.	F_1
CoNLL	AIDA	83.2	**83.6**	83.4	**86.0**	82.3	**84.1**	+2.8	-1.3	+0.7
	SPOTL	85.5	**80.5**	82.9	**93.0**	77.6	**84.6**	+7.5	-2.9	+1.7
	FL	85.3	**85.2**	85.2	**89.2**	84.7	**86.9**	+4.0	-0.5	+1.7
	PH	90.5	**90.5**	90.5	**93.2**	89.1	**91.1**	+2.7	-1.4	+0.6
TAC15	TAC-1	71.2	**61.1**	65.8	**81.6**	58.6	**68.2**	+10.4	-2.5	+2.4
	TAC-2	71.4	**57.9**	63.9	**81.2**	53.3	**64.4**	+9.8	-4.6	+0.5
	TAC-3	68.0	**55.6**	61.1	**77.6**	52.0	**62.2**	+9.6	-3.2	+1.1

TABLE 3.11: Results on CoNLL and TAC15 test sets. *Baseline* shows performance of the original systems, *Verified* shows performance after application of our automatic verification method, and Δ shows the corresponding change. Bold font indicates best results for each metric and system.

System	Prec	Rec	F1
FL baseline	85.3	85.2	85.2
FL filter	**89.2**	84.7	**86.9**
FL rerank	87.9	**85.6**	86.7

TABLE 3.12: Comparison of filtering and candidate entity reranking performance on the CoNLL test set.

3.2.8 Candidate Reranking

We resort to the somewhat crude decision of either retaining or removing an entity linked by an entity linking system if no candidate entities and no meaningful confidence scores are available. This is the case for the output of many entity linking systems, such as the systems participating in the TAC 2015 entity discovering and linking shared task.

In case the entity linking system outputs not only the top-ranked candidate entity, but also lower-ranked ones, we can apply our verification method to all candidates and rerank them according to their probability of being correct, as determined by our trained classifier. For example, if the entity linking system linked a mention to candidate entity e_1 over candidate e_2, but verification assigns a higher probability of being correct to e_2, we rerank e_2 over e_1. Since we assume that the document's semantic profile derived from entity linking results is sufficiently accurate, we do not recreate it after reranking a candidate.

Reranking the candidate entities produced by the FL system on the CoNLL test set, this achieves a similar increase in $F1$, but with a different precision-recall tradeoff (Table 3.12). We observe highest precision at the cost of a lower recall for filtering, while reranking increases both precision and recall.

3.2.9 Ablation Study

We conduct an ablation study to assess the impact of the proposed global coherence features on prediction performance. Applying backward elimination (John et al., 1994), we iteratively remove one feature set and successively eliminate the feature set with the largest impact.

Ablation results are shown in Figure 3.13. Surprisingly, the string similarity features have a large effect across all three systems. This suggests that current systems do not optimally utilize mention string similarity when selecting and ranking candidate entities for a given mention.

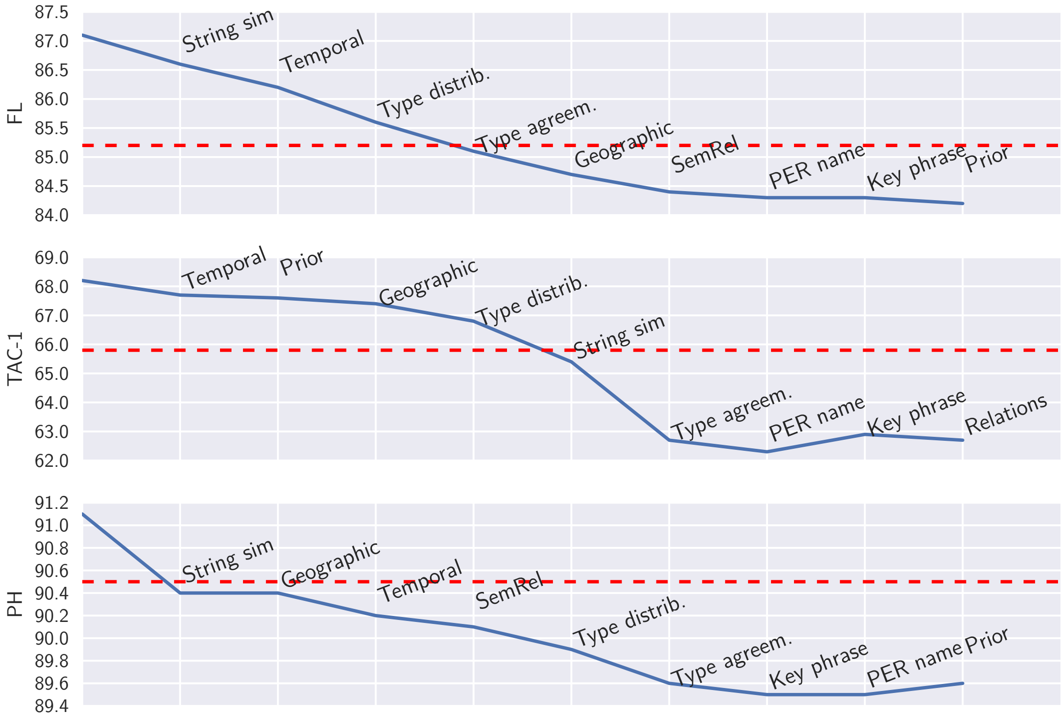


FIGURE 3.13: Feature set ablations for the FL, TAC-1, and PH systems. The solid blue lines show the performance impact in terms of `strong_link_match` $F1$ incurred from eliminating feature sets. The red dashed line indicates baseline performance without verification.

Our proposed global coherence features are among the top features for all systems. This shows that exploiting global coherence has a considerable impact on entity linking performance, and contradicts prior findings by (Ratinov et al., 2011), who did not observe improvements when using global instead of local inference. We believe that this improvement is due to our proposed coherence features being more informative than the generic semantic relatedness measures used in prior work. Our ablation study gives evidence for this, as it shows a relatively low importance of generic semantic relatedness features, which are grouped under the name SemRel in Figure 3.13.

3.2.10 Automatic Verification on Noisy Text

The TAC15 dataset consists of different text genres: "clean" newswire articles, and "noisy" discussion forum threads. Analysis of verification performance on these two genres reveals that verification has the biggest impact on noisy text (Table 3.13 on the facing page, bottom), while the improvement is smaller for two systems on

		Baseline			After verification			Δ		
Genre	System	Prec.	Rec.	$F1$	Prec.	Rec.	$F1$	Prec.	Rec.	F_1
News	TAC-1	66.5	**60.3**	63.2	**75.8**	57.0	**65.0**	9.3	-3.3	1.8
	TAC-2	69.7	**59.9**	**64.4**	**79.3**	53.9	64.2	9.6	-6.0	-0.2
	TAC-3	63.0	**59.1**	61.0	**71.3**	54.3	**61.7**	8.3	-4.8	0.7
Forum	TAC-1	76.0	**61.8**	68.1	**87.4**	60.0	**71.2**	11.4	-1.8	3.1
	TAC-2	73.1	**56.1**	63.5	**83.0**	52.8	**64.6**	9.9	-3.3	1.1
	TAC-3	73.8	**52.4**	61.3	**84.7**	49.9	**62.8**	10.9	-2.5	1.5

TABLE 3.13: Verification on different text genres. See caption of Table 3.11 on page 58 for details.

clean text, and even slightly negative for one system, namely the global inference system TAC-2 (Table 3.13 , top).

3.2.11 Comparison of Automatic Verification to Related Methods

Global coherence has been successfully exploited for entity linking in a number of seminal works (Kulkarni et al., 2009; Hoffart, Yosef, et al., 2011; Han et al., 2011), and more recently by (Moro et al., 2014), (Pershina et al., 2015), and (Globerson et al., 2016), among others. These approaches maximize global coherence based on a general notion of semantic relatedness, while considering a fixed number of candidate entities for each mention. Our approach differs from these in in two regards. Firstly, we introduce specific aspects of coherence, namely entity type coherence, geographic coherence, and temporal coherence. While these aspects are limited to certain entities, such as entities with a clearly defined location and temporal range, our experiments showed that features based on these notions of coherence are useful on the types of texts found in common datasets. Secondly, in an automatic verification setting, these rich coherence measures can be efficiently incorporated since their computational complexity scales linearly in the number of entities mentioned in a document, while they would be prohibitively expensive in the global inference entity linking setting.

Entity types have been used in prior work. Cucerzan (2007) maximizes the agreement of Wikipedia categories associated with candidate entities. Due to intractability of the resulting global optimization problem, the agreement of the candidate entities for a given mention is maximized with respect to all categories of all candidate entities of all other mentions, and hence includes many wrong categories. Our approach is more precise, since verification of entity linking system output allows using only the types of the top-ranked candidate entities. Sil and Yates (2013) also employ entity types, but only maximize type agreement of entity mentions in a small context window. In contrast, our approach uses global context and hence allows capturing long-distance relations.

Post-processing of entity linking system output has been approached as an ensembling task (Rajani and Mooney, 2016). In this setting, a meta-classifier combines the output of different entity linking systems on a given dataset, taking into account features such as system confidence scores, past system performance, and number of systems agreeing with a given decision. Our approach differs from ensembling, since we post-process the output of a single system, using rich semantic features. In contrast, ensembling requires multiple system outputs and relies on meta-information about system performance and decision confidence. Combining these two post-processing methods is an interesting problem for future work and could lead to further improvements, since the two methods rely on different types of information.

3.2.12 Summary

We have introduced automatic verification as a post-processing step for entity linking. Our method uses the output of an existing entity linking system to create a semantic profile of the given text using entity types, as well as geographic and temporal information. Due to the high precision achieved by state-of-the-art entity linking systems, this profile is a sufficiently accurate representation of the text's main topic, and further situates the text temporally and geographically This profile is then used to automatically verify each linked mention individually, i.e., to predict whether it has been linked correctly or not. Verification allows leveraging a rich set of global and pairwise features that would be prohibitively expensive for entity linking systems employing global inference. Evaluation showed consistent improvements when applying our method to seven different entity linking systems on two different datasets.

As the experimental results show, entity linking systems still commit a large number of mistakes – even with automatic verification. In the next section we introduce a visualization tool that allows researchers to inspect and analyze such entity linking mistakes.

Document ID	Start	End	Link ID	Confidence	Entity type
AFP_ENG_20090605.0363	903	914	E0455080	1.0	PER
AFP_ENG_20090605.0363	1021	1025	E0360581	1.0	PER
AFP_ENG_20091014.0713	73	80	NIL0011	1.0	LOC
AFP_ENG_20091014.0713	97	102	E0162920	1.0	PER
AFP_ENG_20091014.0713	191	198	NIL0165	1.0	GPE

TABLE 3.14: An example of entity linking system output.

TP	FP	FN	Precision	Recall	F_1	**measure**
1696	1642	1608	0.508	0.513	0.511	link_match
1646	774	1285	0.680	0.562	0.615	nil_match
3342	2416	2893	0.580	0.536	0.557	all_match
4700	1058	1531	0.816	0.754	0.784	mention_match
2144	797	1997	0.729	0.518	0.606	muc
3015	2742	3966	0.524	0.364	0.429	b_cubed+

TABLE 3.15: Example of evaluation script output, showing aggregate statistics for various metrics, as well as counts of true positives (TP), false positives (FP), and false negatives (FN).

3.3 Visual Entity Explorer: A Tool for Analysis of Entity Linking Errors

In this section, we present the Visual Entity Explorer (VEX), an interactive tool for visualizing and analyzing the output of entity linking systems. VEX is designed to aid researchers and system developers in improving their systems by visualizing system results, gold annotations, and various mention detection and entity linking error types in a clear, concise, and customizable manner.

3.3.1 Motivation

Entity linking systems take a text document as input, detect and link entity mentions, and then produce output in the form of character offset spans and knowledge base IDs. An example of such entity linking system output is shown in Table 3.14 .

To evaluate the performance of an entity linking system, this output is then read by an evaluation script, which calculates aggregate statistics using various metrics, as shown in Table 3.15 .

The increasing interest in entity linking, reflected in the emergence of shared tasks such as the TAC Entity Discovery and Linking track (Ji, Nothman, and Hachey, 2014), ERD 2014 (Carmel et al., 2014), and NEEL (Cano et al., 2014), has fostered research on evaluation metrics for entity linking systems, leading to the development of a dedicated scoring tool that covers different aspects of entity linking system results using multiple metrics (Hachey, Nothman, et al., 2014).

Based on the observation that representations in entity linking (mentions linked to the same KB entry) are very similar to those encountered in coreference resolution (mentions linked by coreference relations to the same entity), these metrics include metrics originally proposed for evaluation of coreference resolutions systems, such as the MUC score (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), and *CEAF* (X. Luo, 2005) and variants thereof (Cai and Strube, 2010).

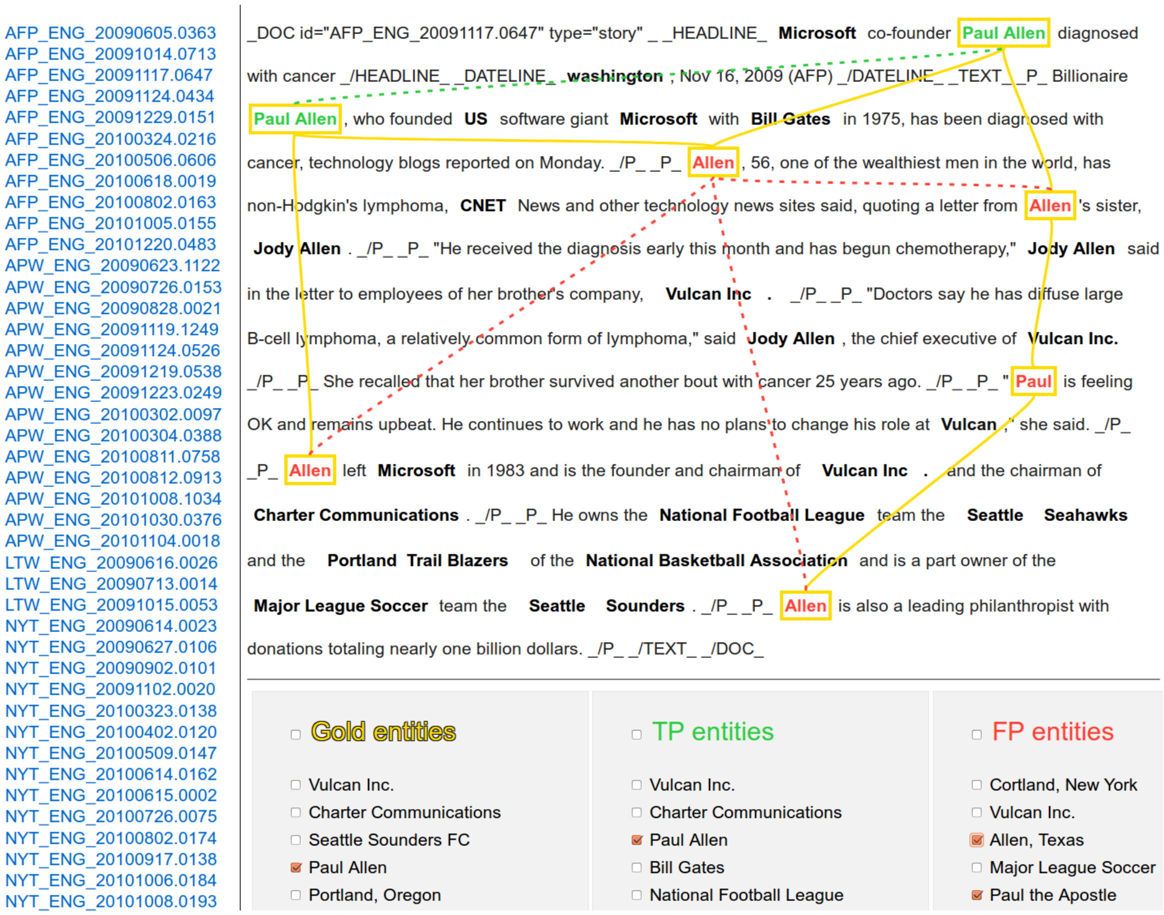
While such metrics, which express system performance in numeric terms of precision, recall, and F_1 scores, are well-suited for comparing systems, they are of limited use to entity linking system developers trying to identify problem areas and components whose improvement will likely result in the largest performance increase. To address this problem, we present the Visual Entity Explorer an interactive tool for visually exploring the results produced by an entity linking system. To our knowledge, there exist no other dedicated tools for visualizing the output of entity linking systems.

3.3.2 The Visual Entity Explorer

After loading system results and gold standard annotations in JSON format or TAC format shown in Table 3.15 on the preceding page, as well as the original document text files, VEX displays gold annotations, correct results, and errors as shown in Figure 3.14 on the next page. The document to be analyzed can be selected via the clickable list of document IDs on the left. Located bottom right, the entity selectors for gold, true positive, and false positive entities (defined below) can be used to toggle the display of individual entities[16]. The selected entities are visualized in the top-right main area.

Similarly to the usage in coreference resolution, where a cluster of mentions linked by coreference relations is referred to as an *entity*, by saying *entity* we mean a cluster of mentions clustered either implicitly by being linked to the same knowledge base entry in case of linkable mentions or clustered explicitly by performing NIL clustering, in case of NIL mentions.

[16] For space reasons, the entity selectors are shown only partially.



AFP_ENG_20090605.0363
AFP_ENG_20091014.0713
AFP_ENG_20091117.0647
AFP_ENG_20091124.0434
AFP_ENG_20091229.0151
AFP_ENG_20100324.0216
AFP_ENG_20100506.0606
AFP_ENG_20100618.0019
AFP_ENG_20100802.0163
AFP_ENG_20101005.0155
AFP_ENG_20101220.0483
APW_ENG_20090623.1122
APW_ENG_20090726.0153
APW_ENG_20090828.0021
APW_ENG_20091119.1249
APW_ENG_20091124.0526
APW_ENG_20091219.0538
APW_ENG_20091223.0249
APW_ENG_20100302.0097
APW_ENG_20100304.0388
APW_ENG_20100811.0758
APW_ENG_20100812.0913
APW_ENG_20101008.1034
APW_ENG_20101030.0376
APW_ENG_20101104.0018
LTW_ENG_20090616.0026
LTW_ENG_20090713.0014
LTW_ENG_20091015.0053
NYT_ENG_20090614.0023
NYT_ENG_20090627.0106
NYT_ENG_20090902.0101
NYT_ENG_20091102.0020
NYT_ENG_20100323.0138
NYT_ENG_20100402.0120
NYT_ENG_20100509.0147
NYT_ENG_20100614.0162
NYT_ENG_20100615.0002
NYT_ENG_20100726.0075
NYT_ENG_20100802.0174
NYT_ENG_20100917.0138
NYT_ENG_20101006.0184
NYT_ENG_20101008.0193

_DOC id="AFP_ENG_20091117.0647" type="story" _ _HEADLINE_ **Microsoft** co-founder Paul Allen diagnosed with cancer _/HEADLINE_ _DATELINE_ **washington** , Nov 16, 2009 (AFP) _/DATELINE_ _TEXT_ _P_ Billionaire Paul Allen , who founded **US** software giant **Microsoft** with **Bill Gates** in 1975, has been diagnosed with cancer, technology blogs reported on Monday. _/P_ _P_ Allen , 56, one of the wealthiest men in the world, has non-Hodgkin's lymphoma, **CNET** News and other technology news sites said, quoting a letter from Allen 's sister, **Jody Allen** . _/P_ _P_ "He received the diagnosis early this month and has begun chemotherapy," **Jody Allen** said in the letter to employees of her brother's company, **Vulcan Inc** . _/P_ _P_ "Doctors say he has diffuse large B-cell lymphoma, a relatively common form of lymphoma," said **Jody Allen** , the chief executive of **Vulcan Inc.** _/P_ _P_ She recalled that her brother survived another bout with cancer 25 years ago. _/P_ _P_ " Paul is feeling OK and remains upbeat. He continues to work and he has no plans to change his role at **Vulcan** ," she said. _/P_ _P_ Allen left **Microsoft** in 1983 and is the founder and chairman of **Vulcan Inc** . and the chairman of **Charter Communications** . _/P_ _P_ He owns the **National Football League** team the **Seattle Seahawks** and the **Portland Trail Blazers** of the **National Basketball Association** and is a part owner of the **Major League Soccer** team the **Seattle Sounders** . _/P_ _P_ Allen is also a leading philanthropist with donations totaling nearly one billion dollars. _/P_ _/TEXT_ _/DOC_

☐ Gold entities
☐ Vulcan Inc.
☐ Charter Communications
☐ Seattle Sounders FC
☑ Paul Allen
☐ Portland, Oregon

☐ TP entities
☐ Vulcan Inc.
☐ Charter Communications
☑ Paul Allen
☐ Bill Gates
☐ National Football League

☐ FP entities
☐ Cortland, New York
☐ Vulcan Inc.
☑ Allen, Texas
☐ Major League Soccer
☑ Paul the Apostle

FIGURE 3.14: Screenshot of the main display. The main display is split into document list (left), entity selectors (bottom right), and the annotated document text (top right).

3.3.3 Visualizing Entity Linking Errors

Errors committed by an entity linking system can be broadly categorized into mention detection errors and linking/clustering errors. Mention detection errors, in turn, can be divided into partial errors and full errors, depending on whether there is partial text span overlap between a system mention and a gold mention, or no overlap at all.

Partial Mention Detection Errors

A partial mention detection error is a system mention span that overlaps but is not identical to any gold mention span. In VEX, partial mention detection errors are displayed using red square brackets, either inside or outside the gold mention spans signified by golden-bordered rectangles, as exemplified by the first and last mention in Figure 3.15 on the facing page.

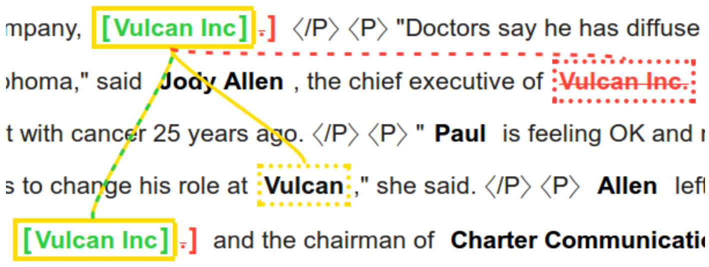
Full Mention Detection Errors

A full mention detection error is either (a) a system mention span that has no overlapping gold mention span at all, corresponding to a false positive detection, i.e. a precision error, or (b) a gold mention span that has no overlap with any system mention span, corresponding to a false negative detection, i.e. a recall error. In VEX, false positive mention detections are marked by a dashed red border and struck-out red text, such as the second mention in Figure 3.15. False negative mention detections are marked with a dashed gold-colored border and black text, for example, the third mention in Figure 3.15 on the next page. For further emphasis, both gold and system mentions are displayed in bold font.

Linking/Clustering Errors

Entities identified by the system are categorized into *true positive* and *false positive* entities. A true positive entity is an entity that has been correctly identified by the system. A false positive entity has been identified by the system as occurring in the document even though no such entity is actually mentioned in the document.

The mentions of system entities are connected using dashed green lines to indicate true positive entities and dashed red lines indicate false positive entities, while gold entity mentions are connected by solid gold-colored lines. This choice of line styles prevents loss of information through occlusion in case of two lines connecting the same pair of mentions, as is the case with the first and last mention in Figure 3.15.



mpany, [Vulcan Inc] .] ⟨/P⟩ ⟨P⟩ "Doctors say he has diffuse

›homa," said Jody Allen , the chief executive of ~~Vulcan Inc.~~

t with cancer 25 years ago. ⟨/P⟩ ⟨P⟩ " Paul is feeling OK and ı

s to change his role at Vulcan ," she said. ⟨/P⟩ ⟨P⟩ Allen lefi

[Vulcan Inc] .] and the chairman of Charter Communicati

FIGURE 3.15: Visualization of various mention detection and entity linking error types (see subsection 3.3.2 on page 64 for a detailed description).

Additionally, the text of system mentions linked to the correct knowledge base entry or identified correctly as NIL is colored green and any text associated with erroneous system entity links is colored red.

3.3.4 Usage examples

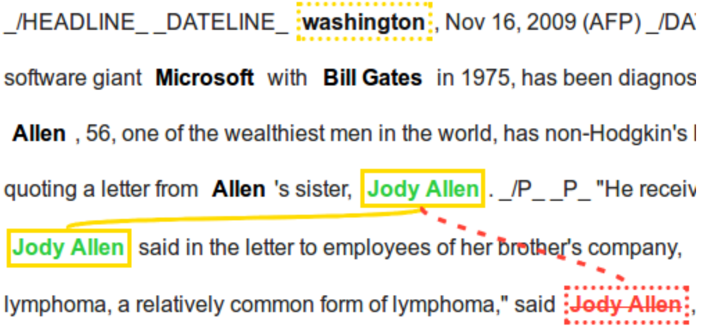
In this section we show how VEX can be used to perform a visual error analysis, gaining insights that arguably cannot be attained by relying only on evaluation metrics.

Example 1

Figure 3.15 shows mentions of VULCAN INC. as identified by an entity linking system (marked red and green) and the corresponding gold annotation, highlighted in gold color.[17] Of the three gold mentions, two were detected and linked correctly by the system and are thus colored green and connected with a green dashed line. One gold mention is surrounded with a gold-colored dashed box to indicate a false negative mention not detected by the system at all. The dashed red box signifies a false positive entity, resulting from the system having detected a mention that is not listed in the gold standard. However, rather than a system error, this is arguably an annotation mistake.

Inspection of other entities and other documents reveals that spurious false positives caused by gold annotation errors appear to be a common occurrence. See Figure 3.16 on the next page for another example of an gold error. Since the supervised machine learning algorithms commonly used for named entity recognition, such as Conditional Random Fields (Sutton and McCallum, 2007), require consistent training data, these inconsistencies hamper performance.

[17] The gold annotations are taken from the `TAC 2014 EDL Evaluation Queries and Links (V1.1)` dataset (Ji, Nothman, and Hachey, 2014).



/HEADLINE _DATELINE_ **washington** , Nov 16, 2009 (AFP) _/DA
software giant **Microsoft** with **Bill Gates** in 1975, has been diagnos
Allen , 56, one of the wealthiest men in the world, has non-Hodgkin's l
quoting a letter from **Allen** 's sister, Jody Allen . _/P_ _P_ "He receiv
Jody Allen said in the letter to employees of her brother's company,
lymphoma, a relatively common form of lymphoma," said ~~Jody Allen~~ ,

FIGURE 3.16: Visualization showing a mention detection error and an annotation error (see subsection 3.3.4 on the preceding page for a description).

Example 2

From Figure 3.15 on the preceding page we can also tell that two mention detection errors are caused by the inclusion of sentence-final punctuation that has a double function as abbreviation marker. The occurrence of similar cases in other documents in this dataset, such as inconsistent annotation of “U.S.” and “U.S” as mentions of UNITED STATES, shows the need for consistently applied annotation guidelines.

Example 3

Another type of mention detection error is shown in Figure 3.16 : Here the system fails to detect washington as a mention of WASHINGTON, D.C., likely due to the non-standard lower-case spelling.

Example 4

The visualization of the gold mentions of PAUL ALLEN in Figure 3.14 on page 65 shows that the entity linking system simplistically partitioned and linked the mentions according to string match, resulting in three system entities, of which only the first, consisting of the two Paul Allen mentions, is a true positive. Even though the four Allen mentions in Figure 3.14 on page 65 align correctly with gold mentions, they are categorized as a false positive entity, since the system erroneously linked them to the knowledge base entry for the city of Allen, Texas. This results in a system entity that does not intersect with any gold entity. The system commits a similar mistake for the mention Paul.

3.3.5 Insights Gained from Error Analysis

This analysis of only a few examples has already revealed several categories of errors, either committed by the entity linking system or resulting from gold annotation mistakes:

- Mention detection errors due to non-standard letter case. Such errors suggest incorporating truecasing (Lita et al., 2003) and/or a caseless named entity recognition model (Manning et al., 2014) into the mention detection process could improve performance.

- Mention detection errors due to off-by-one errors involving punctuation. Such errors show the need for clear and consistently applied annotation guidelines, which would allow developers to add hard-coded, task-specific post-processing rules for dealing with such cases.

- Mention detection errors due to missing gold standard annotations. Such errors suggest performing a simple string match against already annotated mentions to find cases of un-annotated mentions could significantly improve the gold standard at little cost.

- Linking and NIL clustering errors due to the overly strong influence of features based on string match with Wikipedia article titles. For one of the systems analyzed, string match with Wikipedia article titles appeared to outweigh features designed to encourage clustering of mentions if there exists a substring match between them, hence leading to an erroneous partitioning of the gold entity by its various surface forms.

3.3.6 Implementation

In this section we describe the implementation of VEX and some of the design decisions made to create an entity visualization suited for convenient error analysis.

VEX is divided into three main components. The input component, implemented in Java 8, reads gold and system annotation files, as well as the original documents. Currently, the annotation format read by the official TAC 2014 scorer[18], as well as a simple JSON input format are supported. All system and gold character offset ranges contained in the input files are converted into HTML spans and inserted into the document text. Since HTML elements are required to conform to

[18] http://github.com/wikilinks/neleval

a tree structure, any overlap or nesting of spans is handled by breaking up such spans into non-overlapping subspans.

At this point, gold NIL clusters and system NIL clusters are aligned by employing the Kuhn-Munkres algorithm[19] (Kuhn, 1955; Munkres, 1957), as is done in calculation of the *CEAF* metric (X. Luo, 2005) which is part of the TAC 2014 scorer. The input component then stores all inserted, non-overlapping spans in an in-memory database.

The processing component queries gold and system entity data for each document and inventorizes all errors of interest. The data collected by this component is added to the respective HTML spans in the form of CSS classes, enabling simple customization of the visualization via a plain-text style sheet.

The `output` component employs a template engine[20] to convert the data collected by the `processing` component into HTML and JavaScript for handling display and user interaction in the web browser.

3.3.7 Design Decisions

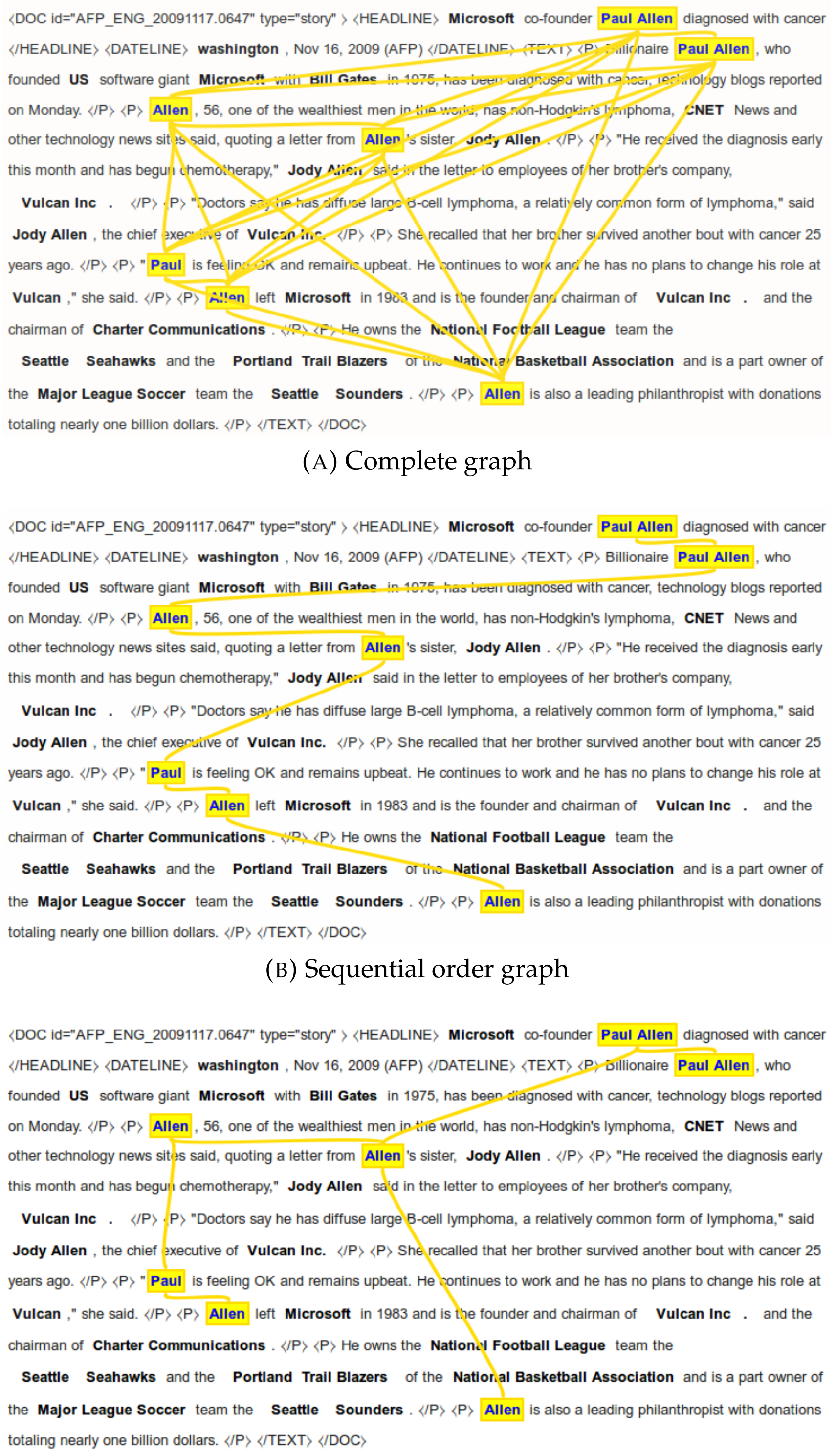
The main design goal of VEX is enabling the user to quickly identify entity linking and clustering errors. Because a naive approach to entity visualization by drawing edges between all possible pairings of mention spans quickly leads to a cluttered graph as shown in 3.17a on the facing page, we instead visualize entities using Euclidean minimum spanning trees, inspired by Martschat and Strube's 2014 use of spanning trees in error analysis for coreference resolution.

An Euclidean minimum spanning tree is a minimum spanning tree of a graph whose vertices represent points in a metric space and whose edge weights are the spatial distances between points. In our case, the metric space is the two-dimensional pixel space in which the HTML document is rendered by the web browser, a point is the top-left corner of a text span element, and the distance metric is the pixel distance between the top-left corners of text span elements. Since the minimum spanning tree spans all graph vertices while minimizing total edge length, it allows for a more concise visualization as shown in 3.17c on the next page.

Since the actual positions of mention span elements on the user's screen depend on various user environment factors such as font size and browser window dimensions, the minimum spanning trees of displayed entities are computed in

[19] Also known as Hungarian algorithm.

[20] `https://github.com/jknack/handlebars.java`



‹DOC id="AFP_ENG_20091117.0647" type="story" › ‹HEADLINE› **Microsoft** co-founder **Paul Allen** diagnosed with cancer ‹/HEADLINE› ‹DATELINE› **washington** , Nov 16, 2009 (AFP) ‹/DATELINE› ‹TEXT› ‹P› Billionaire **Paul Allen** , who founded **US** software giant **Microsoft** with **Bill Gates** in 1975, has been diagnosed with cancer, technology blogs reported on Monday. ‹/P› ‹P› **Allen** , 56, one of the wealthiest men in the world, has non-Hodgkin's lymphoma, **CNET** News and other technology news sites said, quoting a letter from **Allen** 's sister, **Jody Allen** . ‹/P› ‹P› "He received the diagnosis early this month and has begun chemotherapy," **Jody Allen** said in the letter to employees of her brother's company, **Vulcan Inc** . ‹/P› ‹P› "Doctors say he has diffuse large B-cell lymphoma, a relatively common form of lymphoma," said **Jody Allen** , the chief executive of **Vulcan Inc.** ‹/P› ‹P› She recalled that her brother survived another bout with cancer 25 years ago. ‹/P› ‹P› " **Paul** is feeling OK and remains upbeat. He continues to work and he has no plans to change his role at **Vulcan** ," she said. ‹/P› ‹P› **Allen** left **Microsoft** in 1983 and is the founder and chairman of **Vulcan Inc** . and the chairman of **Charter Communications** . ‹/P› ‹P› He owns the **National Football League** team the **Seattle Seahawks** and the **Portland Trail Blazers** of the **National Basketball Association** and is a part owner of the **Major League Soccer** team the **Seattle Sounders** . ‹/P› ‹P› **Allen** is also a leading philanthropist with donations totaling nearly one billion dollars. ‹/P› ‹/TEXT› ‹/DOC›

(A) Complete graph

‹DOC id="AFP_ENG_20091117.0647" type="story" › ‹HEADLINE› **Microsoft** co-founder **Paul Allen** diagnosed with cancer ‹/HEADLINE› ‹DATELINE› **washington** , Nov 16, 2009 (AFP) ‹/DATELINE› ‹TEXT› ‹P› Billionaire **Paul Allen** , who founded **US** software giant **Microsoft** with **Bill Gates** in 1975, has been diagnosed with cancer, technology blogs reported on Monday. ‹/P› ‹P› **Allen** , 56, one of the wealthiest men in the world, has non-Hodgkin's lymphoma, **CNET** News and other technology news sites said, quoting a letter from **Allen** 's sister, **Jody Allen** . ‹/P› ‹P› "He received the diagnosis early this month and has begun chemotherapy," **Jody Allen** said in the letter to employees of her brother's company, **Vulcan Inc** . ‹/P› ‹P› "Doctors say he has diffuse large B-cell lymphoma, a relatively common form of lymphoma," said **Jody Allen** , the chief executive of **Vulcan Inc.** ‹/P› ‹P› She recalled that her brother survived another bout with cancer 25 years ago. ‹/P› ‹P› " **Paul** is feeling OK and remains upbeat. He continues to work and he has no plans to change his role at **Vulcan** ," she said. ‹/P› ‹P› **Allen** left **Microsoft** in 1983 and is the founder and chairman of **Vulcan Inc** . and the chairman of **Charter Communications** . ‹/P› ‹P› He owns the **National Football League** team the **Seattle Seahawks** and the **Portland Trail Blazers** of the **National Basketball Association** and is a part owner of the **Major League Soccer** team the **Seattle Sounders** . ‹/P› ‹P› **Allen** is also a leading philanthropist with donations totaling nearly one billion dollars. ‹/P› ‹/TEXT› ‹/DOC›

(B) Sequential order graph

‹DOC id="AFP_ENG_20091117.0647" type="story" › ‹HEADLINE› **Microsoft** co-founder **Paul Allen** diagnosed with cancer ‹/HEADLINE› ‹DATELINE› **washington** , Nov 16, 2009 (AFP) ‹/DATELINE› ‹TEXT› ‹P› Billionaire **Paul Allen** , who founded **US** software giant **Microsoft** with **Bill Gates** in 1975, has been diagnosed with cancer, technology blogs reported on Monday. ‹/P› ‹P› **Allen** , 56, one of the wealthiest men in the world, has non-Hodgkin's lymphoma, **CNET** News and other technology news sites said, quoting a letter from **Allen** 's sister, **Jody Allen** . ‹/P› ‹P› "He received the diagnosis early this month and has begun chemotherapy," **Jody Allen** said in the letter to employees of her brother's company, **Vulcan Inc** . ‹/P› ‹P› "Doctors say he has diffuse large B-cell lymphoma, a relatively common form of lymphoma," said **Jody Allen** , the chief executive of **Vulcan Inc.** ‹/P› ‹P› She recalled that her brother survived another bout with cancer 25 years ago. ‹/P› ‹P› " **Paul** is feeling OK and remains upbeat. He continues to work and he has no plans to change his role at **Vulcan** ," she said. ‹/P› ‹P› **Allen** left **Microsoft** in 1983 and is the founder and chairman of **Vulcan Inc** . and the chairman of **Charter Communications** . ‹/P› ‹P› He owns the **National Football League** team the **Seattle Seahawks** and the **Portland Trail Blazers** of the **National Basketball Association** and is a part owner of the **Major League Soccer** team the **Seattle Sounders** . ‹/P› ‹P› **Allen** is also a leading philanthropist with donations totaling nearly one billion dollars. ‹/P› ‹/TEXT› ‹/DOC›

(C) Euclidean minimum spanning tree

FIGURE 3.17: Cluttered visualization of an entity via its complete graph, drawing all pairwise connections between mentions (a), less cluttered visualization connecting entity mentions in sequential order (b), and a more concise visualization of the same entity using an Euclidean minimum spanning tree, connecting all mentions while minimizing total edge length (c).

real time using a client-side JavaScript library[21] and are automatically redrawn if the browser window is resized. Drawing of edges is performed via jsPlumb[22], a highly customizable library for line drawing in HTML documents.

In order not to overemphasize mention detection errors when displaying entities, VEX assumes a system mention span to be correct if it has a non-zero overlap with a gold mention span. For example, consider the first gold mention "Vulcan Inc" in Figure 3.15 on page 67, which has not been detected correctly by the system, as it detected "Vulcan Inc." instead. While a strict evaluation requiring perfect mention spans will give no credit at all for this partially correct result, seeing that this mention detection error is already visually signified by the red square bracket, VEX treats the mention as detected correctly for the purpose of visualizing the entity graph, and counts it as a true positive instance if it has been linked correctly.

While VEX provides a carefully chosen default configuration, the visualization style can be easily customized via CSS, e.g., in order to achieve a finer-grained categorization of error types such as off-by-one mention detection errors, or classification of linkable mentions as NIL mentions and vice-versa.

3.3.8 Summary

In this section, we have introduced the Visual Entity Explorer (VEX), a tool for visual error analysis of entity linking systems. We have shown how VEX can be used for quickly identifying the components of an entity linking system that appear to have a high potential for improvement, as well as for finding errors in the gold standard annotations. Since visual error analysis of our own entity linking system revealed several issues and possible improvements, we believe performing such an analysis will prove useful for other developers of entity linking systems, as well.

[21] `https://github.com/abetusk/euclideanmst.js`. This library uses Kruskal's algorithm (Kruskal, 1956) to find minimum spanning trees.

[22] `http://www.jsplumb.org`

Chapter 4

Selectional Preferences for Coreference Resolution

In this chapter, we study the application of another aspect of coherence: the semantic agreement between a predicate and its arguments. This agreement is an essential property of language and is one of the factors that distinguish a coherent text from an incoherent one. For example, in the phrase

(47) the ship sinks

the predicate *sinks* and the subject argument *the ship* agree semantically, since, according to our knowledge of the world, it is plausible, albeit unfortunate, that a ship sinks. In contrast, the phrase

(48) ? the ship writes a dissertation

lacks semantic agreement, since, to the best of our knowledge, ships cannot write, and dissertations cannot be written by inanimate objects.

4.1 Preference and Affordance

The notion of semantic agreement between a predicate and its arguments has mainly been approached from the perspective of the predicate. In this view, a predicate *selects* specific arguments: The predicate *sink* selects subject arguments that are not buoyant – either by default such as stones, or by accident such as ships. Similarly, the predicate *write* selects subjects that can write, such as PhD students. The precise mechanism by which this selection happens has been subject of extensive scholarly debate.[1] In this work we adopt the view that "a predicate preferentially associates with certain kinds of arguments" (Resnik, 1993, p. 53). Resnik calls the preferential association between a predicate and its arguments the predicate's *selectional*

[1] See Resnik (1993) for an overview.

preference. A predicate's selectional preference expresses what kind of arguments it typically chooses. This choice is "less a yes-or-no decision and more a function of how easily the predication can be accommodated given information about word meanings and context" (ibid., p. 59). For example, the predicate *sink* has a high preference for the first of the following subject arguments, but the preference for the subsequent subjects becomes smaller:

(49) the ship sinks

(50) the stone sinks

(51) the person sinks

(52) the airplane sinks

(53) the house sinks

(54) the island sinks

(55) ? the balloon sinks

(56) ? the maple leaf sinks

(57) ? the gas sinks

(58) ?? the election sinks

(59) ?? the choice sinks

(60) ?? the quadratic equation sinks

This gradual change in selectional preference reflects the ease or difficulty of imagining contexts in which a specific pairing of predicate and argument makes sense. There are many contexts imaginable in which a ship, a stone, or a person sink. Less typical, but still plausible are situations involving a sinking airplane, house or island. Not typical, but also not impossible are scenarios that lead to the sinking of a balloon, a maple leaf, or gas. Least preferred are the bottom three subjects, since it is difficult to imagine a context that meaningfully combines an event like an election or abstract concepts like *choice* and *quadratic equation* with the literal sense of *sink* that is active here.

Complementary to the analysis of a predicate's preferred arguments is the analysis of an argument's preferred predicates. Instead of asking what kind of object typically *sinks*, we now ask what *a ship typically does*, or what *is typically done to a ship*. Typical things a ship does are:

(61) the ship set sail

(62) the ship arrived

(63) the ship sank

and things that are typically done with or to a ship are:

(64) the ship was built

(65) the ship was christened

(66) passengers boarded the ship

(67) the captain is steering the ship

(68) high waves rocked the ship

Besides (48), examples of things that are not typically done by or with/to ships are:

(69) ? the ship thawed

(70) ? the ship thought

(71) ? the ship was taught algebra

(72) ? the passengers bored the ship

(73) ? the captain is wearing the ship

Our model of selectional preferences, which we introduce in section 4.3 below, captures both a predicate's preferred arguments and an argument's preferred predicates. While the former has an established name, namely *selectional preferences*, the latter does not. Erk and Padó (2008) call what is typically done by/to/with an object the object's *inverse selectional preferences*. Borrowing a term originally coined in perceptual psychology, Attardo (2005) calls it *affordance*. Taking an example by Attardo, the difference in meaning between

(74) He ran to the edge of the cliff and jumped.

and

(75) He ran to the trampoline and jumped.

lies in the difference between what a cliff allows doing and what a trampoline allows doing. The edge of a cliff *affords* jumping off (once, into the sea), while a trampoline affords jumping on it, usually repeatedly.

4.2 Selectional Preferences for Coreference Resolution

Selectional preferences have long been claimed to be useful for coreference resolution. In his seminal work on *Resolving Pronominal References* Hobbs (1978) proposed a semantic approach to coreference solution which requires reasoning about the "demands the predicate makes on its arguments" (p. 328). For identifying the antecedent of the pronoun *its* in

(76) The boy walked into the bank. Moments later he was seen on its roof.

Hobbs gives the following line:

1. *to walk into X* implies that X is a region, location or building
2. *bank* is a financial institution, river bank or building
3. *to walk into a bank* implies that *bank* is a building
4. the fact that buildings have roofs implies that a bank has a roof
5. *its roof* implies that the antecedent of *its* has a roof
6. it follows that the antecedent of *its* is *bank*

Thus the selectional preferences of *to walk into* disambiguate the ambiguous *bank*, which in turn allows connecting *its roof* and *bank*. Realizing that this semantic approach requires world knowledge and complex reasoning, Hobbs also proposed a "naive" *syntactic approach* that traverses the syntactic parse trees of the current and preceding sentences until it finds a noun phrase with matching gender and number. Applied to (76), the following checks find the correct antecedent:

1. *its* does not match *he* due to gender disagreement.
2. *its* does not match *Moments* due to number disagreement.
3. *its* does not match *boy* due to gender disagreement.
4. *its* matches *bank* in gender and number.

While Hobbs readily admits that it is easy to find examples in which it does not work, he also observes that

> [. . .] the naive approach is quite good. Computationally speaking, it will be a long time before a semantically based algorithm is sophisticated enough to perform as well, and these results set a very high standard for any other approach to aim for. Yet there is every reason to pursue a semantically based approach. The naive algorithm does not work. Any one can think of examples where it fails. In these cases it not only fails; it gives no indication that it has failed and offers no help in finding the real antecedent. (p. 324)

A simple example in which a syntactic approach cannot find the correct antecedent is

(77) The Titanic hit an iceberg. It sank quickly.

(78) The Titanic hit an iceberg. It melted quickly.

Here selectional preferences allow resolving the pronoun *it*, since ships sink but icebergs don't, and icebergs melt while ship usually do not.

Research on computational models of selectional preferences has shown considerable progress (Dagan and Itai, 1990; Resnik, 1993; Agirre and Martinez, 2001; Pantel et al., 2007; Erk, 2007; Ritter et al., 2010; Van de Cruys, 2014). However, today's coreference resolution systems (Martschat and Strube, 2015; Wiseman et al., 2016; Clark and Manning, 2016a, i.a.) capture selectional preferences only implicitly at best, for example via a given mention's dependency governor and other contextual features. This lack is our motivation for incorporating a current model of selectional preferences into a state-of-the-art coreference resolution system.

We are not the first to attempt to do so. More than ten years ago, Kehler et al. (2004) integrated selectional preferences into a coreference resolution system. However, they observed only minor improvements on a small dataset and found that these were due to fortuity rather than selectional preferences having captured meaningful world knowledge relations. As a result, Kehler et al. declared the "non-utility of predicate-argument structures for pronoun resolution". This frustration was echoed by later authors. Durrett and Klein (2013) called integrating semantics into coreference resolution an "uphill battle" and Strube (2015) reports that any attempt at incorporating world knowledge into the coreference resolution system by Martschat and Strube (2015) degraded performance: While world knowledge enabled the discovery of more coreference relations, thereby increasing recall, this increase was outweighed by a decrease in precision due to the introduction of spurious connections between non-coreferent entities.

The claim by Kehler et al. is based on selectional preferences extracted from a, by current standards, small number of 2.8 million predicate-argument pairs, resulting in low lexical coverage and susceptibility to noise in the background corpus from which these pairs were collected. Furthermore, they employ a simple maximum entropy classifier, which requires manual definition of feature combinations and is unlikely to fully capture the complex interaction between selectional preferences and other coreference features. Durrett and Klein achieve higher lexical coverage by extracting selectional preferences from a much larger corpus, but use a coarse class inventory comprising only 20 latent clusters, such as "things which announce". In this chapter we attempt to overcome these shortcomings. We propose a fine-grained, high-coverage model of selectional preferences and study its impact on a state-of-the-art coreference resolution system.

4.3 Modeling Selectional Preferences

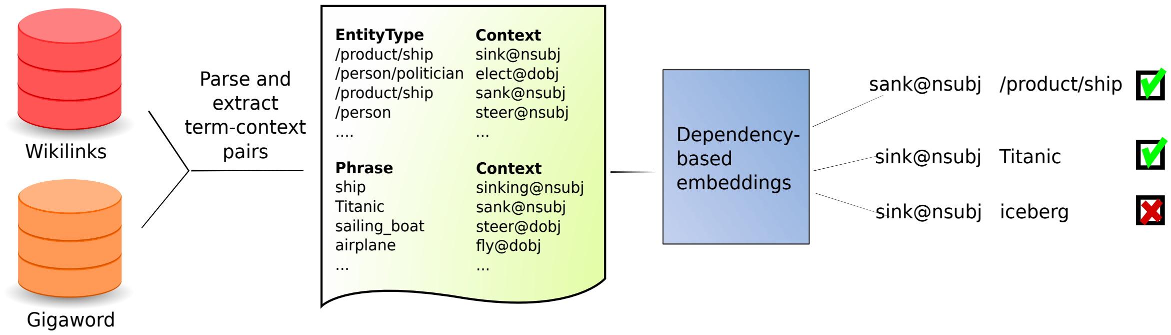
An important design decision to make when modeling selectional preferences is the choice of predicates and arguments. We call a set of predicates and their arguments a *relation inventory*. The relation inventory provides the concepts and entities that can be relation arguments, as well as the predicates that relate arguments to each other. Prior work has studied many relation inventories.

In the simplest case, the relation inventory consists of pairs of words in a certain syntactic relationship, for example a transitive verb and its direct object such as ⟨*eat*, *spaghetti*⟩.[2] Statistics over word-word pairs are easily collected from any syntactically parsed text corpus. The drawback of this approach, however, is that it does not generalize to unseen pairs (Dagan and Itai, 1990), since each word is treated as a discrete symbol with no relation or similarity to other symbols. In a word-word approach, knowledge about the plausibility of ⟨*eat*, *spaghetti*⟩ does not support judging the plausibility of the pair ⟨*eat*, *linguine*⟩.

Class-based approaches aim to overcome this problem by mapping words to semantic classes and then counting word-class pairs (Resnik, 1993). In a word-class approach, *spaghetti* is mapped to a class, e.g. *FOOD*, and all occurrences of ⟨*eat*, *spaghetti*⟩ are counted as instances of ⟨*eat*, *FOOD* ⟩. The plausibility of ⟨*eat*, *linguine*⟩ can then be judged by identifying *linguine* as an instance of *FOOD*. Generalizing further, class-class approaches also map predicates to a class (Agirre and Martinez, 2001). Assuming a taxonomy in which *eat* is an instance of *INGEST*, our example becomes ⟨*INGEST*, *FOOD* ⟩ and we now are able to judge the plausibility of a pair like ⟨*devour*, *linguine*⟩ by identifying it as an instance of the same class-class pair. The main drawbacks of class-based approaches are that they require disambiguation of words to classes and that they are limited by the coverage of the lexical resource providing such classes, such as WordNet.

Word- and class-based approaches of selectional preferences have been studied based on syntactic predicate-argument pairs, namely subject-predicate, e.g. ⟨*ship*, *sinks*⟩, and predicate-object, e.g. ⟨*eat*, *spaghetti*⟩. More expressive relation inventories include semantic representations such as FrameNet frames and roles (Fillmore et al., 2003), event types and arguments, or abstract meaning representation (Banarescu et al., 2013). While these semantic representations are arguably well-suited to model meaningful world knowledge relationships, automatic annotation is limited in speed and accuracy, making it difficult to obtain a large number of such "more semantic" predicate-argument pairs. In comparison, the only requirement

[2] Examples in this sections are taken from Agirre and Martinez (2001).


Wikilinks
Gigaword
Parse and extract term-context pairs
Dependency-based embeddings
sank@nsubj /product/ship ✓
sink@nsubj Titanic ✓
sink@nsubj iceberg ✗

EntityType	Context
/product/ship	sink@nsubj
/person/politician	elect@dobj
/product/ship	sank@nsubj
/person	steer@nsubj
....	...

Phrase	Context
ship	sinking@nsubj
Titanic	sank@nsubj
sailing_boat	steer@dobj
airplane	fly@dobj
...	...

FIGURE 4.1: Dependency-based embedding model of selectional preferences.

for collecting pairs of words in a syntactic relationship is syntactic parsing. The speed and accuracy of freely available syntactic parsers (D. Chen and Manning, 2014) makes it trivial to obtain a large number of accurate, albeit "less semantic" predicate-argument pairs. The drawback of such a syntactic approach to selectional preferences, however, is its susceptibility to lexical and syntactic variation. For example, the two sentences

(79) The Titanic sank.

(80) The ship went under.

differ lexically and syntactically, but would have the same or a very similar representation in a semantic framework such as FrameNet.

Our model of selectional preferences (Figure 4.1) aims to overcome these drawbacks by combining three components:

- a distributed representation of predicate-argument pairs (subsection 4.3.1);
- an inventory of syntactic dependencies that were specifically designed for semantic downstream tasks (subsection 4.3.2); and
- generalization over an important source of lexical variety by resolving named entities to their fine-grained entity types (subsection 4.3.3).

We introduce these components in the following sections.

4.3.1 Distributed Representation of Selectional Preferences

Distributed word representations (Landauer and Dumais, 1997; Bengio et al., 2003; Turian et al., 2010; Mikolov, Sutskever, et al., 2013; Pennington et al., 2014) allow computing semantic similarity between words and thus enable generalization over

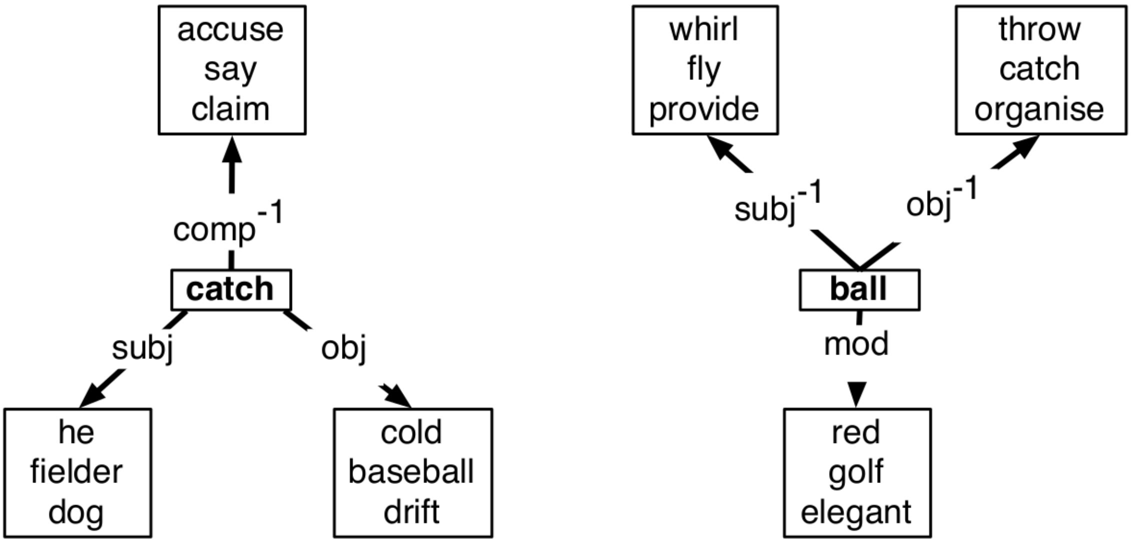


FIGURE 4.2: Erk and Padó's Structured Vector Space model. Image source: Erk and Padó (2008).

lexical variation. For example, the high similarity of the distributed representations of *Titanic* and *ship* allows determining (79) as plausible if we know that

(81) the ship sinks

is plausible and

(82) ? the Titanic writes a dissertation

as implausible if we know that

(83) ? the ship writes a dissertation

is implausible.

Erk and Padó (2008) apply the idea of distributed representation to selectional preferences. Their model, called Structured Vector Space and shown in Figure 4.2, learns vector representations of words using their syntactic dependency context. For example in the dependency context of the word *catch*, words like *he*, *fielder*, and *dog* appear in the subject relation, while words like *cold*, *baseball*, and *drift* appear in the object relation. These words can be interpreted as the selectional preferences for the subject and object slots of the word *catch*. Inverse selectional preferences express what is typically done to or by something. For example, a *ball* may *whirl*, *fly*, and *provide*, or be *thrown*, *caught*, or *organized*.

Inspired by Structured Vector Space, we embed predicates and arguments into a low-dimensional vector space in which (representations of) selectional preferences of predicate slots are close to (representations of) their plausible arguments, as are arguments that tend to fill the same slots of similar predicates, and predicate slots that have similar arguments. For example, *captain* should be close to *pilot*, *ship* to *airplane*, the subject of *steer* close to both *captain* and *pilot*, and also to, say, the subject of *drive*. Such a space allows judging the plausibility of unseen predicate-argument pairs.

We construct this space via dependency-based word embeddings (O. Levy and Goldberg, 2014). To see why this choice is better-suited for modeling selectional preferences than alternatives such as `word2vec` (Mikolov, K. Chen, et al., 2013) or `GloVe` (Pennington et al., 2014), consider the following sentences:

(84) The captain steers the ship.

(85) The pilot steers the airplane.

with the following dependency relations

$$\begin{array}{ccccc} \text{captain} & \xleftarrow{\text{nsubj}} & \text{steers} & \xrightarrow{\text{dobj}} & \text{ship} \\ :: & & & & :: \\ \text{pilot} & \xleftarrow{\text{nsubj}} & \text{steers} & \xrightarrow{\text{dobj}} & \text{airplane} \end{array}$$

Here, *captain* and *ship*, have high syntagmatic similarity, that is, these words are semantically related and tend to occur close to each other. This also holds for *pilot* and *airplane*. In contrast, *captain* and *pilot*, as well as *ship* and *airplane* have high paradigmatic similarity: They are semantically similar and occur in similar contexts. A model of selectional preferences requires paradigmatic similarity. The representations of *captain* and *pilot* in such a model should be similar, since they both can plausibly fill the subject slot of the predicate *steer*. Due to their use of linear context windows, `word2vec` and `GloVe` tend to capture syntagmatic similarity, while dependency-based embeddings capture paradigmatic similarity (see O. Levy and Goldberg, 2014). In the remainder of this chapter we will assume that a predicate such as *steer* has certain "slots" corresponding to syntactic dependency relations it participates in. In the above example *captain* and *pilot* fill the subject slot, which we denote *steer@nsubj*, while *ship* and *airplane* fill the object slot *steer@dobj*.

4.3.2 Enhanced++ Universal Dependencies

Due to the benefits of distributed representation, our model generalizes over syntactic variation such as active/passive alternations: For example, *steer@dobj* is highly

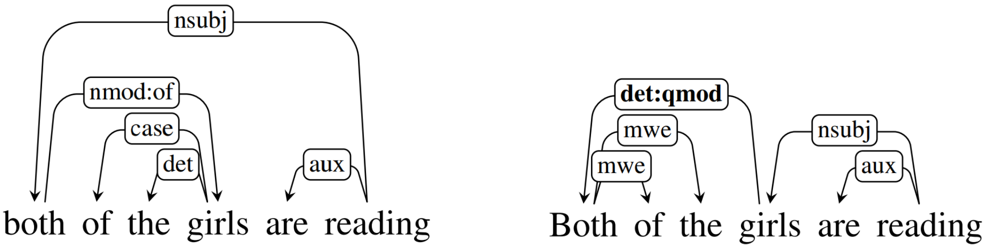


FIGURE 4.3: Basic (left) and enhanced++ (right) universal dependency parse of example (86). Image source: Schuster and Manning (2016).

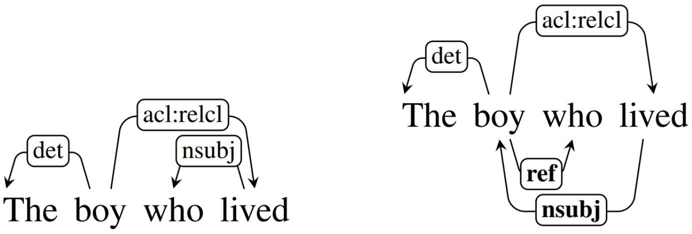


FIGURE 4.4: Basic (left) and enhanced++ (right) universal dependency parse of example (87). Image source: Schuster and Manning (2016).

similar to *steer@nsubjpass* (see Table 4.2 on page 91 for more examples). To further mitigate the effect of employing syntax as a proxy for semantics, we use enhanced++ universal dependencies (Schuster and Manning, 2016). Enhanced++ dependencies support semantic applications by modifying syntactic parse trees to better reflect relations between content words and were found to improve semantic downstream tasks such as event extraction (Silveira, 2016). Taking two examples from Schuster and Manning, the “basic” syntactic dependency parse of the sentence

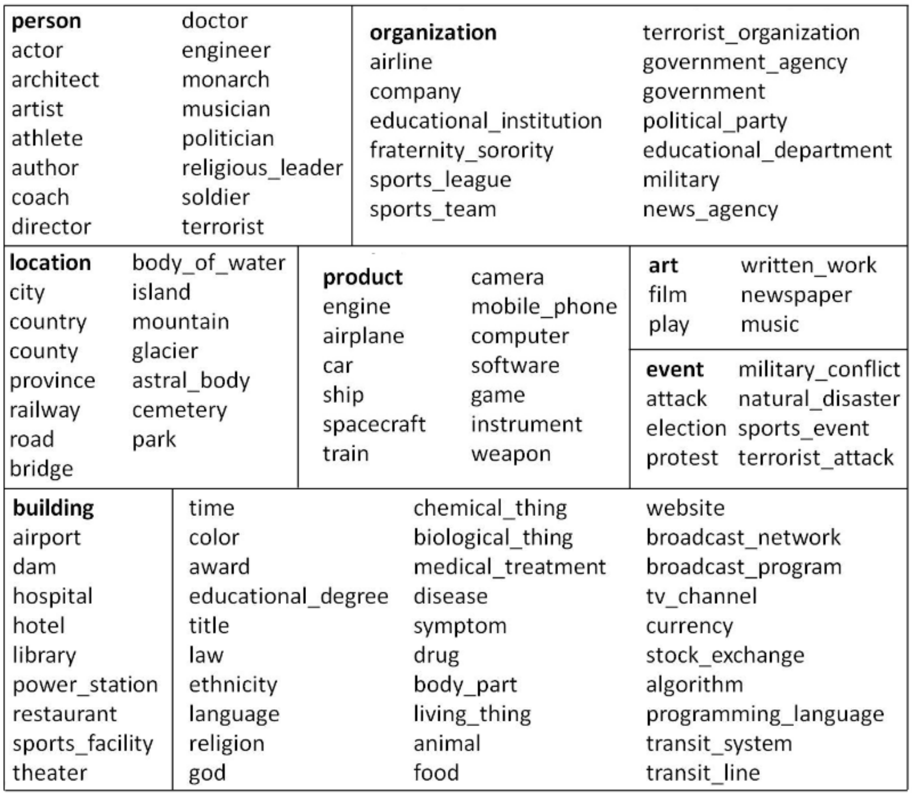
(86) Both of the girls are reading.

identifies *Both* as subject of *reading*. The Enhanced++ representation introduces a subject relation between *girls* and *reading* (Figure 4.3 . This allows learning more meaningful selectional preferences: Our model should learn that girls (and other humans) read, while learning that an unspecified *both* are reading is not helpful.

Another common enhancement concerns relative clauses, as in

(87) The boy who lived

Here the Enhanced++ representation adds a subject relation between *boy* and *lived* (Figure 4.4). The motivation for doing so is analogous to the previous example: It is more informative that a boy lived than an unspecified *who*.



person
actor
architect
artist
athlete
author
coach
director
doctor
engineer
monarch
musician
politician
religious_leader
soldier
terrorist

organization
airline
company
educational_institution
fraternity_sorority
sports_league
sports_team
terrorist_organization
government_agency
government
political_party
educational_department
military
news_agency

location
city
country
county
province
railway
road
bridge
body_of_water
island
mountain
glacier
astral_body
cemetery
park

product
engine
airplane
car
ship
spacecraft
train
camera
mobile_phone
computer
software
game
instrument
weapon

art
film
play
written_work
newspaper
music

event
attack
election
protest
military_conflict
natural_disaster
sports_event
terrorist_attack

building
airport
dam
hospital
hotel
library
power_station
restaurant
sports_facility
theater

time
color
award
educational_degree
title
law
ethnicity
language
religion
god

chemical_thing
biological_thing
medical_treatment
disease
symptom
drug
body_part
living_thing
animal
food

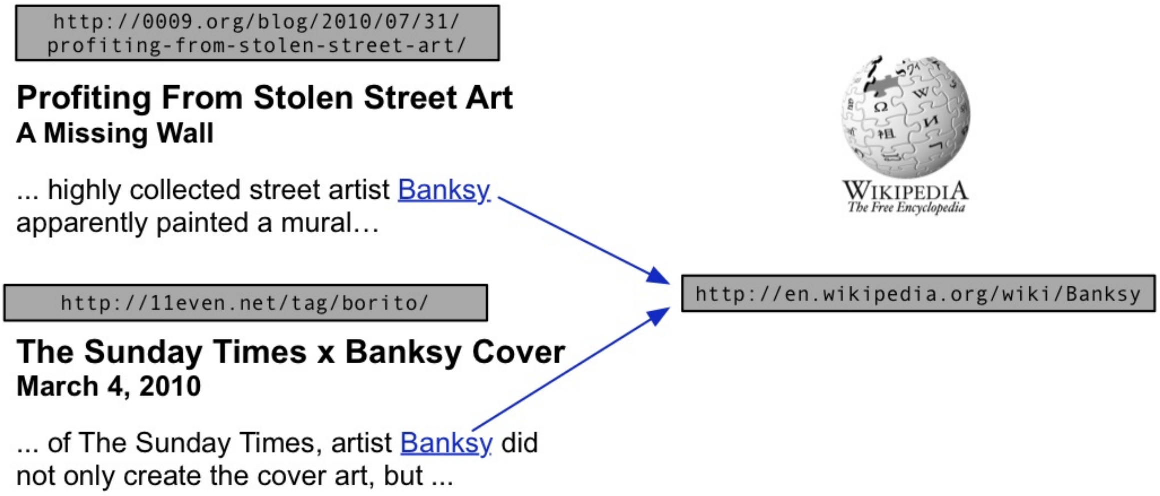
website
broadcast_network
broadcast_program
tv_channel
currency
stock_exchange
algorithm
programming_language
transit_system
transit_line

FIGURE 4.5: The FIGER type inventory we use for generalizing over named entities. Image source: X. Ling and Weld, 2012.

4.3.3 Fine-grained Entity Types.

A good model of selectional preferences needs to generalize over named entities, since named entities are a considerable source of lexical variety.[3] For example, having encountered sentences like *The Titanic sank*, our model should be able to judge the plausibility of an unseen sentence like *The RMS Lusitania sank*. For popular named entities, we can expect the learned representations of *Titanic* and *RMS Lusitania* to be similar, allowing our model to generalize, that is, we expect it to be able to judge the plausibility of *The RMS Lusitania sank* by virtue of the similarity between *Titanic* and *RMS Lusitania*. However, this will not work for rare or emerging named entities, for which no, or only low-quality distributed representations have been learned. To address this issue, we perform fine-grained entity typing using the FIGER inventory proposed by X. Ling and Weld (2012), which is shown in Figure 4.5 . For each named entity encountered during training, we generate an additional training instance by replacing the named entity with its entity type. For

[3] See the discussion of the high prevalence of rare and unknown words in named entities in subsection 5.1.4 and subsection 5.1.5.



http://0009.org/blog/2010/07/31/profiting-from-stolen-street-art/

Profiting From Stolen Street Art

A Missing Wall

... highly collected street artist Banksy apparently painted a mural…

http://11even.net/tag/borito/

The Sunday Times x Banksy Cover

March 4, 2010

... of The Sunday Times, artist Banksy did not only create the cover art, but ...

http://en.wikipedia.org/wiki/Banksy

FIGURE 4.6: Two instances from the Wikilinks corpus. Image source: Singh et al. (2012)

example, the FIGER entity type of *Titanic* is `/product/ship`. With this entity type and the training instance

(88) The Titanic sank.

we generate the additional training instance

(89) The `/product/ship` sank.

4.4 Implementation

We train our model on data mined from two sources. Noun phrases and their dependency context are extracted from GigaWord (Parker et al., 2011) and entity types in context from Wikilinks (Singh et al., 2012). Wikilinks is a large corpus of sentences containing links to Wikipedia articles. Figure 4.6 shows fragments of two such sentences from two different web documents. In both fragments, the mention Banksy links to the Wikipedia article about the anonymous street artist. From a dependency parse of the first fragment we extract dependency relations such as:

compound(Banksy, street)
compound(Banksy, artist)
nsubj(painted, Banksy)
dobj(painted, mural)

Querying the entity type `/person/artist` for *Banksy* from Freebase, we generate pairs of noun phrases or entity types and their dependency context:

(street, Banksy@compound)

Argument	Sel. preference slot
/person/artist	influence@nmod:as
/organization/company	service@nmod:like
/organization/company	Netcom@conj:and
/medicine/symptom	body@nmod:such_as
/disease	Menorrhagia@conj
/broadcast_program	heyday@nsubj
/person/artist	do@nsubj
/person/author	transition@nsubj
April	raise@nmod:from
July	raise@nmod:to
Sun	sentence@nsubjpass
death	sentence@nmod:to
crime	death@nmod:for
fund-raising	crime@nmod:of
life_imprisonment	sentence@nmod:to
imprisonment	sentence@nmod:to

TABLE 4.1: Examples of pairs of arguments and selectional preference slots extracted from the Wikilinks corpus.

(artist, Banksy@compound)
(Banksy, painted@nsubj)
(street, /person/artist@compound)
(artist, /person/artist@compound)
(/person/artist, painted@nsubj)
(mural, painted@dobj)

After parsing each corpus with the `CoreNLP` dependency parser (Manning et al., 2014), we obtain 1.4 billion pairs of noun phrases and the lexicalized dependency relation to their governors and dependents, such as *(Titanic, sank@nsubj)* from GigaWord. From Wikilinks we obtain about 12.9 million pairs of entity types and their dependency context such as *(/product/ship, sank@nsubj)*. A selection of these pairs is listed in Table 4.1 .

Figure 4.7 on the following page shows the distribution of FIGER entity types that occur in the subject slot of the predicate *sank*. Almost 80 percent of subjects have the entity type `/product/ship`. The smaller spike for `/event/natural_disaster` reflects transitive usage as in

(90) The storm sank the ship.

and the spike for `/person/athlete` reveals a different word sense, which is common in Basketball:

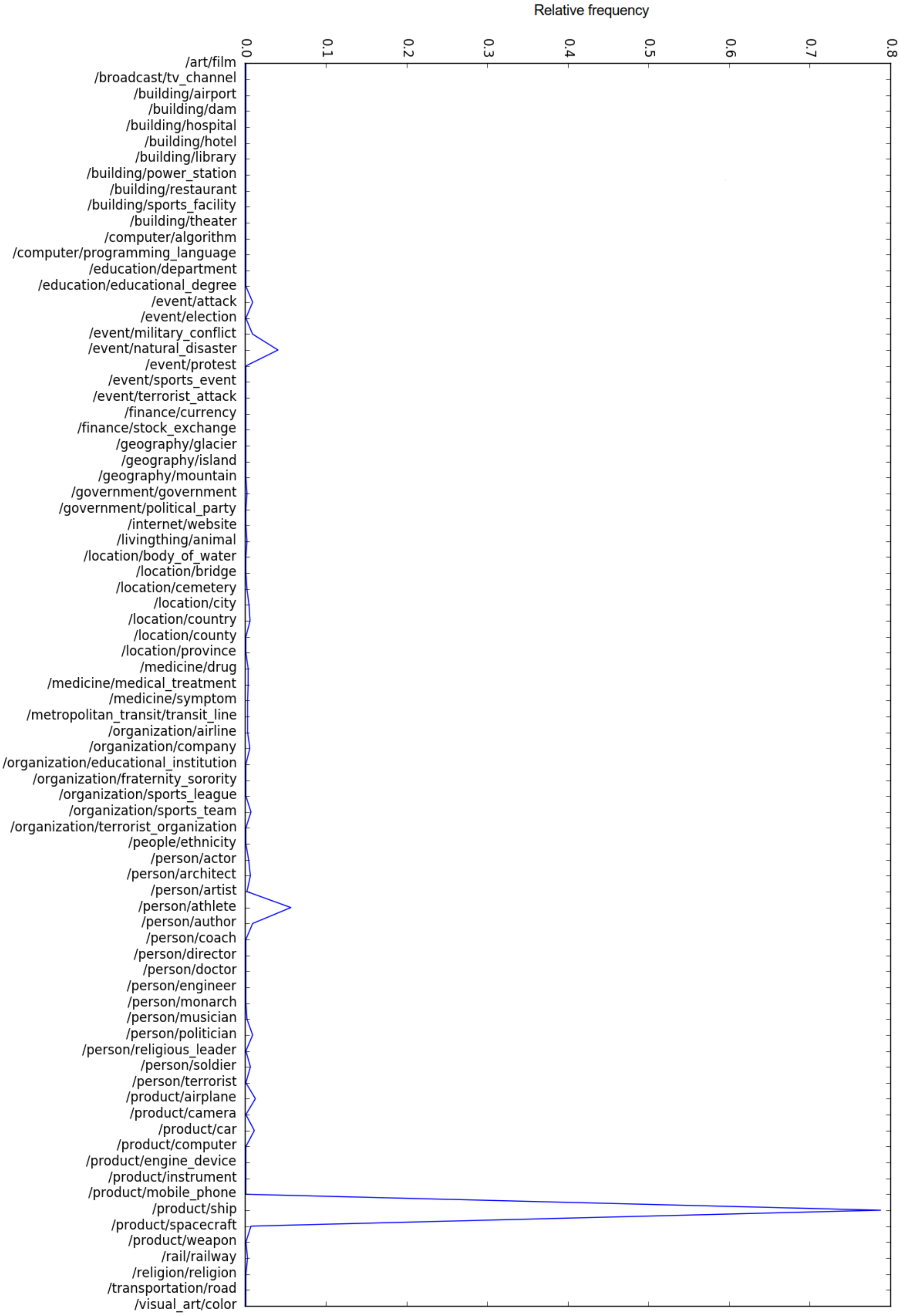


FIGURE 4.7: Empirical entity type distribution for the subject argument of the predicate *sank* in the Wikilinks corpus.

(91) Michael Jordan sank the game-winning shot with only four seconds left on the clock.

Finally, we train dependency-based embeddings using the generalized `word2vec` version by O. Levy and Goldberg (2014), thereby obtaining distributed representations of selectional preferences. To identify fine-grained types of named entities mentions in texts for which no entity annotations exist, we first perform entity linking using the system by (Heinzerling, Judea, et al., 2015), then query Freebase (Bollacker et al., 2008) for entity types and apply the mapping to fine-grained types by (X. Ling and Weld, 2012).

The plausibility of an argument filling a particular predicate slot can now be computed via the cosine similarity of their associated embeddings:

$$\text{Selectional preference similarity} = cos(\boldsymbol{v}_1, \boldsymbol{v}_2) = \frac{\boldsymbol{v}_1 \cdot \boldsymbol{v}_2}{||\boldsymbol{v}_1|| \cdot ||\boldsymbol{v}_2||}$$

where $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are embeddings representing a selectional preference slot or argument. For example, in our trained model, the selectional preference similarity of *(Titanic, sank@nsubj)* is 0.11 while the similarity of *(iceberg, sank@nsubj)* is -0.005, indicating that an iceberg sinking is less plausible.

4.4.1 Qualitative Evaluation

We first perform a qualitative evaluation of our selectional preference embeddings by inspecting visualizations and comparing embedding similarities.

Figure 4.8 on the next page shows a visualization of our selectional preference embeddings. We make two observations. Firstly, embeddings of arguments (dark blue) and selectional preference slots (light blue) are distributed across overlapping regions of the embedding space. This is important, since alternative configurations, such as two distinct clusters of arguments and selectional preference slots, would make it difficult to obtain meaningful similarity scores between argument and selectional preference slot embeddings. Secondly, entity type embeddings (orange) are concentrated in one region of the embedding space. This means that similarity scores between entity type and other embeddings have a different distribution than similarity scores between argument and argument slot embeddings, making comparisons difficult. This problem mirrors results by Gupta et al. (2018), who found that categories and their instances have markedly different distributions.

Figure 4.9 on page 89 gives a closer view of a region of the selectional preference embedding space. On the left side, names of various politicians, e.g. *mandela*, *berlusconi*, *merkel*, are close to selectional preference embeddings that represent some of

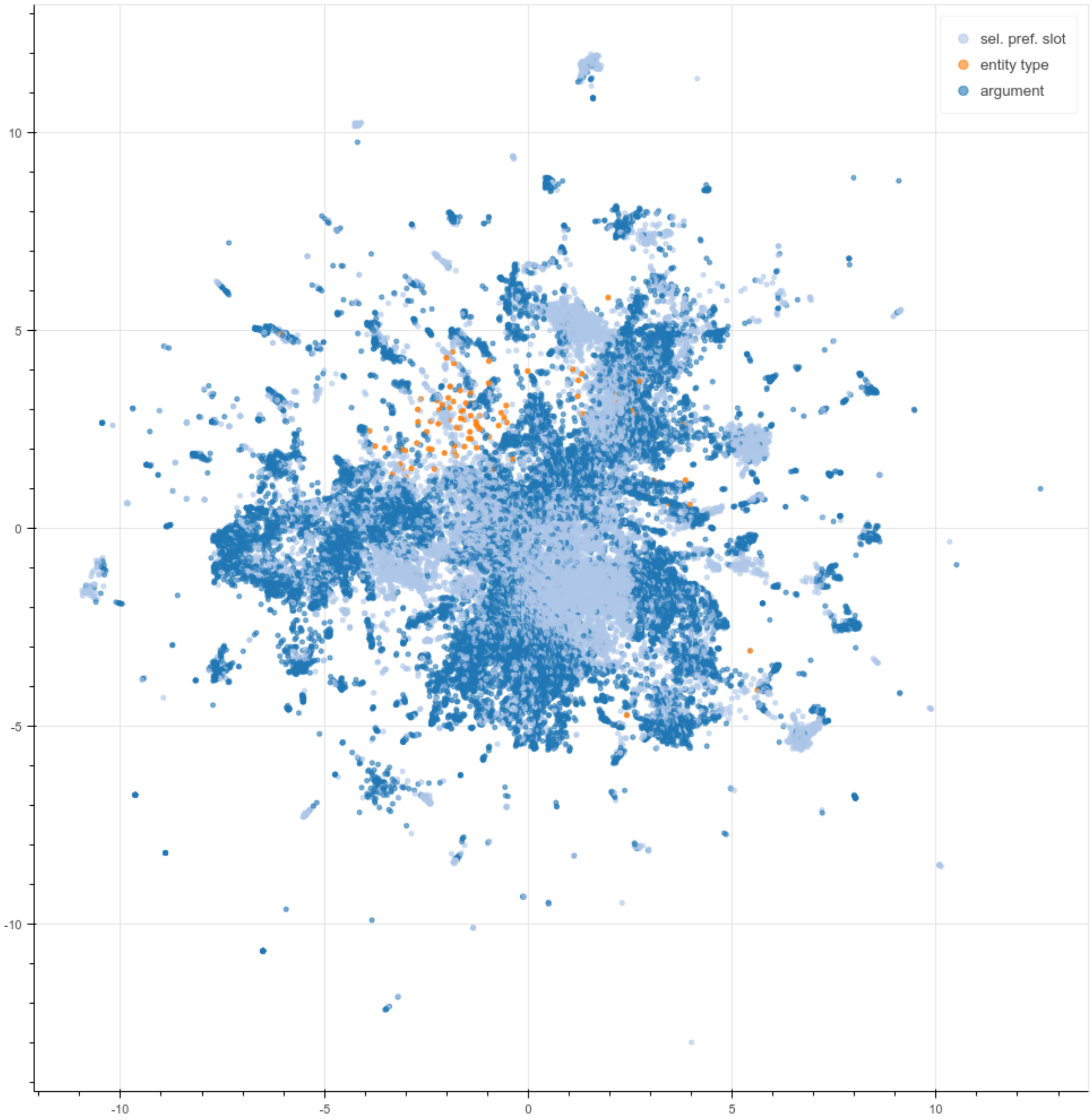


FIGURE 4.8: Visualization of selectional preference embeddings via a two-dimensional UMAP projection (McInnes et al., 2018).

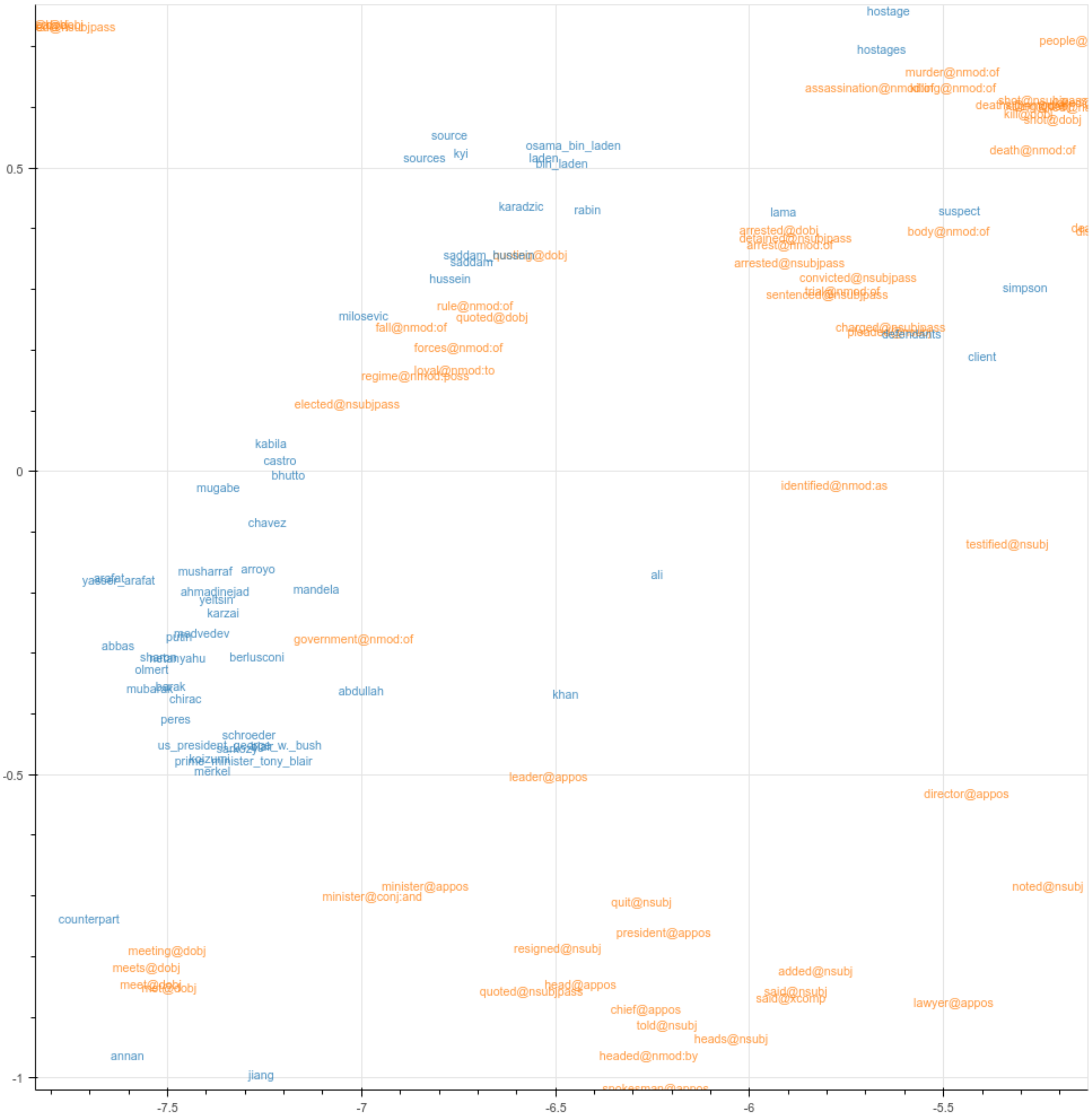


FIGURE 4.9: Zoomed view of selectional preference embeddings. Selectional preference slot embeddings are coloured blue and argument embeddings orange.

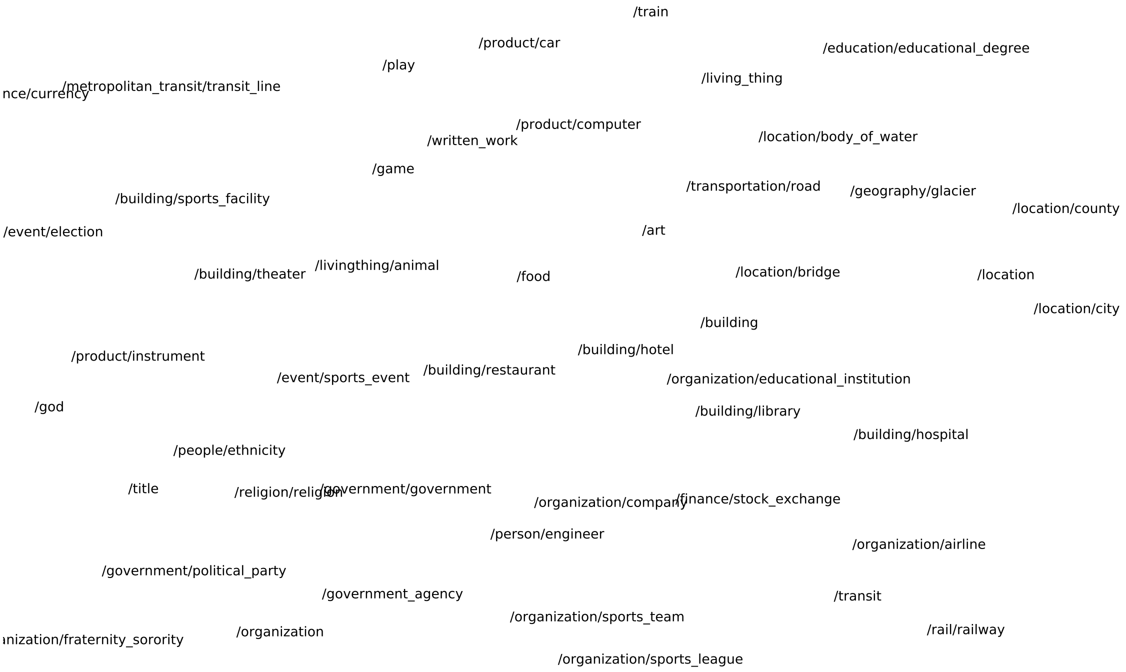


FIGURE 4.10: Zoomed view of selectional preference embeddings, filtered to show only embeddings of entity types.

the things politicians do, such as *meeting* their *counterpart* (lower left). In the top right, selectional preference slot embeddings such as *arrested@nsubjpass* ("was arrested"), *convicted@nsubjpass* and arguments like *defendants* form a cluster related to criminal justice. In the upper left region between these two clusters we find names of dictators (*saddam_hussein*) and war criminals (*milosevic*) and associated selectional preference slots such as *rule@nmod:of* (e.g. "the rule of Saddam Hussein") and *loyal@nmod:to* (e.g. "officials loyal to Slobodan Milosevic").

Turning to embeddings of entity types, in Figure 4.10 we see several regions in which similar entity types are clustered, for example different types of buildings (right of the center), locations on the right, and organizations in the lower left.

Concluding the qualitative evaluation, Table 4.2 on the facing page lists similar terms for queries from our examples. We see that the most similar terms are plausible in many cases. For example, selectional preference slots most similar to *sink@nsubj* include lexical and syntactic variants with similar meaning such as *sinking@nmod:of*, as in *the sinking of the RMS Lusitania*, or *sink@nsubj:xsubj*, i.e. the inherited subject in an open clausal complement, as in *the ship started to sink*. The most similar entity type is `/product/ship`, followed by `/event/natural_disaster` and `/finance/stock_exchange`. The latter also relates to `/finance/currency` and

Query	Most sim. sel. preference slots	Most sim. entity types	Most sim. phrases
sink@nsubj	sink@nsubj:xsubj	/product/ship	Sea_Diamond
	sink@nsubjpass	/event/natural_disaster	Prestige_oil_tanker
	sinking@nmod:of	/finance/stock_exchange	Samina
	slide@nsubj	/astral_body	Estonia_ferry
	capsizing@nmod:of	/person/religious_leader	k-159
	plunge@nsubj	/finance/currency	Navy_gunboat
	sink@nmod:along_with	/military	Dona_Paz
	sinking@nsubj	/geography/glacier	ferry_Estonia
	tumble@nsubj	/product/airplane	add-fisk-independent-nytsf
	slip@nsubj	/transit	Al-Salam_Boccaccio
ship	capsize@nmod:of	/product/ship	vessel
	some@nmod:aboard	/train	cargo_ship
	experience@nmod:aboard	/product/airplane	cruise_ship
	afternoon@nmod:aboard	/transit	boat
	pier@nmod:for	/product/spacecraft	freighter
	escort@nmod:including	/location/bridge	container_ship
	lift-off@nmod:of	/broadcast/tv_channel	cargo_vessel
	disassemble@nsubjpass:xsubj	/location	Navy_ship
	near-collision@nmod:with	/living_thing	warship
	Conger@compound	/chemistry	tanker
steer@dobj	guide@dobj	/broadcast/tv_channel	business_way
	steer@nsubjpass	/product/car	newr_nbkg_nwer_ndjn
	shepherd@dobj	/organization/sports_team	BahrainDinar
	steering@nmod:of	/product/ship	reynard-honda
	nudge@dobj	/product/spacecraft	zigzag_course
	pilot@dobj	/event/election	team_home
	propel@dobj	/medicine/medical_treatment	U.S._energy_policy
	maneuver@dobj	/building/theater	williams-bmw
	divert@dobj	/education/department	interest-rate_policy
	lurch@nsubj	/product/airplane	trimaran
/product/ship	Repulse@conj:and	/product/airplane	battleship_Bismarck
	destroyer@amod	/train	pt_boat
	capsize@nmod:of	/product/car	battleship
	experience@nmod:aboard	/park	USS_Nashville
	near-collision@nmod:with	/military	USS_Indianapolis
	line@cc	/event/natural_disaster	k-159
	brig@conj:and	/award	frigate
	-lrb-@nmod:on	/geography/island	warship
	Umberto@conj:and	/person/soldier	Oriskany
	rumour@xcomp	/location/body_of_water	sister_ship

TABLE 4.2: Most similar terms for the queries *sink@nsubj*, *ship*, *steer*, and */product/ship*.

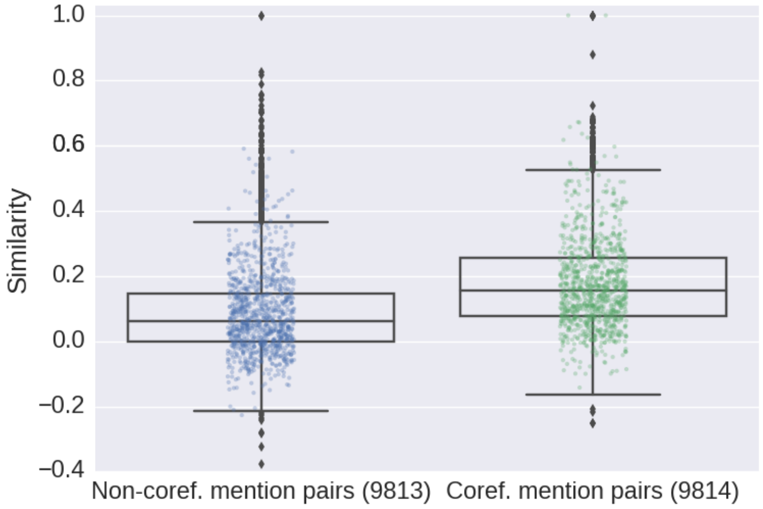


FIGURE 4.11: Selectional preference similarities of 10k coreferent and 10k non-coreferent mention pairs. The number of pairs covered by our embeddings is shown in parentheses. Lines and boxes represent quartiles, diamonds outliers, points subsamples with jitter.

falling prices expressed in phrases like *stocks plunged* or *the dollar tumbles*. We also observe noise (e.g. *add-fisk-independent-nytsf*), likely due to the fact that Wikilinks was created from crawled web documents.

4.5 Do Selectional Preferences Improve Coreference Resolution?

We now turn to the quantitative evaluation of our selectional preference embeddings in coreference resolution. For this, we perform experiments on the English section of the CoNLL'12 dataset (Pradhan et al., 2012), which consists of about 3500 documents from seven genres. To verify that our model of selectional preferences has the potential to improve coreference resolution, we compare selectional preference similarities of pairs of coreferent mentions, that is, two mentions which refer to the same entity, and the similarities of pairs of non-coreferent mentions, i.e. two mentions that refer to different entities. We hypothesize that selectional preference embeddings associated with coreferent mentions are similar, while those associated with non-coreferent mentions are less similar. Figure 4.11 shows the selectional preference similarity of 10.000 coreferent and 10.000 non-coreferent mention pairs sampled randomly from the CoNLL'12 training set. Coreferent mention pairs are indeed more similar than non-coreferent mention pairs, with a Matthews

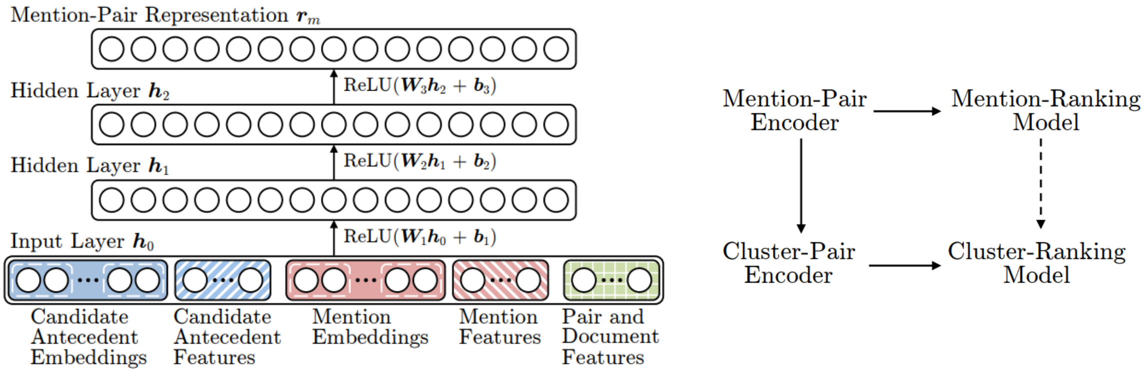

Mention-Pair Representation $\boldsymbol{r}_m$
Hidden Layer $\boldsymbol{h}_2$
ReLU($\boldsymbol{W}_3\boldsymbol{h}_2 + \boldsymbol{b}_3$)
Hidden Layer $\boldsymbol{h}_1$
ReLU($\boldsymbol{W}_2\boldsymbol{h}_1 + \boldsymbol{b}_2$)
Input Layer $\boldsymbol{h}_0$
ReLU($\boldsymbol{W}_1\boldsymbol{h}_0 + \boldsymbol{b}_1$)
Candidate Antecedent Embeddings
Candidate Antecedent Features
Mention Embeddings
Mention Features
Pair and Document Features
Mention-Pair Encoder
Mention-Ranking Model
Cluster-Pair Encoder
Cluster-Ranking Model

FIGURE 4.12: The `deep-coref` mention-pair encoder (left) and its position in the `deep-coref` architecture (right). Image source: Clark and Manning (2016b).

correlation coefficient of 0.30, indicating low to moderate correlation (Matthews, 1975). This suggests that selectional preferences alone constitute only a weak signal for distinguishing coreferent from non-coreferent mentions, but may be useful for coreference resolution in combination with other features.

We incorporate our selectional preferences model into `deep-coref`, the neural coreference resolution system by Clark and Manning (2016b). The system consists of a mention-pair encoder, mention-ranking model, cluster-pair encoder, and cluster-ranking model. Given a candidate antecedent mention and an anaphoric mention, the mention-pair encoder takes as input per-mention features such as the word embeddings associated with each mention, and pairwise features, such as the distance between the two mentions and whether they contain matching strings (Figure 4.12 left). The mention-pair representation produced by this encoder is then passed to the cluster-pair encoder, which builds representations of pairs of clusters of mentions. In parallel, the mention-pair representation is also provided to the mention-ranking model, which scores mention pairs according to whether they are likely to be coreferent or not. Based on these scores and the cluster-pair representations, the cluster ranking model incrementally merges pairs of mention clusters that have been deemed coreferent (Figure 4.12 right).

We measure the impact of our selectional preference model on coreference resolution performance in Table 4.3 on the following page. Evaluation metrics are *MUC* (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), *CEAF*$_e$ (X. Luo, 2005), whose average yields the CoNLL score, as well as *LEA* (Moosavi and Strube, 2016). We provide selectional preference information to the mention-pair encoder in two different forms: First, as per-mention features by concatenating selectional preference embeddings with `deep-coref`'s mention embeddings (+embeddings) and second,

	MUC	B^3	$CEAF_e$	CoNLL	LEA
			development		
baseline	74.10	63.95	59.73	65.93	60.16
+embedding	74.38	64.42	60.45	66.42	60.65
+similarity	74.36	64.54	60.21	66.37	60.77
			test		
baseline	74.72	63.26	58.82	65.60	59.59
+embedding	74.53	63.41	59.03	65.66	59.69
+similarity	74.85	63.64	59.21	65.90	59.98

TABLE 4.3: Comparison of two ways of integrating our selectional preference model into `deep-coref`. See text for details. Source: Heinzerling, Moosavi, et al. (2017).

as pairwise features by computing selectional preference similarities of mention pairs (+similarity). Both methods lead to small improvements across all metrics when evaluating on the CoNLL'12 development set. The improvement transfers to the test set when using pairwise selectional preference similarities, but almost vanishes when providing selectional preference information in form of embeddings. A possible explanation is that the higher number of parameters when using embeddings results in overfitting to the development set, while adding only a few similarity scores does not lead to a large in increase in model parameters and hence does not increase the risk of overfitting.

The mention embeddings used as input by the `deep-coref` mention-pair encoder include the word embedding of the mention's dependency governor. That is, given the mention *Titanic* in

(92) The Titanic sank.

both the embedding of the word *Titanic* and the word *sank* will be used to produce the mention-pair representation. To compare the impact of the governor embedding and the selectional preference slot embedding provided by our model – in this case the embedding of *sank@nsubj* – we ablate the dependency governor embedding feature (-gov) in Table 4.4 on the next page. The governor feature has a negligible impact on coreference resolution quality, while adding selectional preference features (+SP) yields improvements of 0.30 CoNLL F_1 points and 0.39 LEA F_1 points. This improvement is comparable to the improvement achieved by a later version of `deep-coref` (Reinforce) (Clark and Manning, 2016a).[4]

[4] We did not perform ablation experiments with this later version since the greatly increased training time renders such experiments impractical.

	MUC			B^3			CEAF$_e$			CoNLL	LEA		
	R	P	F$_1$	R	P	F$_1$	R	P	F$_1$	Avg. F$_1$	R	P	F$_1$
baseline	70.09	80.01	74.72	57.64	70.09	63.26	54.47	63.92	58.82	65.60	54.02	66.45	59.59
−gov	70.10	79.96	74.71	57.51	70.31	63.27	54.41	64.08	58.85	65.61	53.93	66.76	59.66
+SP	70.85	79.31	74.85	58.93	69.16	63.64	55.25	63.78	59.21	65.90	55.29	65.53	59.98
Reinforce	70.98	78.81	74.69	58.97	69.05	63.61	55.66	63.28	59.23	65.84	55.31	65.32	59.90

TABLE 4.4: Final results on the CoNLL'12 test set. See text for details. Source: Heinzerling, Moosavi, et al. (2017).

Error type	Mention type		
	Proper	Common	Pronoun
Recall	-28	-29	-53
Precision	+18	+74	+61

TABLE 4.5: Changes in recall and precision errors on the CoNLL'12 test set for three mention types. Source: Heinzerling, Moosavi, et al. (2017).

We use the error analysis tool by Martschat and Strube (2014) to analyze the changes caused by selectional preference features.[5] The error analysis tool distinguishes between recall errors, that is, failure to recognize a pair of coreferent mentions as such, and precision errors, that is, deeming two mentions coreferent even though they are not. As Table 4.5 shows, the number of recall errors decreases. That is, selectional preference features enable the system to find more correct links between coreferent mentions. However, the number of precision errors increases. In other words, selectional preference features lead to more spurious links between non-coreferent mentions. For proper noun mentions, the increase in precision errors is outweighed by the reduction in recall errors, suggesting that the entity type information contained in our model of selectional preferences is beneficial. Examples

[5] The toolkit is available at `https://github.com/smartschat/cort`.

Text	Relevant selectional preference pair
does [**that**]$_{ante}$ really impact the case ... [it]$_{ana}$ just shows	(impact@nsubj, shows@nsubj)
[it]$_{ante}$ will ask a U.S. bankruptcy court to allow [it]$_{ana}$	(ask@nsubj, allow@dobj)
[The country **coroner**]$_{ante}$ says [he]$_{ana}$ urged	(says@nsubj, urged@nsubj)
[a **strain** that has n't even presented [itself]$_{ana}$]$_{ante}$	(presented@nsubj, presented@dobj)

TABLE 4.6: Examples of recall improvements over the baseline system due to the inclusion of selectional preference features. Source: Heinzerling, Moosavi, et al. (2017).

of these recall improvements are shown in Table 4.6 Inspection of system output reveals that many of precision errors, i.e. false positive spurious links, are established between mentions with identical governors and dependency relations.

4.5.1 Limitations

Our model of selectional preferences has several limitations:

- Like all distributed representations derived from text that has not been annotated with word senses, our embeddings conflate multiple senses of both predicates and arguments. We saw an example of this in Figure 4.7 on page 86 where typical subjects of the predicate *sank* include both ships and athletes.
- Due to the different distributions of selectional preference slot and argument embeddings on the one hand and entity type embeddings on the other hand, absolute similarity values differ when comparing embeddings in the same category, e.g. entity types to each other, to when comparing embeddings in different categories, e.g. entity types and selectional preference slots. This makes integrating such similarities as features in a coreference resolution system more complex.
- We consider selectional preferences only in terms of syntactic dependency relations, since dependency parsers are fast and reliable, while fast and reliable automatic annotation for semantic representations is currently out of reach. It is possible that integrating "more semantic" representations into coreference resolution, as has been done in previous work, would lead to improvements. For example, Erk (2007) and Rahman and Ng (2011) integrated Frame-semantic relations, Ponzetto and Strube (2006a) semantic roles, and (Ponzetto and Strube, 2006b) word knowledge into coreference resolution systems. However, to the best of our knowledge, no such integration has been successfully attempted with current coreference resolvers.

4.6 Summary

In this chapter, we introduced a distributional model of fine-grained selectional preferences with high coverage. While a qualitative evaluation showed promising results, the impact on a neural coreference resolution system was small. This result mirrors results in prior work on integrating semantics into coreference resolution systems. World knowledge relations lead to higher recall, but to a loss in precision due to spurious connections between non-coreferent mentions.

Chapter 5

Multilingual Entity Typing with Subword Units

5.1 Introduction

When automatically analyzing entities mentioned in a text, entity mentions can only be linked to a knowledge base if the knowledge base contains a corresponding entry. However, knowledge bases commonly used for entity linking, such as Wikipedia, DBpedia (Auer et al., 2007), YAGO (Hoffart, Suchanek, et al., 2011), or Freebase (Bollacker et al., 2008), do not contain entries for all entities in the world. One reason for this incompleteness is the fact that the world is not static: New entities come into existence every day, but not every emerging entity (Nakashole et al., 2013; Hoffart, Altun, et al., 2014) is immediately added into all knowledge bases. A second reason is that complete coverage of all entities is not a design goal of the aforementioned knowledge bases. For example, Wikipedia defines criteria according to which only entities and topics with a certain degree of notability are suitable for inclusion (Wikipedia contributors, 2018).[1]

Since the proportion of mentions of entities not contained in the knowledge base varies with the text being analyzed and the knowledge base being used, it is difficult to quantify the problem of unknown entities in general.[2] Statistics on common benchmark datasets indicate that this ratio can range from about 20 percent to over 50 percent of entity mentions (Table 5.1 on the next page). In other words, it is not uncommon that more than half of the entities mentioned in a text do not have

[1] See `https://en.wikipedia.org/wiki/Wikipedia:Notability` (Accessed: 2018-09-21).

[2] In this chapter, we use the term *unknown entity* to refer to both entities that are not mentioned during training and to refer to entities that are not contained in the knowledge base. Where the distinction matters, we refer to the latter as *out-of-knowledge-base* entities. We do not use the synonymous term *Not-In-Lexicon (NIL) entities* in this chapter, since it is not commonly used outside the series of TAC workshops (McNamee and Dang, 2009).

Dataset	#Mentions	in-KB	NIL	NIL ratio (%)
TAC2010	1500	1074	426	28.4
TAC2011	2162	1001	1161	53.7
TAC2012	2008	1141	867	43.2
TAC2013	6462	3601	2861	44.3
TAC2014	5966	3342	2624	44.0
TAC2015	30834	23985	6849	22.2

TABLE 5.1: Ratio of out-of-knowledge-base entity mentions in common entity linking datasets.

a corresponding entry in the knowledge base. Hence, it is important for entity analyzers to be able to deal with unknown entities.

One way of analyzing unknown entities is classification into semantic categories such as `Person`, `Organization`, or `Location`. These categories are called *entity types* and the task of classifying mentions of entities according to the type of the entity they refer to is called *entity typing*.[3] In entity typing, we assume an entity mention has already been recognized and now want to assign to it one or more entity types. For example, consider the following sentence:

(93) Thomas Austin introduced rabbits to Australia in 1859, for sporting hunters.[4]

Here, the entity mention Thomas Austin refers to an entity with the entity type `Person` and the entity mention Australia to an entity with the entity type `Location`. In examples given in this chapter, we denote entity types by typesetting them directly above their corresponding entity mention:

```
      Person                        Location
(94) Thomas Austin introduced rabbits to Australia in 1859, for sporting hunters.
```

5.1.1 Type Inventory

Entity types are taken from a pre-defined inventory of semantic categories, called *type inventory*. Type inventories differ in:

[3] In the literature, this task is variously also called *semantic typing* (Durrett and Klein, 2014), *entity type tagging* (Gillick et al., 2014), *entity type classification* (Yogatama et al., 2015), or *entity mention typing* (Yosef et al., 2012). Borrowing a particular sense of the term *entity* in coreference resolution, Yaghoobzadeh and Schütze (2015) make a precise distinction between typing of individual entity mentions, called *mention typing* and assigning a type to a set of coreferent entity mentions, which they call *entity typing*. In calling our task *entity typing*, we follow recent and common, but somewhat imprecise usage (Del Corro et al., 2015; Ren, He, Qu, Huang, et al., 2016; Ren, He, Qu, Voss, et al., 2016; Y. Ma et al., 2016; Rabinovich and Klein, 2017; Choi et al., 2018).

[4] Sentence taken from:
`https://csiropedia.csiro.au/myxomatosis-to-control-rabbits/` (Accessed: 2018-09-25).

- size, that is, whether they contain few or many entity types;
- granularity, that is, whether they make fine-grained distinctions between different types or consist of coarse entity types; and
- domain, for example general news, finance, biology, or medicine.

The choice of type inventory is determined by the needs of the end user, for example the user of a semantic search engine, or by the requirements of downstream applications that further process a text. The type inventory for a user searching for companies mentioned in a text might consist of only one `Organization` type (Rau, 1991). A type inventory for military use would include entity types such as `Facility`, `Vehicle`, and `Weapon` (Doddington et al., 2004), and a type inventory for bio-medical applications might contain entity types such as `Protein`, `DNA`, or `Cell Type` (J.-D. Kim et al., 2003). For generic use, a series of shared tasks held at CoNLL workshops (Sang, 2002; Tjong Kim Sang and De Meulder, 2003) established a type inventory consisting of three coarse types, namely `Person`, `Location`, and `Organization`, as well as a `Miscellaneous` type for named entities that do not match any of the other types. In order to provide more detailed information to downstream tasks like relation extraction, which may require finer distinctions, these coarse types were subsequently refined into *fine-grained* type inventories. Fine-grained type inventories contain about 100 fine-grained entity types and subtypes, organized in shallow hierarchies with two to three levels (X. Ling and Weld, 2012; Yogatama et al., 2015).

Figure 5.1 on the following page shows the Google Fine Types inventory (Gillick et al., 2014) we use in this work, which consists of 89 entity types organized in a three-level hierarchy. Taking entity types from this inventory and representing hierarchy levels with a slash ("/"), our example becomes:

`/person` `/location/country`
(95) Thomas Austin introduced rabbits to Australia in 1859, for sporting hunters.

That is, in the Google Fine Types inventory the entity mention Australia has the type coarse type `/location` and the fine type `/location/country`. Lacking a well-matching fine-grained type for Thomas Austin, a 19th-century settler, we abstain from assigning a fine-grained type and resort to using only the top-level `/person` type.

We are now ready to define the entity typing task: Given an entity mention m the task is to label it with a set of one or more entity types $\hat{t} \subseteq T$, where T is a type

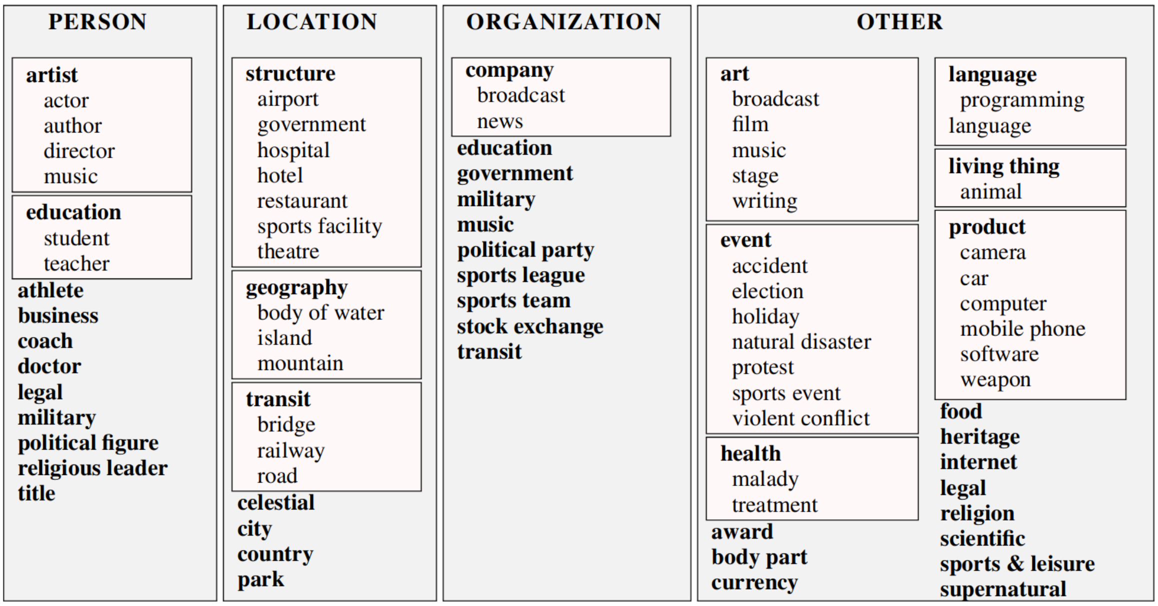


FIGURE 5.1: The Google Fine Types inventory proposed by Gillick et al. (2014). Nested boxes represent hierarchy levels. `PERSON`, `LOCATION`, `ORGANIZATION`, and `OTHER` form the top level. Bold font denotes the second level, e.g., **`artist`** is a subtype of `PERSON`, and light font indicates the third level, e.g. `actor` is a subtype of **`artist`**. Image source: Gillick et al. (2014).

inventory. In the machine learning literature, this kind of task is known as multi-label classification (Tsoumakas and Katakis, 2007) and is typically approached in a supervised learning setting.

5.1.2 Feature-based Entity Typing

A typical supervised entity typing system extracts features for the given mention m and predicts entity types $\hat{t}$ based on these features. During training, the system then receives the true entity types t and uses this supervision signal to adjust itself with the aim of making better predictions in the future.

Features are designed to automatically capture as much relevant information about the given entity mention as possible. Entity typing features fall into three main categories:

1. **Mention surface** features are based on the observation that named entity mentions exhibit certain regularities. For example, *Mary* often refers to a person, Western person names often consist of two capitalized words, place

names in England have suffixes like *-ford* and *-shire*, and the suffix *-osis* is common in the names of medical conditions and bio-chemical processes. To exploit theses regularities, systems extract surface features from the mention such as the mention text itself, its first few characters, its last few characters, whether it is capitalized, or its length.

2. **Mention context** features exploit coherence between entity mentions and their surrounding context. For example, consider the following sentence:

   (96) After a fierce campaign, X won the election by only one vote.

   Here, phrases from the political domain, namely "campaign", "win the election", and "vote", appear in the context of the entity mention X and suggest that X is a politician.

3. **World knowledge** features make use of resources beyond the text itself. A simple way of incorporation world knowledge is the use name lists and gazetteers which allow checking, for example, whether a mention contains a common first name (e.g. *Mary*), place name (*London*), company type (*Inc.*, *Ltd.*), or title (*Dr.*, *Esq.*).

Having investigated the use of world knowledge in chapter 3 and the use of context in chapter 4, in this chapter we turn our focus on mention surface features.

The mention surface features described above are intuitive and interpretable, but suffer from the drawback of symbolic represenation: Knowing that *Stephen* is a person does not help deciding whether *Steven* is a person as well, since these names are taken as discrete, opaque objects without inherent similarities. While it is possible to define similarity measures such as string edit distance, these measures are not robust. For example, *Stephen* and *Steven* have a small string edit distance of 2 and would be judged correctly as similar, but according to this criterion, the number *seven* or the city name *Steuben*, which also have a string edit distance of 2 to *Steven* would also be classified as a person.

While early coarse-grained and fine-grained entity typing systems (McCallum and Li, 2003; X. Ling and Weld, 2012) employed manually-defined surface features like the ones described above, more recent neural approaches overcome some of the limits of symbolic representaion via a combination of neural networks and automatically-learned distributional word representations.

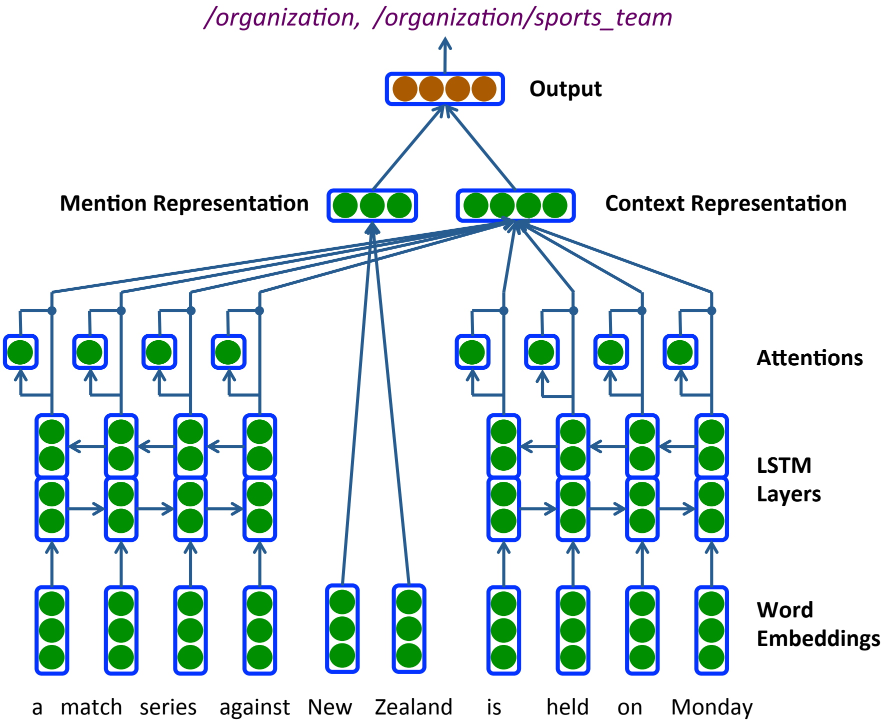


FIGURE 5.2: A typical neural entity typing architecture. See text for details. Image source: Shimaoka et al. (2017).

5.1.3 Neural Entity Typing

Due to the abovementioned disadvantages of manually-defined, sparse, symbolic features, and drawn by the promise of achieving better performance without handcrafted features, a recent line of work has applied deep learning techniques (LeCun, Bengio, et al., 2015; Schmidhuber, 2015; Goodfellow et al., 2016) to the entity typing task (Yogatama et al., 2015; X. Ma and Hovy, 2016; Lample et al., 2016; Chiu and Nichols, 2016; Shimaoka et al., 2017).

A typical neural entity typing architecture is depicted in Figure 5.2 . The architecture consists of several layers. The lowest layer maps the entity mention (here, “New Zealand”) and its context into a sequence of word embeddings. LSTM layers (Hochreiter and Schmidhuber, 1997) encode the mention, as well as its left (“a match series against”) and right context (“is held on Monday”) into fixed-size vector representations. When using an attention mechanism (Bahdanau et al., 2014), the left and right contexts are encoded as a weighted average of the LSTM states corresponding to each context word. The vectors representing the entity mention and its left and right contexts are then concatenated and projected into the output layer, which is a vector having one entry for each entity type in the type inventory.

In its purest configuration, the only features used in this kind of architecture are pretrained word embeddings, for example GloVe (Pennington et al., 2014). Consequently, the quality of the word embeddings used has a direct effect on the quality of entity type predictions. This reliance on pretrained word embeddings leads to two problems which we discuss in the following: the unknown word problem and the rare word problem.

5.1.4 The Unknown Word Problem in Entity Typing

All the neural entity typing systems cited above rely on pre-trained word embeddings as their main, or in some cases, only input representation. While these word embeddings are trained on large corpora, their vocabulary is of a fixed size and does not contain all words encountered by an entity typing sytem.

As shown in Table 5.2 on the following page, such *unknown* or *out-of-vocabulary* (OOV) words constitute a considerable fraction of all entity mentions. In the best case, where embeddings trained on Wikipedia and a large number of news articles from the GigaWord corpus are used for entity typing on the CoNLL'03 dataset, which is also from the news domain, about ten percent of words occurring in entity mentions are unknown, that is, not contained in the vocabulary of the pre-trained embeddings used.

In a *noisy* dataset containing non-standard spelling, inconsistent capitalization, and generally exhibiting more lexical variety than edited and proofread news articles, there are many more unknown words. For example, 67 percent of the words found in entity mentions in the Wikilinks corpus (Singh et al., 2012), which is derived from a web crawl, are not contained in word embeddings trained on Wikipedia and GigaWord (Parker et al., 2011). If the embeddings are instead trained on a much larger corpus from a matching domain, such as Common Crawl,[5] the vocabulary size increases five-fold. However, this vocabulary increase does not lead to a proportional decrease in unknown words, with OOV ratios falling only to about 50 percent.

In case of a domain mismatch, the number of unknown words can increase dramatically. For example, even though the vocabulary of embeddings trained on a corpus derived from Twitter is three times larger than the vocabulary obtained from Wikipedia and GigaWord, it contains considerably fewer words occurring in entity mentions, resulting in OOV ratios of 81 percent for web documents from Wikilinks and 27 percent for news articles in CoNLL'03.

[5] http://commoncrawl.org/.

Corpus	tokens	vocabulary size	Dataset	OOV entity mention words	OOV ratio
Wikipedia and GigaWord	6B	400k	Wikilinks	523k	67%
			CoNLL'03	784	10%
Twitter	27B	1.2m	Wikilinks	630k	81%
			CoNLL'03	2088	27%
Common Crawl (lowercased)	42B	1.9m	Wikilinks	387k	50%
			CoNLL'03	819	11%
Common Crawl (cased)	840B	2.2m	Wikilinks	460k	49%
			CoNLL'03	976	12%

TABLE 5.2: Ratio of out-of-vocabulary (OOV) words occurring in entity mentions from two datasets, CoNLL'03 and Wikilinks. From the 34k entity mentions in the CoNLL'03 training set, we extract all unique tokens,[a] obtaining 7.7k lowercased and 8.3k cased tokens. Similarly, we extract all unique tokens from the 22 million entity mentions in Wikilinks, obtaining 776k unique lowercased tokens and 994k cased tokens. Then, we count how many of these unique tokens are not present in four different versions of `GloVe`, which have been trained on increasingly large corpora from various domains. Finally, the OOV ratio is computed as the ratio of OOV unique entity mention tokens to all unique entity mention tokens.

[a] Corpus linguistics distinguishes between *tokens* and *types*. The latter would be the appropriate term here, but to avoid confusion with *entity types*, we say *unique tokens* instead.

These statistics show that unknown words are a considerable problem when processing named entities. What is more, even known words may cause difficulties if they are so rare that it is difficult to learn good representations for them when training word embeddings.

5.1.5 The Rare Word Problem in Entity Typing

In the previous section we saw that many words occurring in entity mentions are unknown, that is, not contained in the pretrained embeddings used by entity typing systems. A second problem is the fact that entity mentions also contain many rare words, as shown in Figure 5.3 on the next page. Like words in general language, words in entity mentions follow Zipf's law (Zipf, 1946). That is, their distribution is characterized by few very frequent words and a *long tail* of many infrequent words, Figure 5.4 on page 109. The high prevalence of rare words in entity mentions makes using word embeddings difficult. Current word embedding methods (Mikolov, Sutskever, et al., 2013; Pennington et al., 2014; Bojanowski et al., 2017) train embeddings on large corpora. However, even with a large corpus, word embedding quality is high for frequent words, but degrades for less frequent words (Bullinaria and J. P. Levy, 2007; Luong, Socher, et al., 2013). This degradation is due to the fact that during training, the embedding algorithm encounters much fewer contexts for less frequent words. Since word embeddings are optimized to be similar to the embeddings of their context words, less frequent words are more susceptible to noise: Any unrelated words that happen to be in a word's context have a stronger influence on the final word representation than is the case for more frequent words. We now introduce subword embeddings, a promising solution to the problem of rare and unknown words.

5.2 Subwords as a solution to the unknown and rare word problems

As we have seen in the previous two sections, learning good representations of rare words or words not seen during training at all is a difficult challenge in neural entity typing. As a makeshift solution, some systems[6] have replaced unknown words with a generic *UNK* token, which is then treated like any other word. Recently, based on the assumption that a word's meaning can be reconstructed from its parts,

[6] For example, the system by Shimaoka et al. (2017).

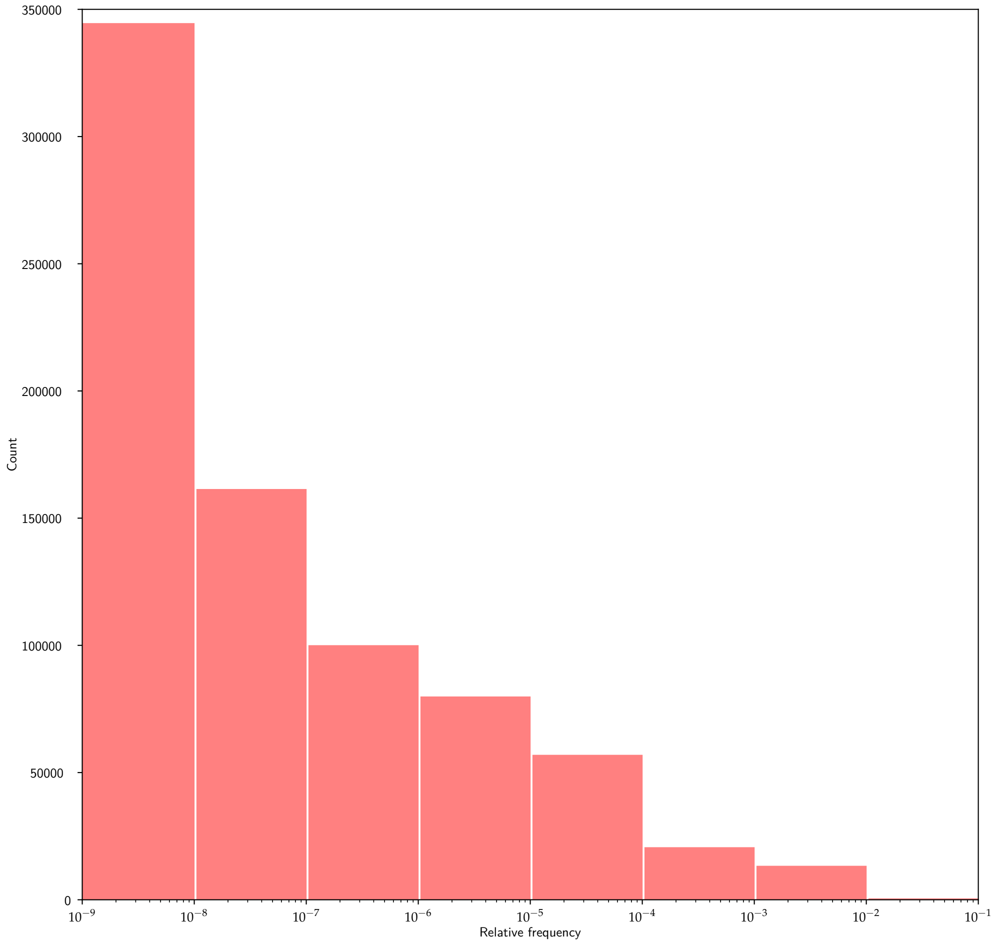


FIGURE 5.3: Relative frequencies of words occurring in entity mentions in the Wikilinks corpus. Of the 776k unique lowercased tokens occurring in entity mentions, about 340k have a relative frequency of 10^{-8} or less. In other words, almost of half of all words in entity mentions appear only once or fewer times per 100 million tokens in a background corpus. Frequencies obtained from `wordfreq` (Speer et al., 2017). For visual clarity, relative frequencies smaller than 10^{-8} are clipped to 10^{-8}.

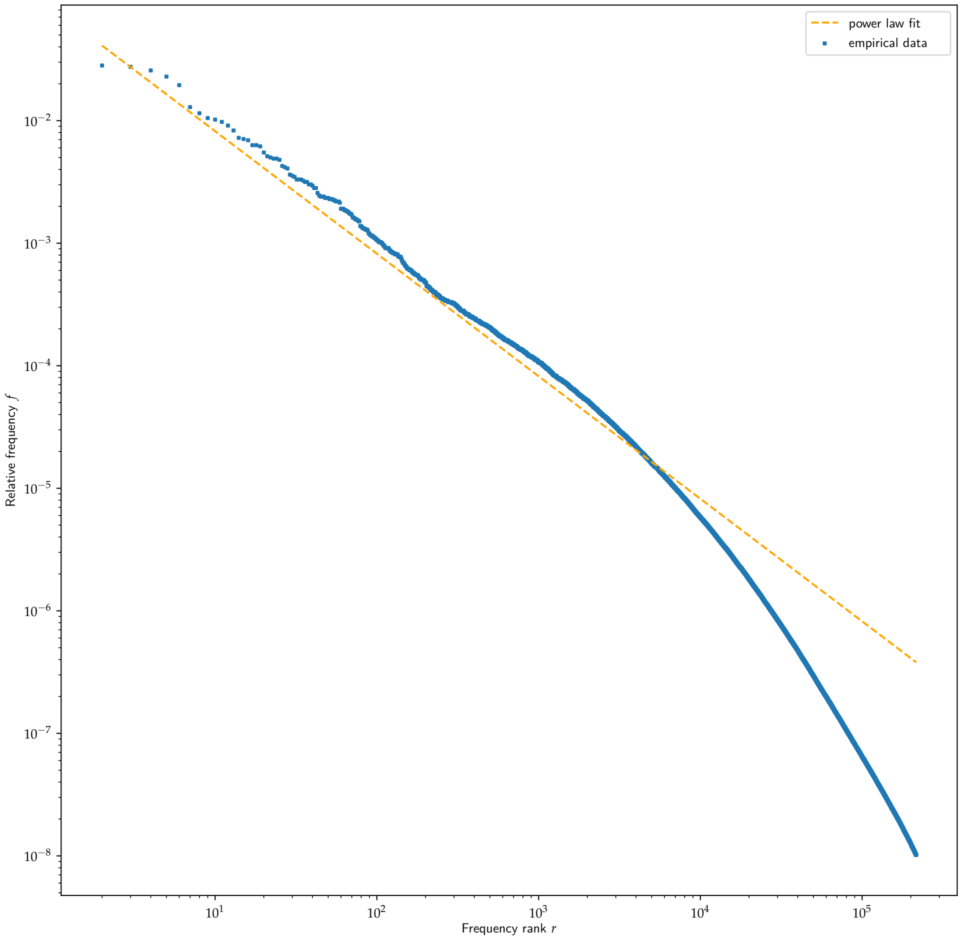


FIGURE 5.4: Zipfian distribution of words occurring in entity mentions in the Wikilinks corpus. Empirical word frequencies f given frequency rank r are approximately distributed according to a power law $f(r) \propto r^{\alpha}$, with $\alpha = -1$. Frequencies obtained from `wordfreq` (Speer et al., 2017).

several subword-based methods have been proposed as a better solution (Chiu and Nichols, 2016; X. Ma and Hovy, 2016; Lample et al., 2016, i.a.)

Subword-based methods divide a word into smaller *subword units*, e.g. morphemes, single characters, or longer character n-grams such as bigrams and trigrams. The meaning of a word is then modeled by first learning representations of its constituting subword units, and then learning a composition function which composes subword representations into a word representation. We now describe different choices of subword units in more detail.

5.2.1 Subword units

In this section, we describe four different kinds of subword units, namely morphemes, character n-grams, `FastText`, and byte pairs, using the word *myxomatosis*, a tumorous disease in rabbits, as a running example.

Morphemes

Morphemes are the smallest meaning-carrying units of language. By performing a morphological analysis, we can split the word *myxomatosis* into three morphemes:

1. *myxo-*: *mucus*, *slime*, from Greek *mýxa*[7]
2. *-omat-*: *tumor*, from Greek *-ōmat-*, *-ōma*[8]
3. *-osis*: *abnormal or diseased condition*, from from Greek *-ōsis*[9]

Among the subword units we discuss in this chapter, morphemes are linguistically most sound: By definition, it does not make sense to split words into smaller units such as characters, since any smaller units do not carry meaning. Composition of learned morpheme representations was found to produce improved word representations in intrinsic word similarity evaluations (Lazaridou et al., 2013; Luong, Socher, et al., 2013), but has the drawback of requiring a morphological analyzer, which may not be available for the language, genre, or domain of interest.

Characters

Splitting a word into its characters, e.g. *myxomatosis* into the character sequence *m, y, x, o, m, a, t, o, s, i, s*, is arguably the simplest way of obtaining subword units. An

[7] Source: Merriam-Webster online dictionary.
`https://www.merriam-webster.com/dictionary/myxo-` (Accessed: 2018-10-04).

[8] Ibid. `https://www.merriam-webster.com/dictionary/-oma-` (Accessed: 2018-10-04).

[9] Ibid. `https://www.merriam-webster.com/dictionary/-osis` (Accessed: 2018-10-04).

additional advantage is the fact that the number of characters in a corpus is much smaller than the number of other subword units like morphemes, making it easier to learn character representations. The main difficulty of character-based subword approaches lies in the composition function. Since character sequences are longer than sequences of larger subword units, simple composition functions do not yield good word representations. For example, while simple addition of morpheme representations works well (Lazaridou et al., 2013), character-based approaches require more complex composition functions such as recurrent neural networks (Luong and Manning, 2016), which, unlike simple addition, are able to retain order information.

Character n-grams

Character n-grams are sequences of characters of length n. Like characters, which are character-ngrams with $n = 1$, they are simple to obtain: For $n = 2$, *myxomatosis* becomes *my, yx, xo, om, ma, at, to, os, si, is*, and for $n = 3$, *myx, yxo, xom, oma, mat, ato, tos, osi, sis*. As n increases words are split into fewer subwords, making composition easier. Among the character n-grams we can identify the trigram *oma* as being quite similar to one of the morphemes found in the morpholical analysis conducted above, and it is conceivable that a system could learn a good representation of this trigram. However, we also observe that many character n-grams do not appear meaningful. Character-trigrams have been shown to work well for language modeling (Vania and Lopez, 2017), but character unigrams are the most common subword unit in neural named entity recognition (Chiu and Nichols, 2016). A drawback of character n-grams is that a fixed n makes the unrealistic assumption that the meaning-carrying units of language all have the same length in characters. In the next sections we introduce two methods that address this shortcoming.

FastText

As it is not clear what the best fixed length of character-based subword units should be, a solution is to use character n-grams of many different lengths. This is the approach taken by `FastText` (Bojanowski et al., 2017), which represents a word w as the sum of its constituting character n-grams g, where, in turn, each character-ngram is represented by a learned vector $\vec{z}_g$:

$$\vec{w} = \sum_{g \in G_w} \vec{z}_g$$

Step	Sequence	Pair counts	Merge operation
1	A B A B C A B C D	AB: 3, BC: 2, BA: 1, CA: 1, CD: 1	A B → X
2	X X C X C D	XC: 2, XX: 1, CD: 1	X C → Y
3	X Y Y D	XY: 1, YY: 1, YD: 1	`Terminate`

TABLE 5.3: BPE compression example.

where G_w is the set of all constituting character n-grams. In practice, all 3-, 4-, 5- and 6-grams are used. For example, *myxomatosis* is represented as:

$$\begin{aligned}\vec{myxomatosis} =& \vec{myx} + \vec{yxo} + \vec{xom} + \vec{oma} + \vec{mat} + \vec{ato} + \vec{tos} + \vec{osi} + \vec{sis}\\ &+ \vec{myxo} + \vec{yxom} + \vec{xoma} + \vec{omat} + \vec{mato} + \vec{atos} + \vec{tosi} + \vec{osis}\\ &+ \vec{myxom} + \vec{yxoma} + \vec{xomat} + \vec{omato} + \vec{matos} + \vec{atosi} + \vec{tosis}\\ &+ \vec{myxoma} + \vec{yxomat} + \vec{xomato} + \vec{omatos} + \vec{matosi} + \vec{atosis}\end{aligned}$$

A drawback of this method is its brute-force nature and the redundancy stemming from large overlap in long character n-grams. Furthermore, the large vocabulary containing not only words, but also all character-n-grams results in unwieldy embedding file sizes of several gigabytes for a single language.

Byte Pair Encoding (BPE)

Byte Pair Encoding (BPE) (Gage, 1994) is a lossless compression algorithm based on substitution coding, that is, it compresses a sequence of symbols such as bytes or characters by replacing frequent long patterns of symbols with shorter symbol patterns. Specifically, BPE counts pairs of adjacent bytes and iteratively replaces the most frequent byte pair with a byte that does not occur in the sequence Table 5.3 .

When applied to text, and with a slight modification to the original algorithm, BPE becomes an unsupervised method for segmenting words into subwords (Sennrich et al., 2016). Instead of replacing symbol pairs with a new, shorter symbol, as is done in the original compression algorithm, BPE for text *merges* the most frequent adjacent pair of symbols. For example, the first step in Table 5.4 on the facing page merges the symbol pair *A B* into a new symbol *AB*.

When iteratively merging symbol pairs, BPE maintains an ordered list of all merges it performed. This ordered list is called a *BPE model* and the set of merged symbols is called the *BPE vocabulary*.

Table 5.5 on the next page shows part of a BPE model trained on English Wikipedia.

Step	Sequence	Pair counts	Merge operation
1	A B A B C A B C D	AB: 3, BC: 2, BA: 1, CA: 1, CD: 1	A B → AB
2	AB AB C AB C D	ABC: 2, ABAB: 1, CD: 1	AB C → ABC
3	AB ABC ABC D	ABABC: 1, ABCABC: 1, ABCD: 1	`Terminate`

TABLE 5.4: BPE segmentation example.

Step	Merge operation
1	_ t → _t
2	_ a → _a
3	h e → he
4	i n → in
5	_t he → _the
6	0 0 → 00
7	e r → er
8	_ s → _s
9	o n → on
10	_ c → _c
11	r e → re
12	_ o → _o
13	_ w → _w
14	i s → is
15	a n → an
16	_ in → _in
...	
702	o ugh → ough
703	_ser ies → _series
704	in t → int
705	a i → ai
706	st it → stit
707	er y → ery
708	is ter → ister
...	
6039	ig o → igo
6040	os is → osis
6041	_jo se → _jose
...	
96513	omat osis → omatosis
...	

TABLE 5.5: BPE merge operations learned on the English edition of Wikipedia. The underscore represents a whitespace character. See Appendix A on page 137 and Appendix B on page 139 for complete lists of 1000 merge operations learned on English and German Wikipedia.

Merge ops	Byte-pair encoded text
5000	豊 田 駅 (と よ だ え き) は 、 東京都 日 野 市 豊 田 四 丁目 にある
10000	豊 田 駅 (と よ だ えき) は 、 東京都 日 野市 豊 田 四 丁目にある
25000	豊 田駅 (とよ だ えき) は 、 東京都 日 野市 豊田 四 丁目にある
50000	豊 田駅 (とよ だ えき) は 、 東京都 日 野市 豊田 四丁目にある
Tokenized	豊田 駅 （ と よ だ え き ） は 、 東京 都 日野 市 豊田 四 丁目 に ある
10000	豐 田 站 是 東 日本 旅 客 鐵 道 (JR 東 日本) 中央 本 線 的 鐵路 車站
25000	豐田 站是 東日本旅客鐵道 (JR 東日本) 中央 本 線的鐵路車站
50000	豐田 站是 東日本旅客鐵道 (JR 東日本) 中央 本線的鐵路車站
Tokenized	豐田站 是 東日本 旅客 鐵道 （ JR 東日本 ） 中央本線 的 鐵路車站
1000	to y od a _station is _a _r ail way _station _on _the _ch ū ō _main _l ine
3000	to y od a _station _is _a _railway _station _on _the _ch ū ō _main _line
10000	toy oda _station _is _a _railway _station _on _the _ch ū ō _main _line
50000	toy oda _station _is _a _railway _station _on _the _chū ō _main _line
100000	toy oda _station _is _a _railway _station _on _the _chūō _main _line
Tokenized	toyoda station is a railway station on the chūō main line

TABLE 5.6: Effect of the number of BPE merge operations on the beginning of the Japanese (top), Chinese (middle), and English (bottom) Wikipedia article TOYODA_STATION. Since BPE is based on frequency, the resulting segmentation is often, but not always meaningful. For example, in the Japanese text, 豊 (toyo, "abundant") and 田 (ta, "rice field") are correctly merged into 豊田 (Toyoda, a Japanese city) in the second occurrence, but the first 田 is first merged with 駅 (eki, "train station") into the meaningless 田駅 (ta-eki, "*rice field train station"). In the Chinese text, 豊田 is correctly merged, but 豐田站 是 (Toyoda zhàn shì, "Toyoda train station is") is wrongly segmented into 豐田 (Toyoda) and 站是 (zhàn shì, "*train station is". In the English text, *toyoda* is wrongly segmented into *toy* and *oda*, likely due to the fact that *toy* is a frequent English noun, while the correct *toyo* is rare, making it difficult for the BPE model to produce this segment.

We see that BPE first finds articles like *_a* and *_the*, as well as common word beginnings, such as *_t*, *_s*, *_w*. After a few hundred merges, longer symbols appear and include morphemes like *osis* and short words like *_series*. Tens of thousands of merges produce even longer symbols such as *omatosis*.

To segment words into subwords, the merge operations listed in a BPE model are simply repeated in the same order. Depending on the number of merge operations, this yields many short segments or fewer long ones. Examples of the impact of the number of merge operation are shown in Figure 5.5 on the facing page in Table 5.6 . By varying the number of merge operations, we can interpret BPE as an interpolation between characters and words: When applying zero merge operations, the input character sequence is left unchanged. Increasing the number of merge operations creates longer subword segments and after an infinite number of merges, all characters have been merged into words.

_an ar ch is m _is _a _polit ical _ph il os op hy _that _ad v oc ates _s el f - g o vern ed _s oci et ies _b ased _on _v ol unt ary _in stit ut ions . _the se _are _of ten _desc rib ed _as _st at el ess _s oci et ies , _al th ough _several _aut h ors _have _def ined _the m _more _spec if ical ly _as _in stit ut ions _b ased _on _n on - h ier ar ch ical _f ree _associ ations . _an ar ch is m _h old s _the _state _to _be _un d es ir able , _un n ec ess ary , _and _h ar m f ul . _while _an t i - st at is m _is _cent ral , _an ar ch is m _spec if ical ly _ent ail s _op p os ing _aut h or ity _or _h ier ar ch ical _or gan is ation _in _the _con d uct _of _all _h um an _rel ations , _including , _but _not _l im ited _to , _the _state _sy st em . _an ar ch is m _is _us ual ly _cons id er ed _a _r ad ical _le ft - w ing _ ide olog y , _and _m uch _of _an ar ch ist _e c on om ics _and _an ar ch ist _le g al _ph il os op hy _re f lect s _an t i - a ut h or it ar ian _inter p re t ations _of _commun is m , _col lect iv is m , _sy n d ical is m , _m ut ual is m , _or _part icip at ory _e c on om ics . _many _t y p es _and _tr ad it ions _of _an ar ch is m _ex ist

1000 BPE merge operations

_anarch ism _is _a _political _philosophy _that _advocates _self - govern ed _societies _based _on _voluntary _institutions . _these _are _often _described _as _stat eless _societies , _although _several _authors _have _defined _them _more _specifically _as _institutions _based _on _non - h ier arch ical _free _associations . _anarch ism _holds _the _state _to _be _un des ir able , _un n ecess ary , _and _harm ful . _while _anti - stat ism _is _central , _anarch ism _specifically _ent ails _opposing _authority _or _hier arch ical _organisation _in _the _conduct _of _all _human _relations , _including , _but _not _limited _to , _the _state _system . _anarch ism _is _usually _considered _a _radical _left - wing _ideology , _and _much _of _anarch ist _economics _and _anarch ist _legal _philosophy _reflects _anti - author itarian _interpret ations _of _commun ism , _collect iv ism , _synd ical ism , _mutual ism , _or _particip atory _economics . _many _types _and _traditions _of _anarch ism _exist

25,000 BPE merge operations

_anarchism _is _a _political _philosophy _that _advocates _self - governed _societies _based _on _voluntary _institutions . _these _are _often _described _as _stateless _societies , _although _several _authors _have _defined _them _more _specifically _as _institutions _based _on _non - hierarchical _free _associations . _anarchism _holds _the _state _to _be _undesirable , _unnecessary , _and _harmful . _while _anti - stat ism _is _central , _anarchism _specifically _entails _opposing _authority _or _hierarchical _organisation _in _the _conduct _of _all _human _relations , _including , _but _not _limited _to , _the _state _system . _anarchism _is _usually _considered _a _radical _left - wing _ideology , _and _much _of _anarchist _economics _and _anarchist _legal _philosophy _reflects _anti - authoritarian _interpretations _of _communism , _collect ivism , _syndicalism , _mutualism , _or _participatory _economics . _many _types _and _traditions _of _anarchism _exist

200,000 BPE merge operations

FIGURE 5.5: Text encoded with few (top), more (middle), and many (bottom) BPE merge operations. Words that have been segmented into subwords are highlighted. The highlight color indicates whether the word is segmented into meaningful subwords (green), overseg-mented (yellow), or segment wrongly (red). For example, the word *anarchism* is segmented into two meaningful subwords *anarch* and *ism* after 25,000 merges, but not segmented after 200,000 merges. The word *self-governed* is segmented into four meaningful parts *self*, *-*, the stem *govern*, and the passive suffix *ed* after 25,000 merges. The last two parts are further merged into *governed* after 200,000 merges. Common segmentation errors are found in suffixes: After 1000 merges, *political* is segmented into *polit* and *ical* instead of the stem *politic* and the adjectival suffix *al*. Similarly, plural suffixes and third-person singular verb endings are merged into *ates* (instead of *advocate s*), *ors* (instead of *author s*), or *ations* (instead of *relation s*). In summary, we observe that performing few merge operations results in oversegmentation (top), while a high number of merge operations yields very few subwords (bottom).

Since it both reduces vocabulary size and circumvents the unknown word problem, BPE has found widespread use in neural machine translation (Sennrich et al., 2016; Wu et al., 2016). After segmenting the text to be translated into subwords, the first layer of a neural machine translation model learns subword embeddings specific to the training data and language pair. In contrast to `FastText`, for which pre-trained embeddings for general use are available in many languages, no such resource exists for pre-trained BPE-based subword embeddings. Our contribution in the next section addresses this problem.

5.3 `BPEmb` : Byte Pair Embeddings in 275 Languages

One of the main advantages of BPE is that it is applicable to any sequence of symbols. In particular, it can be applied to text, regardless of language.[10] We use this advantage to train subword segmentation models in many languages, then employ these models to segment large text corpora into subwords, and finally train subword embeddings which we publish for general use. This yields a collection of Byte Pair Embeddings in 275 languages, which we call `BPEmb` .[11] Describing each of these steps in more detail, we follow the following procedure:

1. **Text corpus.** To enable learning good BPE models and embeddings, we require a large corpus of texts. We use Wikipedia as corpus and extract plain article texts with `WikiExtract`.[12] After removing Wikipedia language editions with very little content, we obtain article texts in 275 languages.

2. **Preprocessing.** Two preprocessing steps aim to improve BPE model training. We lowercase all characters, since we expect that sentence-initial capitalization, title case, capitalization of nouns, and other case variations are not relevant for subword segmentation. Similarly, we replace all digits with 0 to prevent the BPE model from making irrelevant distinctions between individual numbers.

[10] Whether it is *meaningful* to apply BPE to any language, that is, whether the algorithm learns meaningful subword segmentations in a given language, dependends on the properties of the language and the availability of training data. Also see the limitations discussed in section 5.4.3.

[11] `https://github.com/bheinzerling/bpemb`

[12] `http://attardi.github.io/wikiextractor`

3. **BPE model training.** Having prepared texts in 275 languages, we now learn BPE models using `SentencePiece`.[13] A priori, it is not clear how the number of BPE merge operations should be set. Hence, we train different models with varying numbers of merge operations and evaluate the impact of this hyper-parameter later. Specifically, we train BPE models with 1000, 3000, 5000, 10000, 25000, 50000, 100000 and 200000 merge operations.

4. **BPE subword segmentation.** By applying the BPE models trained in the previous step to the texts in our training corpus, namely, Wikipedia editions in a given language, we obtain subword-segmented texts, as shown in Figure 5.5 on page 115.

5. **Subword embedding training.** Finally, we use off-the-shelf software, namely `GloVe`,[14] to train subword embeddings for each language and each number of merge operations. Since it is not clear what the best embedding dimensionality is, we train embeddings with various dimensions, leaving another hyper-parameter setting to empirical evaluation.

To assess `BPEmb` , that is, the subword embeddings trained according to this procedure, we first perform a qualitative evaluation, before comparing it with other subword representations in a quantitative evaluation in the next section.

5.3.1 Qualitative Evaluation

We first analyze the subword segmentations induced by the BPE models we trained on Wikipedia as described above. Table 5.7 on the next page shows subword segmentations of our English running example *myxomatosis* and its translation in German, Polish, and Japanese. We observe that, for this particular word, our trained BPE models yield reasonable segmentations in three of these languages. However, it also becomes apparent that there is no best number of merge operations which yields good segmentations across several languages, as the arguably best segmentations emerge after 10000 BPE merges for English, 50000 for German, and 100000 for Japanese.

Next, we qualitatively assess our trained subword embeddings by inspecting nearest neighbors in the embedding space. Figure 5.6 on page 119 shows the nearest neighbors of the embedding of the English morpheme *osis* ("disease", "abnormal state"). We find several words and subwords with similar or related meanings:

[13] `https://github.com/google/sentencepiece`

[14] `https://nlp.stanford.edu/projects/glove/`

Language	Merge operations	Subword segmentation
English	1000	_m y x om at os is
	3000	_my x om at os is
	5000	_my x om at os is
	10000	_my x om at osis
	25000	_my x omat osis
	50000	_my x omat osis
	100000	_myx omatosis
	200000	_myx omatosis
German	1000	m y x om at o se
	3000	_m y x om at o se
	5000	_my x om at ose
	10000	_my x om at ose
	25000	_my x om at ose
	50000	_my x omat ose
	100000	_my x omat ose
	200000	_myx omatose
Polish	1000	_m y k so m ato za
	3000	_my k so m ato za
	5000	_my k so m ato za
	10000	_my k so m ato za
	25000	_my k so mato za
	50000	_my k so mato za
	100000	_my kso mato za
	200000	_my kso mato za
Japanese	5000	_ 兎 粘 液 腫
	10000	_ 兎 粘 液 腫
	25000	_ 兎 粘 液 腫
	50000	_ 兎 粘 液 腫
	100000	_ 兎 粘液 腫
	200000	_ 兎 粘液 腫

TABLE 5.7: Learned subword segmentantions for different numbers of merge operations. In English, BPE yields reasonable subword segments of the word *myxomatosis*, which, after 10000 merge operations, include the two morphemes *omat* ("tumor") and *osis* ("sickness"). Similarly, the segmentation of the German *Myxomatose* includes *omat* and *ose*, the German equivalent of the English *osis*. For the Polish *Myksomatoza*, our trained models fail to find any morphemes, producing *mato* instead of *omat* and *za* instead of *oza*, the Polish equivalent of *osis*. In Japanese, the disease has the name 兎粘液腫, consisting of the characters for "rabbit", "sticky", "fluid", and "tumor" (recall that myxomatosis is a tumorous disease in rabbits). The BPE model trained on the Japanese edition of Wikipedia finds the correct segmentation into 兎 ("rabbit"), 粘液 ("mucus"), and 腫 ("tumor").

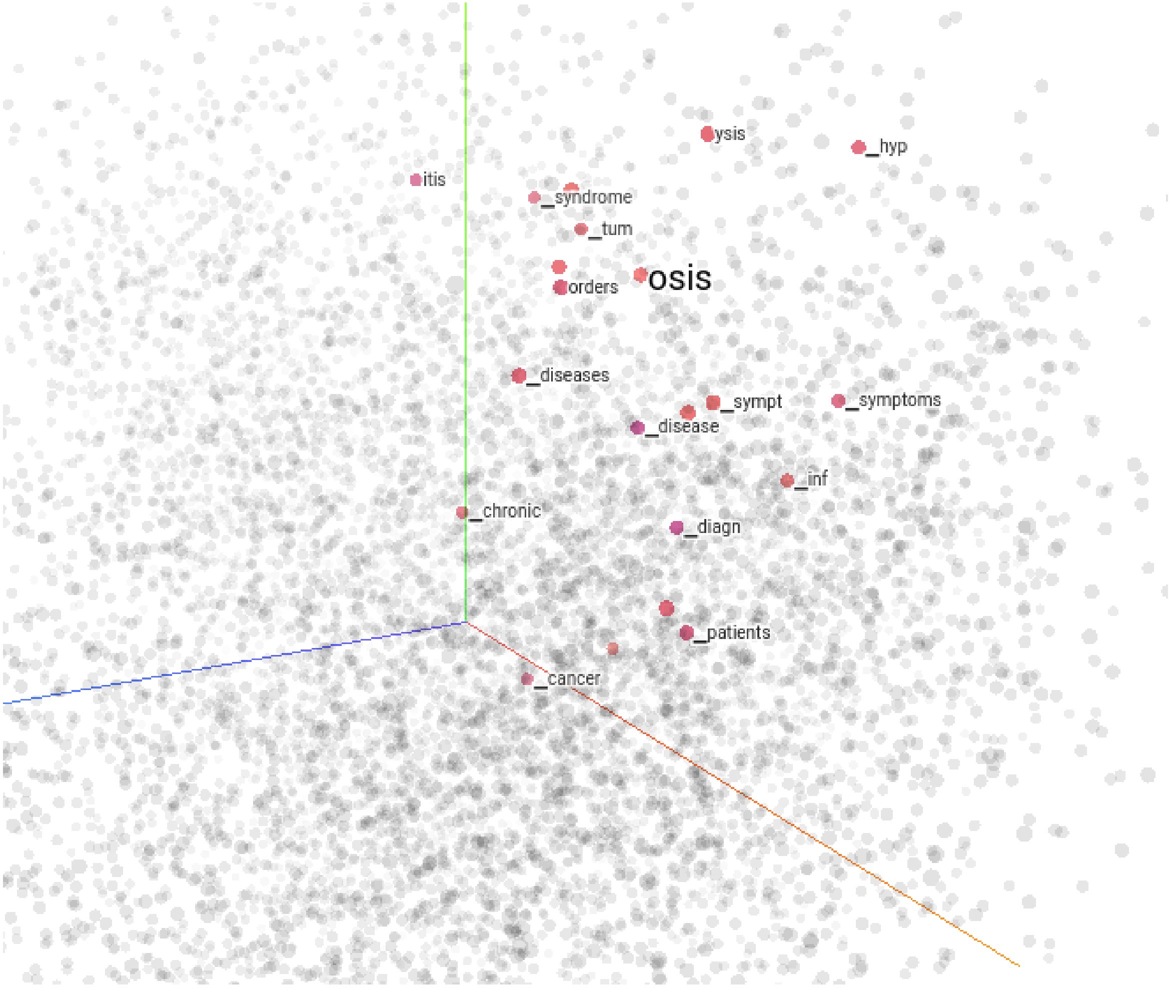


FIGURE 5.6: BPE embeddings most similar to the subword *osis*. t-SNE projection (Maaten and Hinton, 2008) with `http://projector.tensorflow.org`.

- *itis*: a suffix indicating disease, occurring, for example, in *bronchitis*;
- *disease*, *diseases*;
- *_tum*: a character trigram occurring in the word *tumor*;
- *_inf*: a character trigram occurring in the word *inflammation*; and
- related words such as *_symptoms*, *_patients*, *_chronic*.

The fact that the nearest neighbours include the embedding of the subword *_diagn* is an interesting case: On the one hand, the concatentation with *osis* yields *_diagnosis*, which is quite related to the topic of sickness. On the other hand, the subword segmentation implied here is wrong, since the word *diagnosis* originates from the Ancient Greek *día* (“through”, “apart”) and *gnōsis* (“knowledge”).

shire (English)	ingen (German)	ose (German)
ington	lingen	krank
_england	hausen	_erkrank
ford	hofen	itis
_wales	heim	_behandlung
outh	bach	_krankheit
_kent	sheim	hy
bridge	weiler	fekt
well	dorf	pt
_scotland	_bad	apie
orth	berg	_krank

TABLE 5.8: Examples of subword similarities. Shown are subwords most similar to the English morpheme *shire*, which is commonly found in place names in the United Kingdom; subwords similar to the German morpheme *ingen*, which is common in German place names; and subwords similar to the morpheme *ose*, the German equivalent of the English *osis*.

Table 5.8 shows more examples of similar subwords. The first example is the English suffix *shire*, which occurs in place names like *Leicestershire*, *Berkshire*, or *Yorkshire*. Among its most similar subwords, we find similar suffixes:

- *ington*, which occurs, for example, in *Kensington* or *Islington*;
- *ford*, which occurs, for example, in *Stratford*;
- *outh*: which occurs, for example, in *Plymouth*;
- *bridge*: which occurs, for example, in *Cambridge*;

The list also includes related words: *_england*, *_wales*, *_kent*, and *_scotland*. For the German morpheme *ingen*, which is common in names of German cities and villages, all similar subwords commonly occur as word-final morphemes[15] in place names, with the exception of the word-initial *_bad* ("bath"), among whose many word senses is one indicating spa towns, for example in *Bad Säckingen*. Subwords similar to the German *ose* ("osis") include subwords with similar meaning, such *krank* ("sick"), *_erkrank* (word stem, "to become sick"), and *itis*, as well as related words such as *_behandlung* ("treatment").

In the next section, we compare `BPEmb` and other subword approaches.

[15] Strictly speaking, the subword *sheim* is not a morpheme but the concatenation of an epenthetic *s* and the morpheme *heim*.

5.4 Multilingual Entity Typing with Subword Units

5.4.1 Evaluation: Comparison to `FastText` and Character Embeddings

In section 5.2 we discussed subwords as a possible solution to the problem of rare and unknown words in entity typing and introduced characters, character n-grams, FastText, and BPE symbols as subword units. Since character representations are learned during training, pretrained FastText embeddings are publicly available, and pretrained embeddings of BPE-based subwords are one of the contributions of this thesis, we are now ready to compare the utility of these subwords units for entity typing.

5.4.2 Experimental Setup

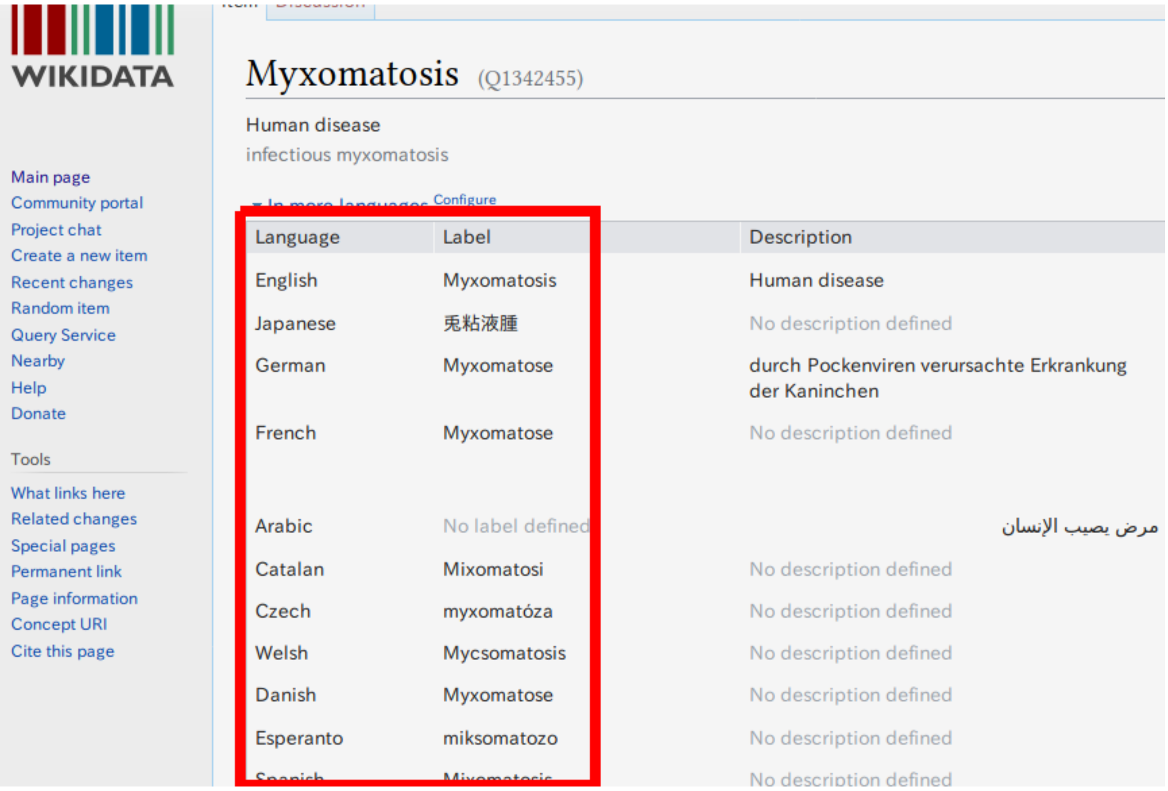
We now describe the setup of our entity typing experiments, in which we compare subword units in multiple languages.

Fine-grained Entity Typing

As discussed in section 5.2, subwords offer to alleviate the problem of rare and unknown words in entity typing. Also recall that a typical neural entity typing system takes an entity mention and its context as input, encodes both the mention and its context into fixed-size vector representations, and then performs entity type classification based on those representations (subsection 5.1.3 on page 104). In order to study the effect of different subword units in isolation, we disregard the context representation in our experiments and perform entity type classification based only the mention representation. For example, given the entity mention *myxomatosis* and using the Google Fine Types inventory (Figure 5.1 on page 102), our task is to label it with the entity type `/other/health/malady`. This experimental setup follows comparisons of subword approaches by Schütze (2017) and Yaghoobzadeh and Schütze (2017).

Data

We collect multilingual entity mentions from Wikidata (Vrandečić and Krötzsch, 2014). Wikidata is database that, among other purposes, provides inter-language links for Wikipedia articles, as shown in Figure 5.7 on the following page. Thus, for a given entity, we can easily obtain entity mentions in all languages for which



Myxomatosis (Q1342455)

Human disease

infectious myxomatosis

Language	Label	Description
English	Myxomatosis	Human disease
Japanese	兎粘液腫	No description defined
German	Myxomatose	durch Pockenviren verursachte Erkrankung der Kaninchen
French	Myxomatose	No description defined
Arabic	No label defined	مرض يصيب الإنسان
Catalan	Mixomatosi	No description defined
Czech	myxomatóza	No description defined
Welsh	Mycsomatosis	No description defined
Danish	Myxomatose	No description defined
Esperanto	miksomatozo	No description defined
Spanish	Mixomatosis	No description defined

FIGURE 5.7: A Wikidata entry showing multilingual entity labels.

a corresponding Wikipedia article exists. To our knowledge, entity types are not available in Wikidata. Hence, we map Wikidata entries to Freebase entries, since Freebase contains rich entity type information.[16] Depending on the size of the Wikipedia edition in a given language, this process yields large numbers of mention-type pairs for high-resource languages and only few such pairs for low-resource ones. The languages with the largest yield are English, French, Dutch, German, Spanish, and Italian, with over 1 million mention-type pairs each. For 132 languages, more than 10000 pairs are available, while for 49 low-resource languages, the yield is fewer than 1000 pairs. Table 5.9 on the next page gives examples of pairs of English entity mentions and entity types obtained via this procedure.

We experiment with subword-based entity typing in 15 languages:

- Five high-resource languages: English, German, Russian, French, and Spanish;

- Two high-resource languages without explicit token markers: Chinese and Japanese; and

[16] We use the mapping available here: `https://github.com/Samsung/KnowledgeSharingPlatform/tree/master/sameas/freebase-wikidata`

Entity mention	Entity types
Pasha Kola, Dasht-e Sar	/location
Ferdosi Mashhad FSC	/organization
Alahärmä	/location
Sin and Bones	/other/art/music
Street Ashton	/location
Pierre-Marie Dupuy	/person
Fomitopsis cajanderi	/other/living_thing
Deh-e Chati	/location
Ilya Kokorev	/person/athlete, /person
Tátrai Quartet	/organization/music, /person/artist
1994–95 First Macedonian Football League	/other/event
Swaziland Democratic Party	/organization
Lyratoherpia	/other/living_thing
Villey-Saint-Étienne	/location/city, /location
Ağbaşlar	/location/city, /location
Les Pennes-Mirabeau	/location/city, /location
Bob Thomas Equestrian Center	/location
Thomas Ridgeway Gould	/person
Piabucina	/other/living_thing
Jean Marchat	/person/artist/actor, /person

TABLE 5.9: Randomly selected pairs of English entity mentions and entity types collected from Wikidata and Freebase.

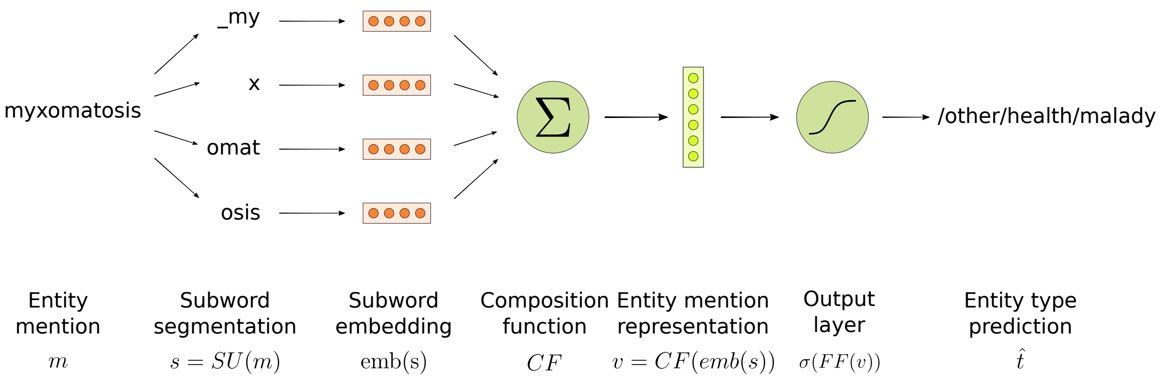


FIGURE 5.8: Architecture of the entity typing model used in our experiments.

- Eight medium- to low-resource languages: Tibetan, Burmese, Vietnamese, Khmer, Thai, Lao, Malay, Tagalog.

Method

Given an entity mention m, our task is to assign one or more of the 89 fine-grained entity types $\hat{t} \subseteq T$, where T is a type inventory. With m = *myxomatosis* and using the Google Fine Types inventory, $\hat{t} = \{$`/other/health/malady`$\}$. If an entity has multiple entity types, all should be predicted. For example, according to Freebase the *Tátrai Quartet* has the entity types `/organization/music`, and `/person/artist`.

Figure 5.8 shows the architecture of our neural entity typing model. Given an entity mention m as input, the function SU segments a word into a sequence of subword units s with length l:

$$s = SU(m) \in \mathbb{R}^{l}$$

Using either pretrained or learnable subword embeddings, we embed this subword sequence into a d-dimensional embedding space:

$$emb(s) \in \mathbb{R}^{l \times d}$$

The length l of the sequence of subwords varies depending on the entity mention m. However, the output layer, which will perform entity type classification, requires a fixed-size representation. To obtain such a fixed-size representation, we apply a composition function $CF : \mathbb{R}^{l \times d} \mapsto \mathbb{R}^{h}$, which encodes the variable-length sequence of subword embeddings $emb(s)$ into a fixed-sized entity mention representation $\mathbf{v}$

of length h:

$$\mathbf{v} = CF(emb(s)) \in \mathbb{R}^h$$

The output layer predicts a score y_i for each entity type t_i in the type inventory T, where $1 \leq i \leq |T|$. Parametrizing the output layer as a feed-forward neural network $FF : \mathbb{R}^h \mapsto \mathbb{R}^{|T|}$ followed by a sigmoid function, the prediction scores y are obtained by:

$$\mathbf{y} = \sigma(FF(\mathbf{v})) \in [0,1]^{|T|}$$

where

$$\sigma(x) = \frac{1}{1 + exp(-x)}$$

Finally, we find the set of predicted entity types $\hat{t}$ by taking all entity types whose predicted score exceeds 0.5:

$$\hat{t} = \{t_i \in T | y_i > 0.5\}$$

In our experiments we compare several subword units and composition functions.

Subword Units

Characters. In this setting, mentions are represented as a sequence of their constituting character unigrams.[17] We learn character embeddings during training, for the k most frequent characters in a given language.

FastText. As described in more detail in subsection 5.2.1, `FastText` enriches word embeddings with subword information by additionally learning embeddings of character n-grams. A word is then represented as the sum of its associated character n-gram embeddings. In practice, representations of unknown words are obtained by adding the embeddings of their constituting character 3- to 6-grams. We use the pre-trained embeddings provided by the authors.[18]

BPE-based subwords. We obtain BPE-based subword segmentations and embeddings of entity mentions via the BPE models and pretrained subword embeddings introduced in section 5.3.

Token. As baseline, we also compare to pretrained word embeddings of each token in an entity mention, without any subword information.

[17] We also experimented with character bigrams and trigrams. Results were similar to unigrams and are omitted here.

[18] `https://github.com/facebookresearch/fastText`

Model	Subword unit(s)	Composition function
Luong, Socher, et al. (2013)	`Morfessor`	recursive neural network
Sperr et al. (2013)	words, character n-grams	addition
Botha and Blunsom (2014)	words, `Morfessor`	addition
Santos and Zadrozny (2014)	words, characters	CNN
Qiu et al. (2014)	words, `Morfessor`	addition
Cotterell and Schütze (2015)	words, morphological analyses	addition
W. Ling et al. (2015)	characters	RNN
Kann and Schütze (2016)	characters, morphological analyses	RNN
Y. Kim et al. (2016)	characters	CNN
Miyamoto and Cho (2016)	words, characters	RNN
(Chiu and Nichols, 2016)	characters	CNN
Rei et al. (2016)	words, characters	RNN
Sennrich et al. (2016)	BPE	none
Wieting et al. (2016)	character n-grams	addition
Bojanowski et al. (2017)	`FastText`	addition
Heigold et al. (2017)	words, characters	RNN, CNN
J. Lee et al. (2017)	characters	CNN
Vania and Lopez (2017)	character n-grams, BPE, `Morfessor`	addition, RNN, CNN
Vylomova et al. (2017)	characters, `Morfessor`	RNN, CNN
Heinzerling and Strube (2018)	characters, `FastText`, BPE	average, RNN, CNN

TABLE 5.10: Overview of subword units and composition functions used in neural models proposed in the literature. `Morfessor` (not discussed in the main text) refers to the subword segementation toolkit by Creutz and Lagus (2002). Table reproduced from Vania and Lopez (2017).

Having introduced different kinds of subword units, we next discuss the three most common composition functions that have been proposed in the literature to obtain word representations from subword representations (see Table 5.10).

Composition Functions

Previous work has primarily studied three kinds of functions for composing subword representations in neural models: addition, convolutional neural networks (CNNs), and recurrent neural networks (RNNs) (Table 5.10). Addition, or similarly, averaging, has the advantage of not requiring any trainable parameters, but loses all positional information. Both convolutional neural networks (LeCun, Bottou, et al., 1998) and recurrent neural networks (Elman, 1990; Hochreiter and Schmidhuber, 1997) are able to retain positional information to varying degrees, but have many parameters and hyperparameters that need to be found during training and hyper-parameter search.

Hyper-parameter Search

As described above, in our experiments we will compare different subword units and composition functions. Subword units differ in their vocabulary size, pretrained subword embeddings differ in embedding dimensionality, and composition functions differ in their number of parameters and hyper-parameters. Furthermore, a particular combination of subword unit and composition function that performed well in one language might not be best for another. To allow for a fair comparison, we conduct an extensive hyper-parameter search for each combination of subword unit and composition function for each language. Specifically, for each combination of subword unit and composition function, we perform a Tree-structured Parzen Estimator hyper-parameter search (Bergstra et al., 2011) with at least 1000 hyper-parameter search trials for the highest-resource language, English, and at least 50 trials for other languages. For each trial and each language, we randomly sample a train and development set of 80,000 and 20,000 mention-type pairs or a proportionally smaller split for smaller Wikipedia editions. Table 5.11 on the next page shows hyper-parameter spaces explored for each subword unit and composition function.

Evaluation

We evaluate entity typing perfomance by taking the average of the `strict`, `loose micro`, and `loose macro` metrics, established for fine-grained entity typing by X. Ling and Weld (2012). After a system made predictions for all entity mentions m in the evaluation set M, we have a set S of system-predicted entity types:

$$S = \{\hat{t}_m | m \in M\}$$

Similarly, the gold annotations G provide gold entity types for each entity mention:

$$G = \{t_m | m \in M\}$$

The evaluation metrics are the F_1 score of three variants of precision P and recall R computed as follows:

- **Strict.** Strict precision and recall require a perfect match between the system prediction and gold annoation for a given entity mention:

$$P = \frac{|G \cap S|}{|S|} \qquad R = \frac{|G \cap S|}{|G|}$$

Subword unit	Hyper-parameter	Space
Token	embedding type	GloVe, word2vec
Character	vocabulary size embedding dimension	50, 100, 200, 500, 1000 10, 25, 50, 75, 100
FastText	-	-
BPE	merge operations embedding dimension	1k, 3k, 5k, 10k, 25k 50k, 10k, 200k 25, 50, 100, 200, 300

Comp. function	Hyper-parameter	Space
RNN	hidden units layers RNN dropout output dropout	100, 300, 500, 700, 1000, 1500, 2000 1, 2, 3 0.0, 0.1, 0.2, 0.3, 0.4, 0.5 0.0, 0.1, 0.2, 0.3, 0.4, 0.5
CNN	filter sizes number of filters output dropout	(2), (2, 3), (2, 3, 4), (2, 3, 4, 5), (2, 3, 4, 5, 6), (3), (3, 4), (3, 4, 5), (3, 4, 5, 6), (4), (4, 5), (4, 5, 6), (5), (5, 6), (6) 25, 50, 100, 200, 300, 400, 500, 600, 700 0.0, 0.1, 0.2, 0.3, 0.4, 0.5
Average	output dropout	0.0, 0.1, 0.2, 0.3, 0.4, 0.5

TABLE 5.11: Subword unit (top) and composition function (bottom) hyper-parameter spaces searched.

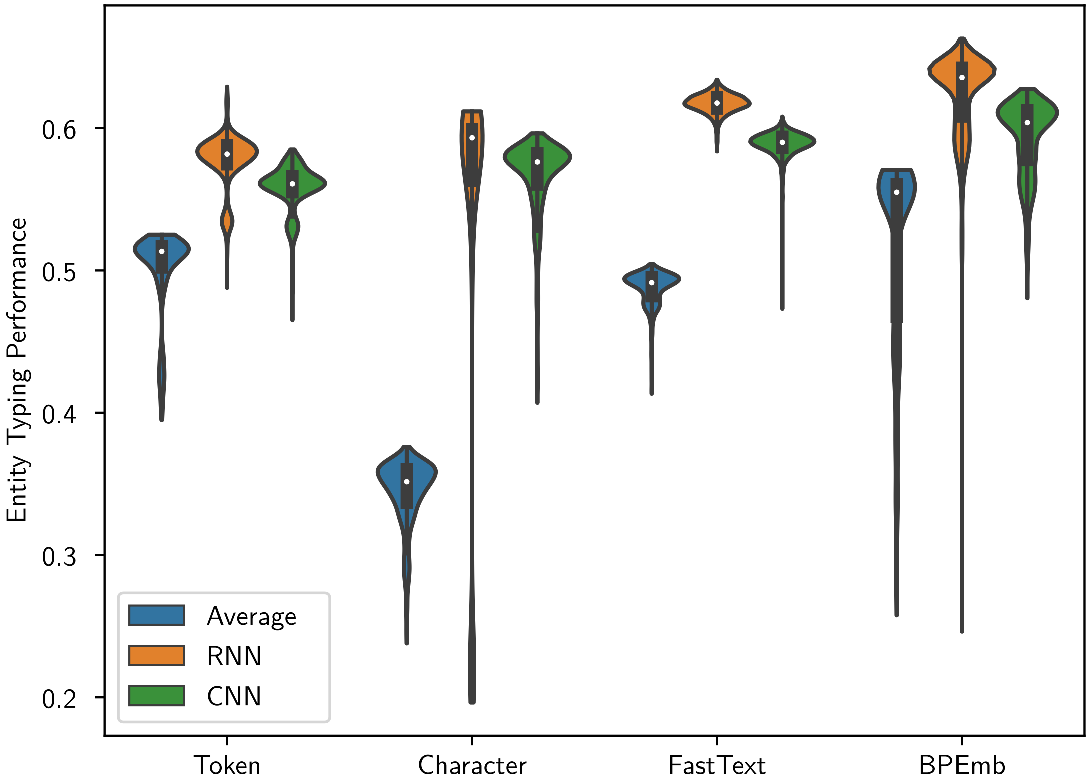


FIGURE 5.9: English entity typing performance of subword embeddings across different composition functions. This violin plot shows smoothed distributions of the scores obtained during hyper-parameter search. White points represent medians, boxes quartiles. Distributions are truncated to reflect highest and lowest scores.

- **Loose micro.** Counts of correct predictions are aggregated over the entire evaluation set before normalization.

$$P = \frac{\sum_{m \in M} |t_m \cap \hat{t}_m|}{\sum_{m \in M} |\hat{t}_m|} \qquad R = \frac{\sum_{m \in M} |t_m \cap \hat{t}_m|}{\sum_{m \in M} |t|}$$

- **Loose macro.** Counts of correct predictions are normalized for each entity mention in the evaluation set, and then aggregated and normalized.

$$P = \frac{1}{|M|} \sum_{m \in M} \frac{|t_m \cap \hat{t}_m|}{|\hat{t}_m|} \qquad R = \frac{1}{|M|} \sum_{m \in M} \frac{|t_m \cap \hat{t}_m|}{|t_m|}$$

5.4.3 Results and Discussion

Figure 5.9 on the preceding page shows our main result for English: score distributions of 1000 hyper-parameter search trials for each subword unit embedding and composition function. Token-based results using two sets of pre-trained embeddings (Mikolov, K. Chen, et al., 2013; Pennington et al., 2014) are included as baseline. We report score distributions for our largest experiment, since distributions provide additional information compared to average scores (Reimers and Gurevych, 2017). We analyze the results in terms of subword units and composition functions.

Comparison of Subword Units

`BPEmb` outperforms all other subword units across all architectures. In combination with an RNN composition function, it achieves a mean entity typing score of 0.624 ± 0.029, with a maximum score of 0.65. FastText performs slightly worse with a mean score of 0.617 ± 0.007 and maximum of 0.63, even though the FastText vocabulary is much larger than the set of BPE symbols.[19]

As shown in Figure 5.10 on the next page, `BPEmb` performs well with low embedding dimensionality and can match FastText with a fraction of its memory footprint. The English FastText model has a vocabulary size of three million and a file size of 6 GB, while the 25-dimensional `BPEmb` model with a vocabulary size of 100000 and a file size of 11 MB matches its performance in this task. As both FastText and `BPEmb` were trained on similar corpora, namely, different versions of the English edition of Wikipedia, these results suggest that, for English, the compact BPE representation strikes a better balance between learning embeddings for more frequent words and relying on compositionality of subwords for less frequent ones. The number of BPE merge operations has a considerable impact, with more merges giving better results (Figure 5.11 on page 132).

FastText performance shows the lowest variance, that is, it robustly yields good results across many different hyper-parameter settings. `BPEmb` and character-based models show higher variance and therefore require more careful hyper-parameter tuning to achieve good results.

Comparison of Composition Functions

Averaging a mention's associated embeddings is the worst choice of composition function. This is expected when using characters as subword units, since averaging

[19] Difference to `BPEmb` significant, $p < 0.001$, Approximate Randomization Test (Noreen, 1989).

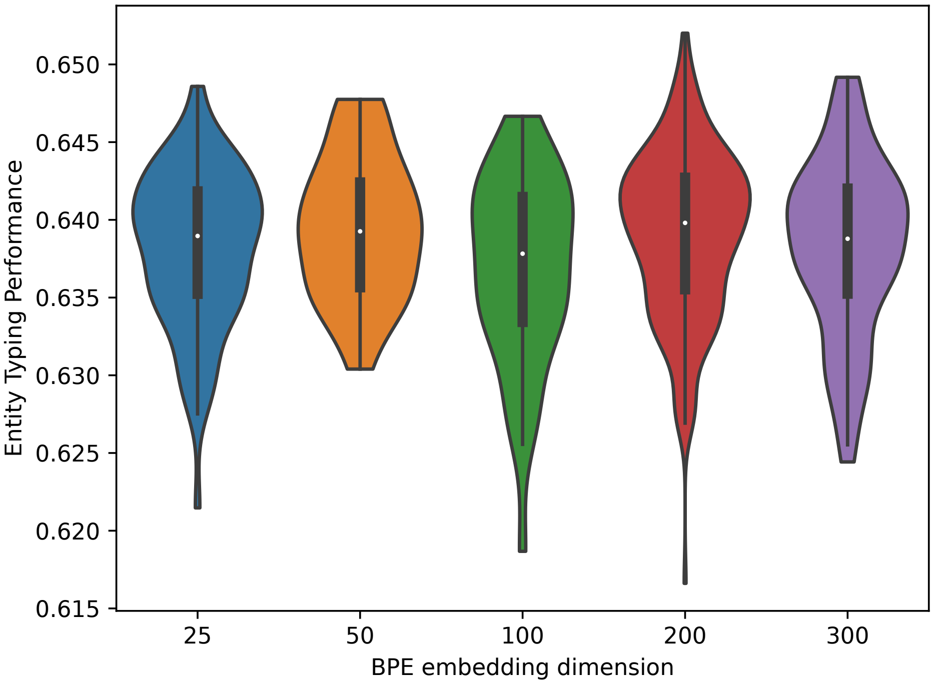


FIGURE 5.10: Impact of the BPEmb embedding dimension on English entity typing.

loses all positional information. The bad performance of averaging tokens is somewhat unexpected, given the fact that averaging is a common composition function for entity mentions in entity typing (Shimaoka et al., 2017) and coreference resolution (Clark and Manning, 2016b). RNNs perform slightly better than CNNs in combination with all subword embeddings, but come with the cost of much longer training time.

Multilingual Results

Table 5.12 on the next page compares FastText and BPEmb across various languages. For high-resource languages (top) both approaches perform equally, with the exception of BPEmb giving a significant improvement for English. For high resources languages without explicit tokenization (middle), byte-pair encoding appears to yield a subword segmentation which gives performance comparable to the results obtained when using FastText with pre-tokenized entity mentions.[20]

[20] Tokenization for Chinese was performed with CoreNLP (Manning et al., 2014) and for Japanese with Kuromoji (https://github.com/atilika/kuromoji).

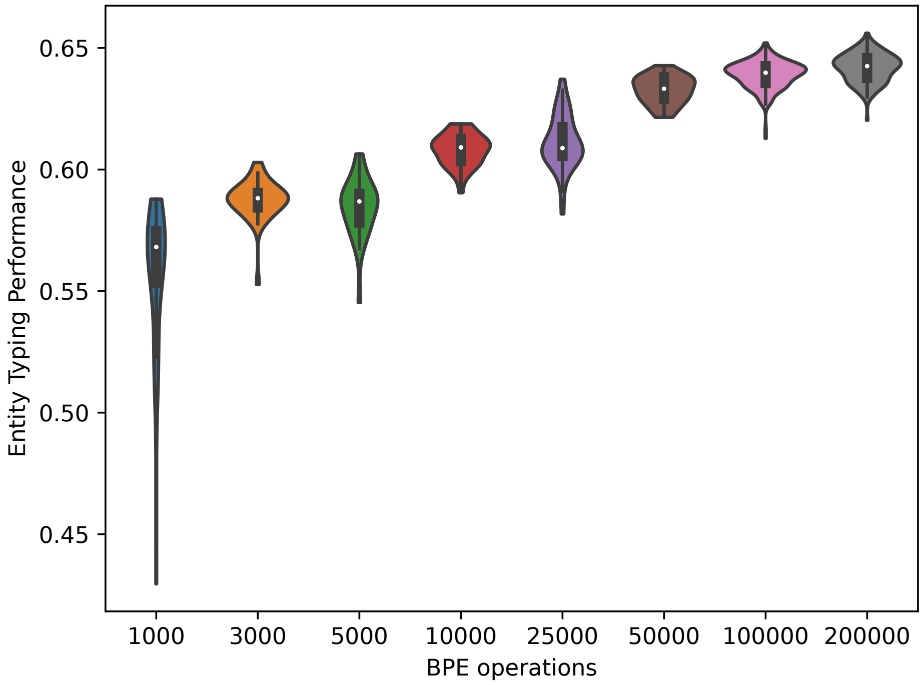


FIGURE 5.11: Impact of the number of BPE merge operations on English entity typing.

Language	FastText	BPEmb	Δ
English	62.9	**65.4**	2.5
German	65.5	**66.2**	0.7
Russian	**71.2**	70.7	-0.5
French	**64.5**	63.9	-0.6
Spanish	**66.6**	66.5	-0.1
Chinese	71.0	**72.0**	1.0
Japanese	**62.3**	61.4	-0.9
Tibetan	37.9	**41.4**	3.5
Burmese	**65.0**	64.6	-0.4
Vietnamese	81.0	81.0	0.0
Khmer	**61.5**	52.6	-8.9
Thai	63.5	**63.8**	0.3
Lao	44.9	**47.0**	2.1
Malay	75.9	**76.3**	0.4
Tagalog	**63.4**	62.6	-1.2

TABLE 5.12: Entity typing scores for five high-resource languages (top), two high-resource languages without explicit tokenization (middle), and eight medium- to low-resource Asian languages (bottom). All values in percent.

Results are more varied for the mid- to low-resource languages in our experiments (bottom), with small `BPEmb` gains for Tibetan and Lao. The large performance degradation for Khmer appears to be due to inconsistencies in the handling of Unicode control characters between different software libraries used in our experiments and have a disproportionate effect due to the small size of the Khmer Wikipedia.

Limitations

Our study of subword units and composition functions for multilingual entity typing has several limitations. Due to limited computational resources, our evaluation was performed only for a few of the many languages in which both `BPEmb` and FastText embeddings are available. While our experimental setup allowed for a fair comparison between FastText and `BPEmb` through extensive hyper-parameter search, it is somewhat artificial, since it disregards context. For example, *Myxomatosis* in the phrase *Radiohead played Myxomatosis* has the entity type `/other/music`, which can be inferred from the contextual music group and the predicate *plays*, but this ignored in our specific setting. How our results transfer to other tasks requires further study.

Our application of BPE to entity typing has shortcomings, as well. By merging frequent *adjacent* symbol pairs, BPE makes a locality assumption, that is, it assumes compressible patterns take the form of contiguous symbol sequences. By applying this algorithm to text in a given language, we thereby assume meaningful words and subwords in this language are formed by concatenation of smaller units. However, this is not the case for languages with non-concatenative morphology. For example, triliteral roots in Semitic languages are patterns consisting of three consonants that are combined with vowels or other consonants to form words. The root *k-t-b* relates to the concept of "writing" and forms words such as the Arabic *kataba* ("he wrote"), *kātib* ("writer"), *kitāb* ("book"), and *kutub* ("books"). Since the root *k-t-b* does not occur in isolation, but only in combination with other vowels and consonants, BPE cannot identify it as the pattern common to all these words.

Finally, the subword embeddings in our experiments are not contextualized. For example, in the BPE segmentation *_my x omat osis*, the subwords most similar to *_my* are *_your*, *_you*, *my*, *_me*, *you*, that is, *_my* is interpreted as the first person possessive pronoun. This meaning is, of course, irrelevant, and we need to rely on the composition function to learn the meaning of the word in spite of this misleading subword embedding. See Appendix C on page 141 for more such examples.

5.4.4 Summary

In this chapter, we introduced `BPemb`, a collection of subword embeddings trained on Wikipedia editions in 275 languages and compared it to other subword approaches using entity typing as a test bed. Entity typing is an important task, since it provides information about entities that are not contained in the knowledge base. Entity typing is a suitable test bed for subword evaluation, since many rare, long-tail entities do not have good representations in common token-based pre-trained embeddings such as word2vec or `GloVe`. Subword-based models are a promising approach for this task, since morphology often reveals the semantic category of unknown words: The suffix *-shire* in *Melfordshire* indicates a location or city, and the suffix *-osis* in *myxomatosis* a sickness. Subword methods aim to allow this kind of inference by learning representations of subword units such as character n-grams, morphemes, or byte pairs. Our evaluation showed that `BPEmb` performs as well as, and for some languages, better than other subword-based approaches. `BPEmb` requires no tokenization and is orders of magnitudes smaller than alternative embeddings, enabling potential use under resource constraints, e.g. on mobile devices.

Chapter 6

Conclusions

In this thesis we investigated different aspects of coherence with regards to their impact on the three tasks comprising entity analysis, namely entity linking, coreference resolution, and entity typing. In entity linking, both the interactions between subtasks and the use of coherence in global disambiguation pose computational challenges. As an answer to the former problem and our first research question, we proposed an interleaved multitasking approach. This approach allows a certain degree of free interaction between interdependent tasks, while avoiding the computational cost of joint multitasking. We implemented this approach in a simple, rule-based entity linking system and demonstrated its effectiveness in the English portion of the TAC 2015 Entity Discovery and Linking shared task.

Adressing the challenges of incorporating global coherence into entity linking system and our second research question, we introduced automatic verification as a post-processing step for entity linking systems. This allowed us to formulate specific, knowledge-rich measures of global coherence, which lead to consistent improvements across all evaluated entity linking systems.

Turning to error analysis, we answered our third research question by developing a visualization tool tailored to entity linking system developers. Our tool employs Euclidean minimum spanning trees to achieve a more concise visualization of entities than alternative methods.

While we devised a model of selectional preferences with much higher coverage than prior work, the answer to our fourth research question is inconclusive. Integrating this model into a neural coreference resolution system leads to a small improvement in performance, but this improvement is too small to claim progress in the long-standing uphill battle of incorporating semantics into coreference resolution.

Finally, addressing our fifth research question, we performed an extensive evaluation of different subword units in entity typing. We saw that FastText and Byte-Pair embeddings performed best, with the Byte-Pair-based approach striking a good balance between learning representations of frequent words, and relying on subword composition for less frequent ones.

6.1 Outlook

Our work opens several avenues for future research.

A dynamic version of interleaved multitasking. In our implemenation of interleaved multitasking, the order of each group of decisions is fixed, in a fashion similar to precision-ordered sieves in coreference resolution. A possible improvement over this fixed order is to select the next action to be performed dynamically, based on the decisions up to the current state.

Other aspects of global coherence in entity linking. We introduced geographical, temporal, and entity type coherence as specific aspects of coherence and show their utility in entity linking. Are there other aspects of coherence usefule for entity linking?

Integration into a joint entity analysis system. In this work we studied the impact of aspects of coherence on the individual subtasks of the full entity analysis task. An important goal for future work is the integration of these aspects into an entity analysis system that tackles the three subtasks jointly or in an interleaved multitasking fashion in order to fully exploit subtask interactions.

Appendix A

BPE model trained on English Wikipedia with 1000 merge operations

```
<unk>
<s>
</s>
_t
_a
he
in
_the
00
er
_s
on
_c
re
_o
_w
is
an
_in
ed
_f
_b
or
at
en
it
_p
ar
al
_of
_00
_m
_an
as
es
_and
_d
ic
ro
_h
_0000
ing
_l
ion
_to
il
un
_n
_is
am
le
_g
_re
us
_was
ent
ur
_co
el
_he
_e
_on
ut
ad
ol
st
ac
_st
et
iv
_for
_(
_"
ec
ay
ation
_th
_as
ir
_r
id
_it
ist
ly
ot
ig
_j
ch
_v
_al
em
_k
ers
ow
_at
_by
ul
im
and
ith
ag
ter
rom
ct
_with
om
_ch
um
ce
_0
_com
_wh
_un
_be
_from
_con
th
est
ain
ber
os
oc
ia
ian
_pro
_0000.
_pl
her
ies
all
_se
art
_de
ver
ip
op
ill
ity
_0000,
_his
ud
ub
ab
av
ated
ap
ug
up
res
_y
_fo
ate
_or
_ne
_sp
_mo
_are
_ro
if
_le
ak
ard
und
ri
ich
qu
ort
out
ary
od
ast
igh
_ar
so
ish
ame
ge
ong
_that
ess
_tr
ric
ant
ive
_sh
_comp
mer
ear
ous
_en
ire
orn
ord
ell
_play
ine
unt
_has
_cl
ust
uc
_also
_wor
_te
ov
ie
_ex
ember
_mar
_ser
_ap
age
ial
cl
_af
_fir
ment
der
_which
ere
_000
ect
_br
_first
ions
og
ue
_part
ure
land
ld
_new
ugh
ren
0000
_were
_she
ational
_its
_ad
ack
_00,
ical
_sc
ore
_this
_col
wn
ra
man
own
_whe
_tw
ok
_comm
hed
amp
se
ight
_dist
lect
fer
ond
_one
are
ass
_bo
ry
end
_man
_us
_who
_ac
ang
_ind
_count
_sch
_loc
ign
ide
_jo
ice
",
tern
ach
clud
_sout
ree
urn
_rec
_may
_year
iz
),
per
_after
).
_app
_cont
_ab
ther
ts
ased
_her
_fe
ool
_res
000
_rep
_includ
ater
ball
_not
_nor
".
_two
ans
ens
ubl
low
_pr
hip
ail
br
_serv
pt
ome
ivers
_can
ey
ace
_par
_gr
ition
amil
_per
te
_had
_cent
_distric
ave
ates
inal
ited
to
_reg
_fil
ik
_ag
io
_school
_go
_kn
ied
_0000)
_team
_but
_work
ason
we
_district
ory
_des
ild
ral
aw
_mus
uring
_alb
_bu
_their
ade
_univers
_ed
_all
_bec
_film
_spec
_fr
_pol
ican
_south
ib
_0,
_national
rit
les
ward
_gro
_po
_sec
ime
_famil
uct
act
_gen
ince
ton
_found
ance
_foot
_been
old
_pre
ult
_ass
ations
_pop
ence
az
the
_album
_university
uary
_amer
_season
_so
_vill
ite
ished
_rele
_they
_north
ah
_qu
ff
_off
_other
for
_act
_bro
ath
_eng
_county
_(0000
way
_up
_football
_elect
_ph
iss
_pres
_known
_played
_z
_dec
_during
_me
ings
_world
_stud
rop
ely
ted
lo
_record
_publ
_car
anc
vent
hn
_fol
_have
_sing
port
ual
_min
ics
_pe
_prov
born
_popul
_family
_am
_there
_i
ae
_bet
ague
ile
rib
_located
gan
ional
_win
_under
_village
ors
ative
_state
_over
ision
_hous
_med
_prod
_town
_united
vel
_form
_cons
_follow
ase
_out
oth
_bl
ank
.0
_city
_gre
_sub
_champ
_mon
able
_american
ina
_high
ames
one
_where
_dis
```

```
ep
_second
_im
_sy
_when
irect
_league
uch
co
sh
ork
_born
_ear
une
che
_name
_years
ick
_op
ier
_ma
iver
iel
_nov
ft
ake
_we
ix
_into
mp
_direct
_art
_refer
_song
_three
_lo
ween
_released
_hist
_won
_club
_between
_most
eral
_war
enc
red
_u
olog
ident
_dr
_fin
_species
_cur
son
_ret
_music
ved
_states
bo
_time
_sup
_do
unc
gr
oci
_will
_-
ious
_gu
ne
_four
_cap
_sm
_award
rent
ten
_aust
_station
ress
_prof
_made
_list
icip
_champions
_only
_area
_met
_compet
_dep
ley
_mat
_jan
_att
any
ock
ung
_later
_population
_thro
ons
ural
_house
_member
_group
_became
form
iam
_west
ose
amed
_num
_again
ents
ale
_bar
air
_air
_end
_char
til
_wo
_no
ict
min
ived
_acc
_div
_writ
str
_ent
_est
_profess
emb
_former
_about
_following
fore
_colle
ve
_ju
ann
uld
_ger
_then
_through
enn
_intern
ket
_lar
als
_until
_sw
_bel
_fl
ute
_used
_main
_ter
_inter
_def
_championship
_ge
_trans
ury
vern
_some
_band
ating
hern
ss
_more
_well
urch
_river
pl
_cre
_orig
ven
ugust
by
_event
_str
_since
ough
_series
int
ai
stit
ery
ister
_march
_john
_before
ph
_served
_feat
_commun
_college
_oper
ost
_set
_repres
ck
_la
_august
_current
_company
_el
_including
ind
_east
_austral
_sur
_build
_january
ril
ield
_german
ities
ptember
_govern
round
_rem
_call
ife
_ir
_rel
ober
_july
inc
sp
_oct
ople
ific
_mun
ilt
_june
our
igned
_international
tle
_september
_run
ains
let
ll
_held
ese
ired
_open
ke
_rom
ality
_him
_represent
ins
ures
_people
uss
_many
_final
-00
_october
_named
ning
ys
_april
_rece
ular
_cal
_own
_brit
_long
_number
_show
eng
velop
_sever
ium
cess
_start
_book
_tit
-00
_november
_aut
ement
ird
ized
_canad
_near
_head
ital
ty
_island
_care
ists
lev
_york
_var
ner
_being
_december
_both
_such
ather
_ann
ica
_histor
_church
_against
_develop
_inc
ior
_best
its
_produc
ined
_how
_while
urg
lish
bruary
_cr
_original
de
_eu
_lead
_yo
_park
_several
hy
_municip
ained
ull
ark
af
_desc
cc
ert
ash
_single
au
_-
_polit
ject
ike
_add
ont
_general
_sum
_mil
_mid
idae
_games
_offic
_associ
_small
_built
ales
_
e
a
t
i
n
o
s
r
h
l
d
c
0
m
u
f
p
g
b
w
y
.
,
v
k
"
j
-
(
)
z
x
’
q
:
–
;
/
é
%
&
ó
>
<
á
’
=
ā
í
$
!
ü
ö
ł
#
ä
–
ī
ø
è
"
"
_
ć
]
2
[
š
ç
ō
|
ū
č
+
ã
ñ
ń
?
ı
‘
â
ú
å
É
ă
°
а
ę
*
ş
И
£
о
ô
Ł
à
ș
ë
ž
е
Š
ą
ś
Н
|
α
æ
р
×
ź
ß
В
ê
Á
Ö
ż
Ś
ğ
```

Appendix B

BPE model trained on German Wikipedia with 1000 merge operations

```
<unk>
<s>
</s>
er
en
d
ch
in
00
s
a
ie
un
st
ei
b
an
00
w
on
ein
der
sch
te
ge
m
v
un
ar
0000
i
in
f
or
es
ur
die
und
al
it
ein
k
de
er
n
z
at
h
ich
au
p
el
l
em
as
isch
gen
ter
hr
r
ol
ge
am
au
t
g
ten
j
is
von
ung
il
im
der
den
ber
an
(
ach
st
um
al
et
mit
den
ist
ier
ver
and
"
ig
des
om
be
us
wur
le
lich
ür
ion
das
eit
ir
ra
sch
o
ent
wurde
war
ann
ischen
ro
c
auf
ck
ut
rei
ik
als
re
ft
dem
ahr
aus
iel
für
bei
zu
00.
sein
am
ben
sp
sie
eine
nach
0
ien
se
sen
hn
sp
e
de
ist
ad
kt
scha
ver
jahr
bis
wei
ort
ster
he
hl
ul
sich
re
ger
os
ell
ff
ab
ün
än
so
eich
ab
schaft
men
ation
eil
sten
ern
unter
ot
fr
ar
ag
lie
auch
ische
ri
icht
ne
zum
ing
ste
ru
ät
ind
och
ör
0.
all
op
ne
über
und
tel
vor
rie
ang
le
ber
be
utsch
ch
mein
urch
über
im
me
la
pro
um
ser
gt
es
ord
wir
ent
gr
se
einer
ic
durch
deutsch
zur
art
etz
aus
og
we
000
ischer
end
adt
ers
ud
ap
ungen
ierte
zwei
ungs
reich
0000
iv
wer
we
hm
(*
ihr
itz
hat
igen
gr
sind
br
det
uch
wa
är
hen
ia
tet
ler
ma
land
ts
anz
ige
).
gel
urg
gemein
fol
land
wie
lie
rä
spiel
spiel
wer
kon
her
wird
einem
iver
ze
eister
wei
geb
„
qu
iert
her
elt
pp
seit
ed
ant
lin
en
nen
stell
sel
rat
ht
hal
uss
dem
ov
mal
wie
lichen
gew
id
ug
geb
ück
auf
sü
bei
ak
ges
ho
ühr
che
ame
kl
ohn
ess
iz
zw
reis
stadt
mar
fl
chen
amm
einen
atz
ub
sa
ard
ivers
),
werden
oder
ill
seine
ational
burg
omm
ass
iversit
oll
del
teil
fl
ner
ust
hör
da
bes
pt
tern
tete
ok
annt
folgen
od
na
fol
ank
ften
ow
alt
vom
zeich
or
fe
tem
reg
nicht
ahl
berg
sk
ähr
",
stand
süd
ort
he
pf
rit
gemeinde
gen
pol
et
glie
uf
att
ön
sowie
seiner
ke
ember
(0000
mus
universit
orm
".
ste
gro
stud
wurden
ium
tes
ilm
ei
zeit
rich
io
gehör
wo
000
äl
enn
pl
amil
liche
bet
jahre
ob
ha
mann
ßen
beg
ib
ian
zeit
net
sta
bef
ander
ay
mehr
us
amen
ekt
gl
ße
†
zwischen
dar
eter
te
liegt
weiter
hei
meister
ro
nord
ick
onen
pl
orf
hmen
äch
pers
pr
schl
hau
ym
ast
üt
du
gegen
ild
the
schaften
na
ahn
ph
of
jo
iet
oren
zen
ikan
ss
ute
gl
ho
zus
per
ammen
cht
arl
arb
tra
erst
famil
ark
bek
ale
neu
if
hem
ließ
mitglie
jahren
iger
drei
iker
welt
el
```

```
man
verb
gte
ater
hatte
ähl
u
")
ied
spä
gesch
teil
dort
olog
utz
ac
rach
jan
aut
qu
öff
schl
ges
ersten
wohn
mer
beim
ausge
eits
ober
ktion
,0
ßer
issen
atur
dies
amer
ha
arbei
jun
später
bl
ial
lei
stlich
stra
fin
ober
je
eines
ks
verw
bau
all
arr
personen
undes
öl
nur
hei
y
vor
noch
ont
änd
spieler
ball
zusammen
ett
fen
führ
ison
etwa
vier
0000/
bar
kir
kön
pr
wieder
hin
folgender
etzt
mitglied
our
ule
uar
ph
film
ins
ger
dieser
ine
ös
rupp
under
ip
0000.
iss
verein
erfol
kl
fer
juni
mär
aug
bru
dete
waren
bau
universität
aber
gesch
ce
verl
kom
fr
januar
mit
ingen
uß
grün
mo
erh
franz
national
elle
ron
jul
urop
olle
ult
km
la
aten
haus
ma
deutscher
ur
oss
märz
äu
haupt
ben
ieder
amt
einwohn
co
zeichnet
name
seinem
stadt
gemein
-
ätig
itel
ielt
august
ald
diese
kte
dan
bo
kur
orden
ton
hö
dern
polit
dam
inz
ans
me
-0000
räs
0000/00
berlin
was
hol
iga
enz
mai
hunder
okt
bundes
fe
nov
ieren
cke
einz
saison
kreis
sta
0000)
juli
ierung
oktober
ka
rieb
fel
sel
führ
jahrhunder
seinen
klein
dass
("
grün
pro
befin
steht
fahr
lung
gra
studierte
kam
ens
stein
bel
min
gel
öffent
ol
tal
of
mün
anden
name
sep
gewann
dez
mal
mann
keit
az
ange
deutschen
währ
apr
iele
dezember
rieg
ex
eben
april
bach
br
ou
entr
tr
tember
während
hau
prof
september
neben
for
zer
intern
nieder
familien
bl
ken
ihre
spielte
ieden
ina
0000,
str
th
region
ats
jed
spr
november
mus
chsel
platz
ichte
ommen
stel
eigen
iden
ka
namen
ra
schw
da
konn
erste
min
nung
zu
par
schule
rift
gan
tätig
iste
orn
fall
dr
tra
erhielt
ffen
amerikan
tschaft
chr
sterreich
bruar
gleich
nahm
februar
ite
di
heute
th
versch
frei
musik
iere
ition
vert
hoch
erreich
part
form
europ
yn
komm
old
als
son
weise
bt
zurück
glich
Ü
urn
gend
zir
ehem
ys
französ
stra
alen
begann
(0000-0000
jedoch
unde
pen
aut
dann
heit
rech
je
nis
ander
beste
ker
mon
hof
international
up
entw
öffentlich
isten
ille
best
ählt
ersch
wechsel
groß
man
geben
fuß
lu
unden
assen
unt
frank
bekannt
leben
e
n
i
r
s
a
t
d
l
h
u
0
o
m
c
g
b
f
k
w
.
p
z
v
,
ü
ä
"
j
-
y
ö
)
(
ß
:
x
*
;
q
"
„
é
–
/
†
á
Ö
Ü
’
í
ó
’
%
>
_
<
Ä
è
&
2
š
ō
=
ç
č
ø
ł
É
ã
ı
°
ô
â
ć
!
ú
[
]
ý
ñ
å
à
+
‘
Š
ř
ū
ž
3
×
a
'
ā
ė
?
```

Appendix C

Most Similar Subwords for Various BPE Segmentations of *myxomatosis*

Language	Merge operations	Subword segmentation						
		_m	y	x	om	at	os	is
English	1000	ill	ys	t	ot	_s	_p	on
		_p	on	es	at	id	ol	il
		_s	od	ver	et	as	es	iss
		ag	ies	_app	on	est	ot	_le
		_d	ry	_d	_rec	t	as	um
		_my	x	om	at	os	is	
English	3000	_love	_x	ec	ar	ro	us	
		_god	0	od	ad	ol	ro	
		my	ph	ol	as	as	on	
		_you	(	ice	ab	oc	ig	
		_heart	+	ot	um	ios	as	
		_my	x	om	at	os	is	
English	5000	_love	_x	ot	ad	ios	us	
		hing	0,	op	as	ro	ib	
		_your	0	og	ra	ros	ias	
		_you	0.	rom	am	ac	ol	
		...	(	cul	ar	o	it	
		_my	x	om	at	osis		
English	10000	_your	_x	og	as	_disease		
		_you	0,	od	ad	_diagn		
		you	ix	ot	ab	itis		
		?"	ox	ym	ap	_cancer		
		_love	c	op	ar	_treatment		
		_my	x	omat	osis			
English	25000	_your	_x	ophy	emia			
		_you	0	ophora	itis			
		you	0,	oma	_disease			
		.’	ix	tox	_hyp			
		_me	ax	_rhiz	_tumor			
		_my	x	omat	osis			
English	50000	_your	_x	tox	_disease			
		_you	0	ocon	_hyp			
		my	000	emat	_symptoms			
		_me	0,	neum	itis			
		you	0)	ople	otic			
		_myx	omatosis					
English	100000	ozo	omatous					
		_pinoy	_granul					
		_helminth	_papill					
		omatosis	omas					
		_cardi	_cutaneous					
		_myx	omatosis					
English	200000	_pinoy	_granul					
		_helminth	_angi					
		edema	_choroid					
		myx	_leiomy					
		_flukes	_hemangi					

TABLE C.1: Most similar subwords for each subword in different segmentations of the word *myxomatosis* in English and German.

Language	Merge operations	Subword segmentation						
		_m	y	x	om	at	o	se
German	1000	m	ys	_c	ob	ist	ic	ser
		_br	_y	_te	il	ik	ac	sen
		3	yn	_co	ron	ar	_p	kl
		ill	unt	ian	eter	hem	ro	hen
		_000	_of	ite	_aut	sta	or	au
		_m	y	x	om	at	o	se
German	3000	m	ry	ix	yt	al	io	sen
		_d	ys	ure	op	if	ino	ven
		_0,0	ce	cl	ot	it	os	ne
		_h	ly	ac	ys	im	ano	ro
		_mill	_pe	_x	yn	pt	co	de
		_my	x	om	at	ose		
German	5000	_z	_assoc	tin	Т У	_ange		
		ary	_cre	famil	μ	stlich		
		ächsischen	förmig	_bahnstrecke	_salz	_gem		
		ordnung	–0000	gruppen	lern	elle		
		öl	schau	kultur	atte	_benannte		
		_my	x	om	at	ose		
German	10000	ome	cl	od	it	krank		
		ep	_x	op	ol	_erkrank		
		_you	c	ot	id	itis		
		od	ox	yt	il	_behandlung		
		_thr	ix	ol	am	_krankheit		
		_my	x	om	at	ose		
German	25000	my	_x	od	id	_häm		
		_you	_c	ot	it	ämie		
		_thr	cl	op	ap	_leber		
		ep	c	ok	ir	fektion		
		_love	0	ob	ul	osen		
		_my	x	omat	ose			
German	50000	my	_x	ophyll	itis			
		_your	0	_pseud	_erkrankung			
		_you	cl	zyt	ase			
		_heart	ix	heter	osen			
		_to	_c	osis	_leber			
		_my	x	omat	ose			
German	100000	my	_x	_granul	osen			
		_your	0	opath	om			
		_you	c	_opt	ide			
		_this	p	ophyllum	ase			
		_love	ix	_retin	_oste			
		_myx	omatose					
German	200000	obakterien	_granul					
		omyceten	ikose					
		ophaga	_aerod					
		ogast	ozyt					
		ichthy	ilch					

TABLE C.1: Most similar subwords for each subword in different segmentations of the word *myxomatosis* in English and German (continued).

Appendix D

Code and Data Used in this Thesis

The code and data used in this thesis has been published with the following digital object identifiers:

Benjamin Heinzerling (2019c). *Source Code, Data and Additional Material for the Thesis: "Aspects of Coherence for Entity Analysis"*. DOI: `10.11588/data/9JKAVW`. URL: `https://doi.org/10.11588/data/9JKAVW`

Benjamin Heinzerling (2019b). *Selectional Preference Embeddings (EMNLP 2017)*. DOI: `10.11588/data/FJQ4XL`. URL: `https://doi.org/10.11588/data/FJQ4XL`

Benjamin Heinzerling (2019a). *BPEmb: Pre-trained Subword Embeddings in 275 Languages (LREC 2018)*. DOI: `10.11588/data/V9CXPR`. URL: `https://doi.org/10.11588/data/V9CXPR`

Bibliography

Achtert, Elke, Ahmed Hettab, Hans-Peter Kriegel, Erich Schubert, and Arthur Zimek (2011). "Spatial Outlier Detection: Data, Algorithms, Visualizations". In: *Advances in Spatial and Temporal Databases - 12th International Symposium, SSTD 2011, Minneapolis, MN, USA, August 24-26, 2011, Proceedings*, pp. 512–516.

Agirre, Eneko and David Martinez (2001). "Learning class-to-class selectional preferences". In: *Proceedings of the ACL 2001 Workshop on Computational Natural Language Learning (ConLL)*, pp. 15–22. URL: `http://aclweb.org/anthology/W01-0703`.

Attardo, Salvatore (2005). "The role of affordances at the semantics/pragmatics boundary". In: *Proceedings of the Cognitive Science Society*. Vol. 27. 27.

Auer, Sören, Christian Bizer, Jens Lehmann, Georgi Kobilarov, Richard Cyganiak, and Zachary Ives (2007). "DBpedia: A nucleus for a Web of open data". In: *Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference,* Busan, Korea, 11-15 November 2007, pp. 722–735.

Bagga, Amit and Breck Baldwin (1998). "Algorithms for scoring coreference chains". In: *Proceedings of the 1st International Conference on Language Resources and Evaluation,* Granada, Spain, 28–30 May 1998, pp. 563–566.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). "Neural Machine Translation by Jointly Learning to Align and Translate". In: arXiv: `1409.0473v7 [cs.CL]`. URL: `http://arxiv.org/abs/1409.0473v7`.

Banarescu, Laura, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider (2013). "Abstract Meaning Representation for Sembanking". In: *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 178–186.

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Jauvin (2003). "A neural probabilistic language model". In: *Journal of machine learning research* 3.Feb, pp. 1137–1155.

Bergstra, James S, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl (2011). "Algorithms for hyper-parameter optimization". In: *Advances in Neural Information Processing Systems*, pp. 2546–2554.

Bizer, Christian, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann (2009). “DBPedia – A crystallization point for the Web of data”. In: *Journal of Web Semantics* 7, pp. 154–165.

Blei, David M, Andrew Y Ng, and Michael I Jordan (2003). “Latent dirichlet allocation”. In: *Journal of machine Learning research* 3.Jan, pp. 993–1022.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov (2017). “Enriching Word Vectors with Subword Information”. In: *Transactions of the Association for Computational Linguistics* 5, pp. 135–146. ISSN: 2307-387X. URL: `https://transacl.org/ojs/index.php/tacl/article/view/999`.

Bollacker, Kurt, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor (2008). “Freebase: A collaboratively created graph database for structuring human knowledge”. In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data,* Vancouver, B.C., Canada, 10–12 June 2008, pp. 1247–1250.

Botha, Jan and Phil Blunsom (2014). “Compositional Morphology for Word Representations and Language Modelling”. In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, pp. 1899–1907. URL: `http://proceedings.mlr.press/v32/botha14.pdf`.

Bullinaria, John A and Joseph P Levy (2007). “Extracting semantic representations from word co-occurrence statistics: A computational study”. In: *Behavior research methods* 39.3, pp. 510–526.

Bunescu, Razvan and Marius Paşca (2006). “Using encyclopedic knowledge for named entity disambiguation”. In: *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics,* Trento, Italy, 3–7 April 2006, pp. 9–16.

Cai, Jie and Michael Strube (2010). “Evaluation metrics for end-to-end coreference resolution systems”. In: *Proceedings of the SIGdial 2010 Conference: The 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue,* Tokyo, Japan, 24–25 September 2010, pp. 28–36. URL: `http://aclweb.org/anthology/W10-4305.pdf`.

Cano, Amparo E., Giuseppe Rizzo, Andrea Varga, Matthew Rowe, Milan Stankovic, and Aba-Sah Dadzie (2014). “Making sense of microposts named entity extraction & linking challenge”. In: *Proceedings of the 4th Workshop on Making Sense of Microposts,* Seoul, Korea, 7 April 2014, pp. 54–60.

Carmel, David, Ming-Wei Chang, Evgeniy Gabrilovich, Bo-June Paul Hsu, and Kuansan Wang (2014). “ERD’14: Entity recognition and disambiguation challenge”. In: *ACM SIGIR Forum*. Vol. 48. ACM, pp. 63–77.

Chen, Danqi and Christopher D. Manning (2014). “A Fast and Accurate Dependency Parser using Neural Networks”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 740–750. DOI: 10.3115/v1/D14-1082.

Chiu, Jason and Eric Nichols (2016). “Named Entity Recognition with Bidirectional LSTM-CNNs”. In: *Transactions of the Association for Computational Linguistics* 4, pp. 357–370. ISSN: 2307-387X. URL: https://transacl.org/ojs/index.php/tacl/article/view/792.

Choi, Eunsol, Omer Levy, Yejin Choi, and Luke Zettlemoyer (2018). “Ultra-Fine Entity Typing”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 87–96.

Clark, Kevin and Christopher D. Manning (2016a). “Deep reinforcement learning for mention-ranking coreference models”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing,* Austin, Tex., 1–5 November 2016, pp. 2256–2262.

— (2016b). “Improving coreference resolution by learning entity-level distributed representations”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Berlin, Germany, 7–12 August 2016.

Cotterell, Ryan and Hinrich Schütze (2015). “Morphological Word-Embeddings”. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, pp. 1287–1292. URL: http://www.aclweb.org/anthology/N15-1140.

Creutz, Mathias and Krista Lagus (2002). “Unsupervised Discovery of Morphemes”. In: *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*. Association for Computational Linguistics, pp. 21–30. DOI: 10.3115/1118647.1118650. URL: http://www.aclweb.org/anthology/W02-0603.

Csomai, Andras and Rada Mihalcea (2008). “Linking documents to encyclopedic knowledge”. In: *IEEE Intelligent Systems* 23.5, pp. 34–41.

Cucerzan, Silviu (2007). “Large-scale named entity disambiguation based on Wikipedia data”. In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Language Learning,* Prague, Czech Republic, 28–30 June 2007, pp. 708–716.

Dagan, Ido and Alon Itai (1990). “Automatic processing of large corpora for the resolution of anaphora references”. In: *Proceedings of the 13th International Conference on Computational Linguistics,* Helsinki, Finland, 20–25 August 1990. Vol. 3, pp. 330–332.

Dai, Hongliang, Siliang Tang, Fei Wu, Zewu Ma, and Yueting Zhuang (2015). “The ZJU-EDL System for Entity Discovery and Linking at TAC KBP 2015”. In: *Proceedings of the Eighth Text Analysis Conference*. Gaithersburg, MD, USA: National Institute of Standards and Technology.

Daiber, Joachim, Max Jakob, Chris Hokamp, and Pablo N. Mendes (2013). “Improving Efficiency and Accuracy in Multilingual Entity Extraction”. In: *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*. Graz, Austria, pp. 121–124.

Del Corro, Luciano, Abdalghani Abujabal, Rainer Gemulla, and Gerhard Weikum (2015). “FINET: Context-Aware Fine-Grained Named Entity Typing”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 868–878. DOI: `10.18653/v1/D15-1103`.

Doddington, George, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel (2004). “The Automatic Content Extraction (ACE) Program: Tasks, Data, and Evaluation”. In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC’04)*. Lisbon, Portugal: European Language Resources Association (ELRA).

Domingos, Pedro and Daniel Lowd (2009). *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan Claypool Publishers.

Durrett, Greg and Dan Klein (2013). “Easy victories and uphill battles in coreference resolution”. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing,* Seattle, Wash., 18–21 October 2013, pp. 1971–1982.

— (2014). “A Joint Model for Entity Analysis: Coreference, Typing, and Linking”. In: *Transactions of the Association for Computational Linguistics* 2, pp. 477–490. URL: `http://www.aclweb.org/anthology/Q14-1037`.

Eckart de Castilho, Richard and Iryna Gurevych (2014). “A broad-coverage collection of portable NLP components for building shareable analysis pipelines”. In: *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT*. Dublin, Ireland: Association for Computational Linguistics and Dublin City University, pp. 1–11. URL: `http://www.aclweb.org/anthology/W14-5201`.

Elman, Jeffrey L (1990). “Finding structure in time”. In: *Cognitive science* 14.2, pp. 179–211.

Erk, Katrin (2007). “A simple, similarity-based model for selectional preferences”. In: *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics,* Prague, Czech Republic, 23–30 June 2007, pp. 216–223.

Erk, Katrin and Sebastian Padó (2008). “A Structured Vector Space Model for Word Meaning in Context”. In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii, pp. 897–906. URL: `http://aclweb.org/anthology/D08-1094`.

Fahrni, Angela, Benjamin Heinzerling, Thierry Göckel, and Michael Strube (2014). “HITS' monolingual and cross-lingual entity linking system at TAC 2013”. In: *Proceedings of the Text Analysis Conference,* National Institute of Standards and Technology, Gaithersburg, Maryland, USA, 18–19 November 2013. URL: `http://www.nist.gov/tac/publications/2013/participant.papers/HITS.TAC2013.proceedings.pdf`.

Fahrni, Angela and Michael Strube (2012). “Jointly disambiguating and clustering concepts and entities with Markov logic”. In: *Proceedings of the 24th International Conference on Computational Linguistics,* Mumbai, India, 8–15 December 2012, pp. 815–832. URL: `http://www.aclweb.org/anthology/C12-1050.pdf`.

Ferrucci, David A. and Adam Lally (2004). “UIMA: An architectural approach to unstructured information processing in the corporate research environment”. In: *Natural Language Engineering* 10.3, pp. 327–348.

Fillmore, Charles J., Christopher R. Johnson, and Miriam R. L. Petruck (2003). “Background to FrameNet”. In: *International Journal of Lexicography* 16.3, pp. 235–250.

Francis-Landau, Matthew, Greg Durrett, and Dan Klein (2016). “Capturing Semantic Similarity for Entity Linking with Convolutional Neural Networks”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pp. 1256–1261. URL: `http://www.aclweb.org/anthology/N16-1150`.

Gabrilovich, Evgeniy and Shaul Markovitch (2007). “Computing semantic relatedness using Wikipedia-based explicit semantic analysis”. In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence,* Hyderabad, India, 6–12 January 2007, pp. 1606–1611.

Gage, Philip (1994). “A new algorithm for data compression”. In: *The C Users Journal* 12.2, pp. 23–38.

Giles, Jim (2005). “Internet encyclopaedias go head to head.” In: *Nature* 438.7070, pp. 900–901.

Gillick, Dan, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh (2014). "Context-Dependent Fine-Grained Entity Type Tagging". In: *ArXiv e-prints*. arXiv: 1412.1820 [cs.CL].

Globerson, Amir, Nevena Lazic, Soumen Chakrabarti, Amarnag Subramanya, Michael Ringaard, and Fernando Pereira (2016). "Collective Entity Resolution with Multi-Focal Attention". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 621–631. URL: http://www.aclweb.org/anthology/P16-1059.

Godin, Fréderic, Tom De Nies, Christian Beecks, Laurens De Vocht, Wesley De Neve, Erik Mannens, Thomas Seidl, and Rik Van de Walle (2014). "The normalized freebase distance". In: *The Semantic Web: ESWC 2014 Satellite Events*. Anissaras, Crete, Greece: Springer, pp. 218–221.

Goodfellow, Ian, Yoshua Bengio, Aaron Courville, and Yoshua Bengio (2016). *Deep learning*. Vol. 1. MIT press Cambridge.

Gupta, Abhijeet, Gemma Boleda, and Sebastian Pado (2018). "Instantiation". In: arXiv: 1808.01662v1 [cs.CL]. URL: http://arxiv.org/abs/1808.01662v1.

Gurevych, Iryna, Max Mühlhäuser, Christof Müller, Jürgen Steimle, Markus Weimer, and Torsten Zesch (2007). "Darmstadt knowledge processing repository based on UIMA". In: *Proceedings of the First Workshop on Unstructured Information Management Architecture at Biannual Conference of the Society for Computational Linguistics and Language Technology, Tübingen, Germany*, p. 89.

Hachey, Ben, Joel Nothman, and Will Radford (2014). "Cheap and easy entity evaluation". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers),* Baltimore, Md., 22–27 June 2014, pp. 464–469.

Hachey, Ben, Will Radford, Joel Nothman, Matthew Honnibal, and James R Curran (2013). "Evaluating entity linking with Wikipedia". In: *Artificial intelligence* 194, pp. 130–150.

Halliday, M. A. K. and Ruqaiya Hasan (1976). *Cohesion in English*. London, U.K.: Longman.

Han, Xianpei, Le Sun, and Jun Zhao (2011). "Collective entity linking in web text: A graph-based method". In: *Proceedings of the 34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* Beijing, China, 25–29 July 2011, pp. 765–774.

Heigold, Georg, Guenter Neumann, and Josef van Genabith (2017). "An Extensive Empirical Evaluation of Character-Based Morphological Tagging for 14 Languages". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain: Association for Computational Linguistics, pp. 505–513.

Heinzerling, Benjamin (2019a). *BPEmb: Pre-trained Subword Embeddings in 275 Languages (LREC 2018)*. DOI: `10.11588/data/V9CXPR`. URL: `https://doi.org/10.11588/data/V9CXPR`.

— (2019b). *Selectional Preference Embeddings (EMNLP 2017)*. DOI: `10.11588/data/FJQ4XL`. URL: `https://doi.org/10.11588/data/FJQ4XL`.

— (2019c). *Source Code, Data and Additional Material for the Thesis: "Aspects of Coherence for Entity Analysis"*. DOI: `10.11588/data/9JKAVW`. URL: `https://doi.org/10.11588/data/9JKAVW`.

Heinzerling, Benjamin, Alex Judea, and Michael Strube (2015). "HITS at TAC KBP 2015: Entity discovery and linking, and event nugget detection." In: *Proceedings of the Text Analysis Conference*. Gaithersburg, Maryland, USA: National Institute of Standards and Technology. URL: `https://tac.nist.gov/publications/2015/participant.papers/TAC2015.HITS.proceedings.pdf`.

Heinzerling, Benjamin, Nafise Sadat Moosavi, and Michael Strube (2017). "Revisiting Selectional Preferences for Coreference Resolution". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 1343–1350. URL: `http://www.aclweb.org/anthology/D17-1139`.

Heinzerling, Benjamin and Michael Strube (2015). "Visual Error Analysis for Entity Linking". In: *Proceedings of ACL-IJCNLP 2015 System Demonstrations*. Beijing, China: Association for Computational Linguistics and The Asian Federation of Natural Language Processing, pp. 37–42. URL: `http://www.aclweb.org/anthology/P15-4007`.

— (2018). "BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Ed. by Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga. Miyazaki, Japan: European Language Resources Association (ELRA).

Heinzerling, Benjamin, Michael Strube, and Chin-Yew Lin (2017). "Trust, but Verify! Better Entity Linking through Automatic Verification". In: *Proceedings of the*

15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. Valencia, Spain: Association for Computational Linguistics, pp. 828–838. URL: https://www.aclweb.org/anthology/E17-1078.

Hobbs, Jerry R. (1978). "Resolving pronominal references". In: *Lingua* 44, pp. 311–338.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.

Hoffart, Johannes, Yasemin Altun, and Gerhard Weikum (2014). "Discovering Emerging Entities with Ambiguous Names". In: *Proceedings of the 23rd International Conference on World Wide Web*. WWW '14. Seoul, Korea: ACM, pp. 385–396. DOI: 10.1145/2566486.2568003. URL: http://doi.acm.org/10.1145/2566486.2568003.

Hoffart, Johannes, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum (2012). "KORE: Keyphrase overlap relatedness for entity disambiguation". In: *Proceedings of the ACM 21st Conference on Information and Knowledge Management,* Maui, Hawaii, USA, 29 October – 2 November 2010, pp. 545–554.

Hoffart, Johannes, Fabian M. Suchanek, Klaus Berberich, Edwin Lewis-Kelham, Gerard de Melo, and Gerhard Weikum (2011). "YAGO2: Exploring and querying world knowledge in time, space, context, and many languages". In: *Proceedings of the 20th World Wide Web Conference,* Hyderabad, India, 28 March – 1 April, 2011, pp. 229–232.

Hoffart, Johannes, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum (2011). "Robust disambiguation of named entities in text". In: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing,* Edinburgh, Scotland, U.K., 27–29 July 2011, pp. 782–792.

Ji, Heng, Joel Nothman, and Ben Hachey (2014). "Overview of TAC-KBP2014 entity discovery and linking tasks". In: *Proceedings of the Text Analysis Conference,* National Institute of Standards and Technology, Gaithersburg, Maryland, USA, 17–18 November 2014.

Ji, Heng, Joel Nothman, Ben Hachey, and Radu Florian (2015). "Overview of TAC-KBP 2015 tri-lingual entity discovery and linking". In: *Proceedings of the Text Analysis Conference,* National Institute of Standards and Technology, Gaithersburg, Maryland, USA, 16–17 November 2015.

John, George H., Ron Kohavi, and Karl Pfleger (1994). "Irrelevant features and the subset selection problem". In: *Proceedings of the 11th International Conference on Machine Learning*, pp. 121–129.

Kann, Katharina and Hinrich Schütze (2016). "MED: The LMU System for the SIGMORPHON 2016 Shared Task on Morphological Reinflection". In: *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Berlin, Germany: Association for Computational Linguistics, pp. 62–70. DOI: `10.18653/v1/W16-2010`.

Kehler, Andrew, Douglas Appelt, Lara Taylor, and Aleksandr Simma (2004). "The (non)utility of predicate-argument frequencies for pronoun interpretation". In: *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics,* Boston, Mass., 2–7 May 2004, pp. 289–296.

Kim, Jin-Dong, Tomoko Ohta, Yuka Tateisi, and Junichi Tsujii (2003). "GENIA corpus—a semantically annotated corpus for bio-textmining". In: *Bioinformatics* 19.suppl 1, pp. i180–182.

Kim, Yoon, Yacine Jernite, David Sontag, and Alexander M Rush (2016). "Character-Aware Neural Language Models." In: *Proceedings of the 2016 Conference on Artificial Intelligence (AAAI)*, pp. 2741–2749.

Kripke, Saul A (1972). "Naming and necessity". In: *Semantics of natural language*. Springer, pp. 253–355.

Kruskal, Joseph B. (1956). "On the shortest spanning subtree of a graph and the traveling salesman problem". In: *Proceedings of the American Mathematical society* 7.1, pp. 48–50.

Kuhn, Harold W. (1955). "The Hungarian method for the assignment problem". In: *Naval Research Logistics Quarterly* 2.1-2, pp. 83–97.

Kulkarni, Sayali, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti (2009). "Collective annotation of Wikipedia entities in web text". In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* Paris, France, 28 June – 1 July 2009, pp. 457–466.

Lample, Guillaume, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer (2016). "Neural Architectures for Named Entity Recognition". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pp. 260–270. DOI: `10.18653/v1/N16-1030`.

Landauer, T. K. and S. T. Dumais (1997). "A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge". In: *Psychological Review* 104, pp. 211–240.

Lazaridou, Angeliki, Marco Marelli, Roberto Zamparelli, and Marco Baroni (2013). "Compositional-ly Derived Representations of Morphologically Complex Words in Distributional Semantics". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 1517–1526.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning". In: *nature* 521.7553, p. 436.

LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Lee, Heeyoung, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky (2013). "Deterministic Coreference Resolution Based on Entity-Centric, Precision-Ranked Rules". In: *Computational Linguistics* 39.4. DOI: 10.1162/COLI_a_00152.

Lee, Heeyoung, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky (2011). "Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task". In: *Proceedings of the Shared Task of the 15th Conference on Computational Natural Language Learning,* Portland, Oreg., 23–24 June 2011, pp. 28–34.

Lee, Jason, Kyunghyun Cho, and Thomas Hofmann (2017). "Fully Character-Level Neural Machine Translation without Explicit Segmentation". In: *Transactions of the Association for Computational Linguistics* 5, pp. 365–378.

Levy, Omer and Yoav Goldberg (2014). "Dependency-Based Word Embeddings". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland, pp. 302–308. DOI: 10.3115/v1/P14-2050. URL: http://aclweb.org/anthology/P14-2050.

Ling, Wang, Chris Dyer, Alan W Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis (2015). "Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 1520–1530. DOI: 10.18653/v1/D15-1176.

Ling, Xiao and Daniel S. Weld (2012). "Fine-Grained Entity Recognition". In: *Proceedings of the 26th Conference on the Advancement of Artificial Intelligence,* Toronto, Ontario, Canada, 22–26 July 2012, pp. 94–100.

Lita, Lucian Vlad, Abe Ittycheriah, Salim Roukos, and Nanda Kambhatla (2003). "Truecasing". In: *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics,* Sapporo, Japan, 7–12 July 2003. Association for Computational Linguistics, pp. 152–159.

Luo, Gang, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie (2015). "Joint Entity Recognition and Disambiguation". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 879–888. DOI: 10.18653/v1/D15-1104.

Luo, Xiaoqiang (2005). "On coreference resolution performance metrics". In: *Proceedings of the Human Language Technology Conference and the 2005 Conference on Empirical Methods in Natural Language Processing,* Vancouver, B.C., Canada, 6–8 October 2005, pp. 25–32.

Luong, Minh-Thang and Christopher D. Manning (2016). "Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1054–1063. URL: http://www.aclweb.org/anthology/P16-1100.

Luong, Minh-Thang, Richard Socher, and Christopher D. Manning (2013). "Better Word Representations with Recursive Neural Networks for Morphology". In: *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 104–113.

Ma, Xuezhe and Eduard Hovy (2016). "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1064–1074. DOI: 10.18653/v1/P16-1101.

Ma, Yukun, Erik Cambria, and Sa Gao (2016). "Label Embedding for Zero-shot Fine-grained Named Entity Typing". In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, pp. 171–180.

Maaten, Laurens van der and Geoffrey Hinton (2008). "Visualizing data using t-SNE". In: *Journal of machine learning research* 9.Nov, pp. 2579–2605.

Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky (2014). "The Stanford CoreNLP Natural Language Processing Toolkit". In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Baltimore, Maryland: Association for Computational Linguistics, pp. 55–60. DOI: 10.3115/v1/P14-5010. URL: http://aclweb.org/anthology/P14-5010.

Marciniak, Tomacz and Michael Strube (2005). "Beyond the pipeline: Discrete optimization in NLP". In: *Proceedings of the 9th Conference on Computational Natural Language Learning,* Ann Arbor, Mich., USA, 29–30 June 2005, pp. 136–145. URL: http://www.aclweb.org/anthology/W05-0618.pdf.

Martschat, Sebastian and Michael Strube (2014). "Recall error analysis for coreference resolution". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing,* Doha, Qatar, 25–29 October 2014, pp. 2070–2081. URL: http://www.aclweb.org/anthology/D14-1221.pdf.

— (2015). "Latent structures for coreference resolution". In: *Transactions of the Association for Computational Linguistics* 3, pp. 405–418. URL: http://www.aclweb.org/anthology/Q15-1029.pdf.

Matthews, Brian W (1975). "Comparison of the predicted and observed secondary structure of T4 phage lysozyme". In: *Biochimica et Biophysica Acta (BBA)-Protein Structure* 405.2, pp. 442–451.

McCallum, Andrew and Wei Li (2003). "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons". In: *Proceedings of the 7th Conference on Computational Natural Language Learning,* Edmonton, Alberta, Canada, 31 May – 1 June 2003, pp. 188–191.

McInnes, Leland, John Healy, Nathaniel Saul, and Lukas Grossberger (2018). "UMAP: Uniform Manifold Approximation and Projection". In: *The Journal of Open Source Software* 3.29, p. 861.

McNamee, Paul and Hoa Trang Dang (2009). "Overview of the TAC 2009 knowledge base population track". In: *Proceedings of the Text Analysis Conference,* National Institute of Standards and Technology, Gaithersburg, Maryland, USA, 16–17 November 2009. Vol. 17. National Institute of Standards and Technology (NIST) Gaithersburg, Maryland, USA, pp. 111–113.

Mihalcea, Rada and Andras Csomai (2007). "Linking Documents to Encyclopedic Knowledge". In: *Proceedings of the ACM 16th Conference on Information and Knowledge Management,* Lisbon, Portugal, 6–9 November 2007, pp. 233–242.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013). "Efficient estimation of word representations in vector space". In: *Proceedings of the ICLR 2013 Workshop Track.*

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean (2013). "Distributed representations of words and phrases and their compositionality". In: *Proceedings of Advances in Neural Information Processing Systems 26.* Lake Tahoe, Nev., 5–8 December 2013, pp. 3111–3119.

Miller, Tristan, Nicolai Erbs, Hans-Peter Zorn, Torsten Zesch, and Iryna Gurevych (2013). "DKPro WSD: A Generalized UIMA-based Framework for Word Sense Disambiguation". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 37–42. URL: `http://www.aclweb.org/anthology/P13-4007`.

Milne, David and Ian H. Witten (2008a). "An effective, low-cost measure of semantic relatedness obtained from Wikipedia links". In: *Proceedings of the Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy at AAAI-08,* Chicago, Ill., 13 July 2008, pp. 25–30.

— (2008b). "Learning to link with Wikipedia". In: *Proceedings of the ACM 17th Conference on Information and Knowledge Management,* Napa Valley, Cal., USA, 26–30 October 2008, pp. 1046–1055.

Miyamoto, Yasumasa and Kyunghyun Cho (2016). "Gated Word-Character Recurrent Language Model". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 1992–1997. DOI: `10.18653/v1/D16-1209`.

Moosavi, Nafise Sadat and Michael Strube (2016). "Which coreference evaluation metric do you trust? A proposal for a link-based entity aware metric". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Berlin, Germany, 7–12 August 2016, pp. 632–642. URL: `http://www.aclweb.org/anthology/P16-1060.pdf`.

Moro, Andrea, Alessandro Raganato, and Roberto Navigli (2014). "Entity linking meets word sense disambiguation: A unified approach". In: *Transactions of the Association for Computational Linguistics* 2, pp. 231–244.

Munkres, James (1957). "Algorithms for the assignment and transportation problems". In: *Journal of the Society for Industrial & Applied Mathematics* 5.1, pp. 32–38.

Nakashole, Ndapandula, Tomasz Tylenda, and Gerhard Weikum (2013). "Fine-grained Semantic Typing of Emerging Entities". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 1488–1497.

Noreen, Eric W. (1989). *Computer Intensive Methods for Hypothesis Testing: An Introduction*. New York, N.Y.: Wiley.

Ogden, Charles Kay and Ivor Armstrong Richards (1923). *The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism*. London, U.K.: Kegan Paul.

Pantel, Patrick, Rahul Bhagat, Bonaventura Coppola, Timothy Chklovski, and Eduard Hovy (2007). "ISP: Learning Inferential Selectional Preferences". In: *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*. Rochester, New York, pp. 564–571. URL: `http://aclweb.org/anthology/N07-1071`.

Parker, Robert, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda (2011). *English Gigaword Fifth Edition*. LDC2011T07.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Pellissier Tanon, Thomas, Denny Vrandečić, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher (2016). "From freebase to wikidata: The great migration". In: *Proceedings of the 25th international conference on world wide web*. International World Wide Web Conferences Steering Committee, pp. 1419–1428.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning (2014). "Glove: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. URL: `http://www.aclweb.org/anthology/D14-1162`.

Pershina, Maria, Yifan He, and Ralph Grishman (2015). "Personalized Page Rank for Named Entity Disambiguation". In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, pp. 238–243. URL: `http://www.aclweb.org/anthology/N15-1026`.

Ponzetto, Simone Paolo and Michael Strube (2006a). "Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution". In: *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics,* New York, N.Y., 4–9 June 2006, pp. 192–199. URL: `http://www.aclweb.org/anthology/N06-1025.pdf`.

— (2006b). "Semantic role labeling for coreference resolution". In: *Companion Volume to the Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics,* Trento, Italy, 3–7 April 2006, pp. 143–146. URL: `http://www.aclweb.org/anthology/E06-2015.pdf`.

Pradhan, Sameer, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang (2012). "CoNLL-2012 Shared Task: Modeling multilingual unrestricted coreference in OntoNotes". In: *Proceedings of the Shared Task of the 16th Conference on Computational Natural Language Learning,* Jeju Island, Korea, 12–14 July 2012, pp. 1–40.

Qiu, Siyu, Qing Cui, Jiang Bian, Bin Gao, and Tie-Yan Liu (2014). "Co-learning of Word Representations and Morpheme Representations". In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, pp. 141–150.

Rabinovich, Maxim and Dan Klein (2017). "Fine-Grained Entity Typing with High-Multiplicity Assignments". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 330–334. DOI: `10.18653/v1/P17-2052`.

Raghunathan, Karthik, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher D. Manning (2010). "A multi-pass sieve for coreference resolution". In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing,* Cambridge, Mass., 9–11 October 2010, pp. 492–501.

Rahman, Altaf and Vincent Ng (2011). "Coreference resolution with world knowledge". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Portland, Oreg., 19–24 June 2011, pp. 814–824.

Rajani, Nazneen Fatema and Raymond Mooney (2016). "Combining Supervised and Unsupervised Ensembles for Knowledge Base Population". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 1943–1948. URL: `https://aclweb.org/anthology/D16-1201`.

Ratinov, Lev, Dan Roth, Doug Downey, and Mike Anderson (2011). "Local and global algorithms for disambiguation to Wikipedia". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Portland, Oreg., 19–24 June 2011, pp. 1375–1384.

Rau, Lisa F (1991). "Extracting company names from text". In: *Artificial Intelligence Applications, 1991. Proceedings., Seventh IEEE Conference on*. Vol. 1. IEEE, pp. 29–32.

Rei, Marek, Gamal Crichton, and Sampo Pyysalo (2016). "Attending to Characters in Neural Sequence Labeling Models". In: *Proceedings of COLING 2016, the*

26th International Conference on Computational Linguistics: Technical Papers. Osaka, Japan: The COLING 2016 Organizing Committee, pp. 309–318.

Reimers, Nils and Iryna Gurevych (2017). "Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging". In: *CoRR* abs/1707.09861. URL: `http://arxiv.org/abs/1707.09861`.

Ren, Xiang, Wenqi He, Meng Qu, Lifu Huang, Heng Ji, and Jiawei Han (2016). "AFET: Automatic Fine-Grained Entity Typing by Hierarchical Partial-Label Embedding". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 1369–1378. DOI: `10.18653/v1/D16-1144`.

Ren, Xiang, Wenqi He, Meng Qu, Clare R. Voss, Heng Ji, and Jiawei Han (2016). "Label Noise Reduction in Entity Typing by Heterogeneous Partial-Label Embedding". In: *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: ACM, pp. 1825–1834. DOI: `10.1145/2939672.2939822`. URL: `http://doi.acm.org/10.1145/2939672.2939822`.

Resnik, Philip (1993). "Selection and Information: A Class-based Approach to Lexical Relationships". PhD thesis. Department of Computer and Information Science, University of Pennsylvania, Philadelphia, Penn.

Ritter, Alan, Mausam, and Oren Etzioni (2010). "A latent dirichlet allocation method for selectional preferences". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden, pp. 424–434. URL: `http://aclweb.org/anthology/P10-1045`.

Roth, Dan and Wen-tau Yih (2004). "A linear programming formulation for global inference in natural language tasks". In: *Proceedings of the 8th Conference on Computational Natural Language Learning,* Boston, Mass., USA, 6–7 May 2004, pp. 1–8.

Sang, Erik F. Tjong Kim (2002). "Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition". In: *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002).*

Santos, Cicero Dos and Bianca Zadrozny (2014). "Learning Character-level Representations for Part-of-Speech Tagging". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, pp. 1818–1826. URL: `http://proceedings.mlr.press/v32/santos14.pdf`.

Schmidhuber, Jürgen (2015). "Deep learning in neural networks: An overview". In: *Neural networks* 61, pp. 85–117.

Schrijver, Alexander (1998). *Theory of linear and integer programming*. John Wiley & Sons.

Schuster, Sebastian and Christopher D. Manning (2016). "Enhanced English Universal Dependencies: An Improved Representation for Natural Language Understanding Tasks". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Portorož, Slovenia.

Schütze, Hinrich (2017). "Nonsymbolic Text Representation". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain: Association for Computational Linguistics, pp. 785–796. URL: http://www.aclweb.org/anthology/E17-1074.

Sennrich, Rico, Barry Haddow, and Alexandra Birch (2016). "Neural Machine Translation of Rare Words with Subword Units". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1715–1725. URL: http://www.aclweb.org/anthology/P16-1162.

Shimaoka, Sonse, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel (2017). "Neural Architectures for Fine-grained Entity Type Classification". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain: Association for Computational Linguistics, pp. 1271–1280. URL: http://www.aclweb.org/anthology/E17-1119.

Sil, Avirup, Giorgiana Dinu, and Radu Florian (2015). "The IBM Systems for Trilingual Entity Discovery and Linking at TAC 2015". In: *Proceedings of the Eighth Text Analysis Conference*. Gaithersburg, MD, USA: National Institute of Standards and Technology.

Sil, Avirup and Alexander Yates (2013). "Re-ranking for joint named-entity recognition and linking". In: *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM. Orlando, Florida, USA, pp. 2369–2374.

Silveira, Natalia G (2016). "Designing Syntactic Representations for NLP: an Empirical Investigation". PhD thesis. Stanford University.

Singh, Sameer, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum (2012). *Wikilinks: A Large-scale Cross-Document Coreference Corpus Labeled via Links to Wikipedia*. Tech. rep. UM-CS-2012-015. University of Massachusetts, Amherst.

Spärck Jones, Karen (1972). "A statistical interpretation of term specificity and its application in retrieval". In: *Journal of Documentation* 28.1. Reprinted in *Journal of Documentation*, 60(5), pp.493-502 (2004), pp. 11–20.

Speer, Robert, Joshua Chin, Andrew Lin, Sara Jewett, and Lance Nathan (2017). *LuminosoInsight/wordfreq: v1.7*. DOI: `10.5281/zenodo.998161`. URL: `https://doi.org/10.5281/zenodo.998161`.

Sperr, Henning, Jan Niehues, and Alex Waibel (2013). "Letter N-Gram-based Input Encoding for Continuous Space Language Models". In: *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 30–39.

Spitkovsky, Valentin I and Angel X. Chang (2012). "A cross-lingual dictionary for English Wikipedia concepts". In: *Proceedings of the 8th International Conference on Language Resources and Evaluation,* Istanbul, Turkey, 21–27 May 2012, pp. 3168–3175.

Strötgen, Jannik and Michael Gertz (2010). "HeidelTime: High Quality Rule-Based Extraction and Normalization of Temporal Expressions". In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden: Association for Computational Linguistics, pp. 321–324. URL: `http://www.aclweb.org/anthology/S10-1071`.

Strube, Michael (2015). "The (Non)Utility of Semantics for Coreference Resolution". Invited talk given at the EMNLP 2015 workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics (LSDSem 2015). URL: `http://www.coli.uni-saarland.de/%5C%7Emroth/LSDSem/2015/slides%5C_MichaelStrube.pdf`.

Strube, Michael and Simone Paolo Ponzetto (2006). "WikiRelate! Computing semantic relatedness using Wikipedia". In: *Proceedings of the 21st National Conference on Artificial Intelligence,* Boston, Mass., 16–20 July 2006, pp. 1419–1424. URL: `http://www.aaai.org/Papers/AAAI/2006/AAAI06-223.pdf`.

Suchanek, Fabian M., Gjergji Kasneci, and Gerhard Weikum (2007). "YAGO: A core of semantic knowledge unifying WordNet and Wikipedia". In: *Proceedings of the 16th World Wide Web Conference,* Banff, Canada, 8–12 May, 2007, pp. 697–706.

Sutton, Charles and Andrew McCallum (2007). "An introduction to conditional random fields for relational learning". In: *Introduction to Statistical Relational Learning*. Ed. by L. Getoor and B. Taskar. Cambridge, Mass.: MIT Press, pp. 93–128.

Tjong Kim Sang, Erik F. and Fien De Meulder (2003). "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*. CONLL '03. Edmonton, Canada: Association for Computational Linguistics, pp. 142–147. DOI: `10.3115/1119176.1119195`. URL: `https://doi.org/10.3115/1119176.1119195`.

Tsoumakas, Grigorios and Ioannis Katakis (2007). “Multi-label classification: An overview”. In: *International Journal of Data Warehousing and Mining (IJDWM)* 3.3, pp. 1–13.

Turian, Joseph, Lev-Arie Ratinov, and Yoshua Bengio (2010). “Word Representations: A Simple and General Method for Semi-Supervised Learning”. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics, pp. 384–394.

Van de Cruys, Tim (2014). “A Neural Network Approach to Selectional Preference Acquisition”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar, pp. 26–35. URL: `http://www.aclweb.org/anthology/D14-1004`.

Vania, Clara and Adam Lopez (2017). “From Characters to Words to in Between: Do We Capture Morphology?” In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 2016–2027. DOI: `10.18653/v1/P17-1184`.

Vilain, Marc, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman (1995). “A model-theoretic coreference scoring scheme”. In: *Proceedings of the 6th Message Understanding Conference (MUC-6)*. San Mateo, Cal.: Morgan Kaufmann, pp. 45–52.

Vrandečić, Denny and Markus Krötzsch (2014). “Wikidata: a free collaborative knowledgebase”. In: *Communications of the ACM* 57.10, pp. 78–85.

Vylomova, Ekaterina, Trevor Cohn, Xuanli He, and Gholamreza Haffari (2017). “Word Representation Models for Morphologically Rich Languages in Neural Machine Translation”. In: *Proceedings of the First Workshop on Subword and Character Level Models in NLP*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 103–108.

Wieting, John, Mohit Bansal, Kevin Gimpel, and Karen Livescu (2016). “Charagram: Embedding Words and Sentences via Character n-grams”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 1504–1515. DOI: `10.18653/v1/D16-1157`.

Wikipedia contributors (2018). *Wikipedia:Notability — Wikipedia, The Free Encyclopedia*. [Online; accessed 21-September-2018]. URL: `https://en.wikipedia.org/wiki/Wikipedia:Notability`.

Wiseman, Sam, Alexander M. Rush, and Stuart Shieber (2016). “Learning global features for coreference resolution”. In: *Proceedings of the 2016 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, Cal., 12–17 June 2016. To appear.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean (2016). "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation". In: arXiv: `1609.08144v2 [cs.CL]`. URL: `http://arxiv.org/abs/1609.08144v2`.

Yaghoobzadeh, Yadollah and Hinrich Schütze (2015). "Corpus-level Fine-grained Entity Typing Using Contextual Information". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 715–725. DOI: `10.18653/v1/D15-1083`.

— (2017). "Multi-level Representations for Fine-Grained Typing of Knowledge Base Entities". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain: Association for Computational Linguistics, pp. 578–589. URL: `http://www.aclweb.org/anthology/E17-1055`.

Yamada, Ikuya, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji (2016). "Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation". In: *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*. Berlin, Germany: Association for Computational Linguistics, pp. 250–259.

Yogatama, Dani, Dan Gillick, and Nevena Lazic (2015). "Embedding Methods for Fine Grained Entity Type Classification". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Beijing, China: Association for Computational Linguistics, pp. 291–296. DOI: `10.3115/v1/P15-2048`.

Yosef, Mohamed Amir, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum (2012). "HYENA: Hierarchical Type Classification for Entity Names". In: *Proceedings of COLING 2012: Posters*. Mumbai, India: The COLING 2012 Organizing Committee, pp. 1361–1370.

Zipf, George Kingsley (1946). “The psychology of language”. In: *Encyclopedia of psychology*. Philosophical Library, pp. 332–341.