CONCEPT AND ENTITY GROUNDING USING INDIRECT SUPERVISION

BY

CHEN-TSE TSAI

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Doctoral Committee:

Professor Dan Roth, Chair
Professor Kevin Chang
Professor ChengXiang Zhai
Professor Rada Mihalcea, University of Michigan

Abstract

Extracting and disambiguating entities and concepts is a crucial step toward understanding natural language text. In this thesis, we consider the problem of grounding concepts and entities mentioned in text to one or more knowledge bases (KBs). A well-studied scenario of this problem is the one in which documents are given in English and the goal is to identify concept and entity mentions, and find the corresponding entries the mentions refer to in Wikipedia. We extend this problem in two directions: First, we study identifying and grounding entities written in any language to the English Wikipedia. Second, we investigate using multiple KBs which do not contain rich textual and structural information Wikipedia does.

These more involved settings pose a few additional challenges beyond those addressed in the standard English Wikification problem. Key among them is that no supervision is available to facilitate training machine learning models. The first extension, cross-lingual Wikification, introduces problems such as recognizing multilingual named entities mentioned in text, translating non-English names into English, and computing word similarity across languages. Since it is impossible to acquire manually annotated examples for all languages, building models for all languages in Wikipedia requires exploring indirect or incidental supervision signals which already exist in Wikipedia. For the second setting, we need to deal with the fact that most KBs do not contain the rich information Wikipedia has; consequently, the main supervision signal used to train Wikification rankers does not exist anymore. In this thesis, we show that supervision signals can be obtained by carefully examining the redundancy and relations between multiple KBs. By developing algorithms and models which harvest these incidental signals, we can achieve better performance on these tasks.

Publication Notes

Parts of the work in this thesis have appeared in the following publications:

- Chen-Tse Tsai, Stephen Mayhew, and Dan Roth. Cross-Lingual Named Entity Recognition via Wikification In *Proceedings of the SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, 2016.

- Chen-Tse Tsai and Dan Roth. Cross-Lingual Wikification Using Multilingual Embeddings In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2016.

- Chen-Tse Tsai and Dan Roth. Concept Grounding to Multiple Knowledge Bases via Indirect Supervision In *Transaction of the Association for Computational Linguistics (TACL)*, 2016.

- Chen-Tse Tsai and Dan Roth. Illinois Cross-Lingual Wikifier: Grounding Entities in Many Languages to the English Wikipedia In *Proceedings of the International Conference on Computational Linguistics (COLING) (Demonstrations)*, 2016.

- Chen-Tse Tsai, Gourab Kundu and Dan Roth Concept-Based Analysis of Scientific Literature In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM)*, 2013

- Chen-Tse Tsai, Stephen Mayhew, Haoruo Peng, Mark Sammons, Bhargav Mangipundi, Pavankumar Reddy, and Dan Roth Illinois CCG Entity Discovery and Linking, Event Nugget Detection and Coreference, and Slot Filler Validation Systems for TAC 2016 In *Proceedings of Text Analysis Conference (TAC)*, 2016

- Mark Sammons, Haoruo Peng, Yangqiu Song, Shyam Upadhyay, Chen-Tse Tsai, Pavankumar Reddy, Subhro Roy, and Dan Roth Illinois CCG TAC 2015 Event Nugget, Entity Discovery and Linking, and Slot Filler Validation Systems In *Proceedings of Text Analysis Conference (TAC)*, 2015

- Mark Sammons, Yangqiu Song, Ruichen Wang, Gourab Kundu, Chen-Tse Tsai Shyam Upadhyay, Stephen Mayhew, Dan Roth, Siddarth Ancha Overview of UICCG Systems for Event Argument Extraction, Entity Discovery and Linking, and Slot Filler Validation In *Proceedings of Text Analysis Conference (TAC)*, 2014

To my family.

Acknowledgments

I would like to express my sincere gratitude to my advisor, Prof. Dan Roth, for all the supports in the last five years. I really enjoy solving challenging problems with him and I always admire his insightful ideas. He selects good project directions, gives us resources, removes any obstacle in the way, and more importantly, he gives us enough freedom to explore the problem according to our own interest. Besides critical thinking, I learned how to discuss, listen, debate, and present from him. He demonstrates what is a remarkable scholar and a passionate educator.

I would also like to thank my former advisor, Prof. Chih-Jen Lin, who equipped me with fundamental skills, strong interest, and right attitude of doing research. If it was not him, I would not decide to pursue a Ph.D abroad.

I'm fortunate to have Prof. ChenXiang Zhai, Prof. Kevin Chang, and Prof. Rada Mihalcea on my dissertation committee. The valuable comments and questions from them really help me to rethink and thus refine many aspects of the problem. In addition, I would like to thank my mentors during my summer internships, Dr. Scott Wen-tau Yih and Dr. Aria Haghighi, who broaden my horizon to many other problems in the field. Those summers are unforgettable memories.

I would like to thank everyone I encountered in the last five years. They shaped me and made my life in Champaign-Urbana more colorful. I especially acknowledge the members in CogComp group. In my first two years, Kai-Wei Chang, Gourab Kundu, Vivek Srikumar, and Rajhans Samdani helped me fitting into the group and understanding the research area more quickly. Mark Sammons and Eric Horn assisted me in many aspects such as software and traveling issues. Moreover, many thanks to other colleagues and collaborators, Stephen Mayhew, Shyam Upadhyay, Subhro Roy, Daniel Khashabi, Haoruo Peng, Nitish Gupta, Colin Graber, Shashank Gupta, Qiang Ning, Chase Duncan, Snigdha Chaturvedi, Christos Christodoupoulos, Bhargav Mangipudi, Hao Wu, Pavan Muddireddy, John Wieting, and Ching-Pei Lee. We had great discussions and I definitely have learned a lot from you.

Finally, I would like to thank my parents, my brother, and my wife, for their unconditional sacrifice and support. I will not be able to go this far without you.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In recent decades, almost everything happening in the world is documented in text format and can be accessed via the Internet. Moreover, billions of people record their daily life using social networks or web blogs. To automatically understand, organize, and further make this enormous amount of textual information easily accessible by people, natural language processing (NLP) techniques play indispensable roles.

1.1 Motivation

Understanding entities and concepts mentioned in text is a critical step toward understanding text for both human beings and computer programs. For example, the underlined named entity mentions (names of people, organizations, locations, etc.) in the following paragraph provides the most important information of this news piece; who is involved in this event and where does the event happen.

A man holding a gun on a French female soldier at Orly Airport shouted, ”I am here to die for Allah,” before two of the soldier’s comrades shot the attacker dead Saturday morning. None of the soldiers were wounded, Paris Prosecutor Francois Molins said. The suspect, identified by Molins as Ziyed Ben Belgacem also is accused of shooting a police officer earlier in the day when he stole the officer’s weapon.

In many cases, we simply need additional domain knowledge to understand some text. To understand the following paragraph from a scientific paper, for instance, if we are not familiar with this domain, we need to acquire knowledge about the highlighted biological concepts by looking up an encyclopedia or biological ontologies.

Mucolipidosis type IV (MLIV) is an autosomal recessive lysosomal storage disorder characterized by severe neurologic and ophthalmologic abnormalities. Recently the MLIV gene, MCOLN1, has been identified as a new member of the transient receptor potential (TRP) cation channel superfamily.

Furthermore, a lot of information around the world is written in non-English languages. According to a study in 2011[1], there are 172 languages that have more than 3 millions speakers, but Google Translate only supports 104 languages at the moment of writing this thesis (2017). Take the following paragraph from Tamil news as an example.

> சிஐஏ இயக்குநர் மைக் பாம்பேயோ நியமனத்துக்கு அமெரிக்க செனட் சபை ஒப்புதல். ஆனால், சிஐஏ முகமை மற்றும் அமெரிக்க அதிபர் டிரம்ப் இடையே ஒரு பயனுள்ள அலுவல் ரீதியான உறவினை உருவாக்குவதே மைக் பாம்பேயோவின் உடனடி பணியாக இருக்கும்.

When there is no robust machine translation technology, if we can reliably extract and disambiguate the named entities to an English encyclopedia (E.g., Wikipedia), English speakers would be able to reliably estimate the topic of this news story. For instance, if we know that the entities exist in the above paragraph are CIA, Donald Trump, United States, Mike Pompeo, and the United States Senate, we could understand that it describes a political event related to the Trump administration and the director of CIA, Mike Pompeo.

Understanding entities is not only crucial for human readers, but for many NLP applications. For example, in the question answering task, given a question expressed as a natural language sentence, the goal is to find an answer mentioned in a given document, a collection of documents, or a knowledge base. There are different types of questions, but in many cases, named entities are involved in the questions or answers. The following example question-answer pair is taken from the WebQuestions dataset (Berant et al., 2013) which contains questions collected from Google search suggestions.

Q: *Who first voiced Meg on Family Guy?*
A: *Lancy Chabert*

If we can ground the name "Meg" (a character in the television series, Family Guy) to its corresponding entry in Wikipedia, "`https://en.wikipedia.org/wiki/Meg_Griffin`", we can easily obtain a list of actors who voiced her, and the corresponding years when they voiced her. Also, since many entries in Wikipedia can refer to "Meg", knowing that "Family Guy" refers to the television series is important in grounding "Meg".

Machine Translation provides another application of the methods developed in this thesis. Simply identifying named entity mentions is shown to be very useful in machine translation, where the task is to translate text written in one language to another language. Since some names could show up very infrequently or never appear in the training data, the model may not be able to translate these names robustly. The sparsity of named entity mentions requires them to be treated differently in a machine translation system. Therefore, identifying phrases in text which are named entities is important.

[1] `http://www.conservapedia.com/List_of_languages_by_number_of_speakers`

In this dissertation, we investigate the problems of recognizing entities and concepts mentioned in text, and grounding these mentions to one or multiple knowledge bases. We not only consider well-studied English news documents, but address additional challenges from different genres of text such as scientific papers and from text written in many other languages.

1.2 Challenges

Identifying words which are part of names is non-trivial. This problem is usually referred as the mention extraction problem. For instance, in a sentence

Sunday's Super Bowl is between the Denver Broncos and Seattle Seahawks,

"Sunday's Super Bowl" could be recognized as a named entity instead of "Super Bowl" since "Sunday" is also capitalized. On the other hand, a system may only identify "Denver" instead of the full name "Denver Broncos" since Denver is a more well-known location name. Texts from social medias or discussion forums could be more challenging than the news articles since they often contain more nicknames and abbreviations, and do not have proper capitalization. Moreover, for the languages which do not use space to separate words, such as Chinese and Japanese, the problem of recognizing named entity mentions is much more difficult.

The most notable challenges in grounding entities and concepts is from the ambiguity and variability of languages:

- Ambiguity means that a string or phrase can be used to refer to multiple entities. For instance, "Chicago" could refer to the city, the baseball team, the band, or the computer font. Similarly, "Wednesday" could be a day in the week or an English football club, and "Apple" could be the company or the fruit.

- Variability means that we can use different names to refer to the same entity. For example, the city Chicago can also be referred to as "Windy City", and Obama has nicknames such as "Barry" and "Obomber".

Besides these well-known issues, we will introduce and address other challenges that we encounter in different applications in this thesis.

1.3 Indirect and Incidental Supervision

The standard machine learning methodology suggests to collect examples of the given task, and learn a model which generalizes from these training examples. The number of training examples usually plays a crucial role in the quality of the trained model. However, this *directly supervised* protocol does not work for many real world problems. On one hand, annotating data is very costly for complex enough tasks, since we need to find and train human annotators. On the other hand, new tasks are invented frequently. It is not feasible to annotate new training examples whenever the task is changed slightly, for example, adding a new named entity type or moving from news domain to social media domain. To overcome this supervision bottleneck, several works (Klementiev and Roth, 2006; Chang et al., 2008a; Clarke et al., 2010; Chang et al., 2010b; Tsai and Roth, 2016a) have proposed to use indirect or incidental supervision instead.

> *Indirect supervision refers to supervision extracted automatically from information that exists in the data or environment, and is independent of the task at hand. That is, this information is created for other purposes, not for solving the target task. Although these signals are not the exact output of the task, they are co-related with the target task and could be exploited in order to facilitate learning.*

Take the wikification problem (Chapter 3) as an example. The key component of the wikification pipeline is the ranking step which selects the best title from a set of title candidates based on contextual clues in the given document. The anchor texts in Wikipedia documents are commonly used to construct training examples for the step (Milne and Witten, 2008; Ratinov et al., 2011). Although these hyperlinked phrases are not created for this task (i.e., not all entity mentions are linked in Wikipedia documents, some linked mentions are common nouns or numbers, and there is no NIL mention), they can be used to provide useful supervision signals. In Chapter 5, we will show that when using multiple knowledge bases which do not have such hyperlink structure, we can generate training examples for the ranking model by exploring the redundancy and relationship between the KBs. For another example, the inter-language links in Wikipedia connect pages in different languages describe the same entity, which are created for the readers to conveniently navigate information across languages. By developing appropriate algorithms, these inter-language links not only provide supervision for name translation/transliteration (Chapter 4.2), but also can be used to compute cross-lingual word similarity (Chapter 4.3).

For more applications which leverage indirect supervisions, Roth (2017) provides a comprehensive survey on the categories of incidental supervision signals.

1.4 Thesis Overview

In this thesis, we investigate the problem of grounding entities and concepts in text to one or multiple knowledge bases. In particular, we make the following claims in this thesis:

By exploring existing information in a knowledge base or relationships between multiple knowledge bases, we can acquire indirect or incidental supervision signals which facilitate the training of machine learning models for identifying and grounding mentions in text. That is, good performance can be achieved without task-specific annotations even for more challenging problems such as multilingual name recognition, cross-lingual grounding, and using knowledge bases which do not contain the rich textual and structural information Wikipedia does.

The primary contributions of this thesis are summarized below:

1. We show that by grounding words to an encyclopedia, we can generate useful features for a named entity recognition model. More importantly, grounding words in any language to the English encyclopedia generates good language-independent features for cross-lingual named entity recognition, the task in which the model is trained on one language and tested on other languages. (Chapter 4.1)

2. We propose a probabilistic model to learn better name translation. This model is trained using indirect supervision from Wikipedia title pairs. Using the inter-language links in Wikipedia, we can obtain name translation examples for all languages in Wikipedia, and for any type of entities. The proposed model jointly considers word alignments and word transliteration, hence outperforms the traditional transliteration and machine translation models and, in particular, allows one to generate better title candidates in the cross-lingual wikification problem. (Chapter 4.2)

3. We propose a cross-lingual word and entity embeddings model which can be applied to all languages in Wikipedia and does not require other annotations or resources beyond the Wikipedia dump. We show that by jointly embedding words and Wikipedia titles in a pair of languages into the same continuous vector space, we can learn a better cross-lingual similarity metric between words and titles. In particular, this supports better disambiguation of foreign mentions in text to the English Wikipedia. (Chapter 4.3)

4. When using multiple knowledge bases other than Wikipedia, we show that we can obtain reliable indirect supervision signals for training the ranking model by carefully examining the redundancy and relationship among the knowledge bases. These indirect supervision signals facilitate the training of well-studied supervised statistical learning models, resulting in state of the art performance on these tasks. (Chapter 5)

Chapter 2

Background

This chapter consists of two parts. In the first section, we introduce the NLP tasks, along with the common techniques used for solving them and the evaluation methodologies, which are closely related to the problems studied in this thesis. The second section briefly describes the machine learning techniques and protocols that we apply in this thesis. Note that we separate the Wikification (Entity Linking) problem, and provide a more comprehensive survey in Chapter 3.

2.1 Related Tasks

In this section, we introduce the definition and common formulation of word sense disambiguation (WSD), named entity recognition (NER), and wikification/entity linking problems.

2.1.1 Word Sense Disambiguation

Human language is ambiguous: words can have more than one distinct meaning. For instance, consider the following sentences:

I can't hear the bass guitar.

I like to go bass fishing.

The word *bass* clearly has different meanings in the two sentences: low-frequency tones in the first sentence and a type of fish in the second sentence. The correct sense of an ambiguous word depends on the context in which it occurs. The problem of WSD is the task of automatically assigning the correct meaning to a polysemous word within a given context. WSD is a historical task in the field of NLP. It was convinced as an essential task for machine translation in the late 1940s (Weaver, 1955). Ide and Véronis (1998) present more in-depth history of WSD.

A sense inventory provides the senses which a word can associate to. Earlier works on the WSD problem focus on disambiguating words using thesauri or machine-readable dictionaries as the sense inverntories (Lesk, 1986; Guthrie et al., 1991; Yarowsky, 1992). Later, WordNet (Miller et al., 1990) becomes the most widely used sense inventory for WSD. More recently, Mihalcea (2007) considers Wikipedia and uses the hyperlinked text in Wikipedia articles to generate annotated training data automatically. Other encyclopedia, such as BabelNet (Moro et al., 2014), also has been used for multilingual WSD.

Techniques

WSD can be viewed as a classification problem, where word senses are the classes. A classification model assigns one or more senses to each word based on the contextual clues in the given text. The context of a word is usually represented by a set of features. For instances, neighboring words or neighboring part-of-speech tags of the target word, syntactic cues such as argument-head relations between the target word and other words in the same sentence, and previously disambiguated senses of words, etc.

If there are sense-annotated datasets, supervised classifiers in the machine learning literature can be applied. For instance, Yarowsky (1994) uses decision lists, Kelly and Stone (1975) and Black (1988) use decision trees, Cottrell (1985) and Veronis and Ide (1990) employ neural networks, Escudero et al. (2000) and Murata et al. (2001) apply support vector machines, and Mooney (1996) shows that naive bayes compares well with 6 other supervised models.

When there is only little training data, the bootstrapping algorithm is proposed. There are two main approaches to bootstrapping in WSD: co-training and self-training. For instance, Yarowsky (1995) proposes a self-training approach which relies on *one sense per collocation* and *one sense per discourse* assumptions. Mihalcea (2004) compares a co-training approach to a self-training approach. The co-training approach contains two classifiers which use local and topical information respectively, whereas the self-training approach uses both information in a single classifier.

Unsupervised and knowledge-based approaches are also studied extensively when there is no sense-annotated dataset. A more comprehensive survey of techniques used in WSD can be found in Navigli (2009).

Evaluation Methodology

To evaluate a stand-alone WSD system, the most widely used metric is the F1 score which is derived by the *precision* and *recall* of the system:

$$\text{precision} = \frac{\text{number of correct answers provided}}{\text{number of answers provided}}$$
$$\text{recall} = \frac{\text{number of correct answers provided}}{\text{total number of test instances}}$$

The F1 score is the harmonic mean of precision and recall:

$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{2.1}$$

Higher F1 score means the system has better WSD performance.

To demonstrate the real objective of WSD, WSD is also evaluated in end-to-end systems such as information retrieval and machine translation. In this evaluation, WSD is a module embedded in a real-world application. The end-to-end performance of the application is measured while changing the WSD module.

2.1.2 Named Entity Recognition

Named Entity Recognition (NER) is an information extraction problem where the goal is to identify and type phrases that are names of persons, organizations, locations, and so on. For example, given the following sentence:

U.N. official Rolf Ekeus heads for Baghdad,

the goal is to identify that *U.N.* is an organization, *Rolf Ekeus* is a person, and *Baghdad* is a location. The term "named entity" was first used at the 6th Message Understanding Conference (MUC) (Grishman and Sundheim, 1996). Since named entities carry the most informative message of text, NER is important for understanding large bodies of text and is considered an essential pre-processing stage in many natural language processing and information extraction systems.

NER is close to the WSD problem in the sense that they both disambiguate short piece of text based on the context in which it occurs. However, there are two major differences. First, NER requires to identify proper nouns which could consist of more than one words, but WSD focuses on words which are common nouns, verbs, and adjectives. Second, NER uses single set of coarse-grained types for all proper nouns, whereas in WSD, each word has its own set of senses.

Different datasets may focus on different types of named entities. For instance, CoNLL 2003 shared task (Tjong Kim Sang and De Meulder, 2003a) uses PER (person), ORG (organizations), LOC (locations), and MISC (miscellaneous). Ontonotes (Hovy et al., 2006) adds NORP (nationalities, religious or political groups), FAC (facilities), GPE (geographical/political entities, such as countries and cities), PRODUCT (vehicles, weapons, etc.), EVENT (sport events, wars, etc.), WORK OF ART (books, songs, etc.), LAW (named documents made into laws), LANGUAGE, and values in the text are classified into DATE, TIME, PERCENT, MONEY, QUANTITY, ORDINAL, and CARDINAL. The more recent Text Analysis Conference (TAC) Knowledge Base Population (KBP) (Ji et al., 2014, 2015, 2016) focuses on PER, ORG, LOC,

	U.N.	*official*	*Rolf*	*Ekeus*	*heads*	*for*	*Baghdad*
BIO	*B-ORG*	*O*	*B-PER*	*I-PER*	*O*	*O*	*B-LOC*
BILOU	*U-ORG*	*O*	*B-PER*	*L-PER*	*O*	*O*	*U-LOC*

Table 2.1: BIO and BILOU labeling schemes for NER.

GPE, and FAC.

Techniques

NER is known to perform well if there is enough annotated training data. Recent supervised machine learning models often view NER as a sequence prediction problem where a sentence is a sequence of words and each word can take one of the class labels. The two commonly used labeling schemes, BIO and BILOU, are shown in Table 2.1. In both schemes, "B" indicates the beginning of the phrase, "I" indicates the word is inside the phrase, and the "O" label means the word does not belong to any named entity. The BILOU labeling scheme uses two more modifiers: "L" is used to label the last word in the phrase if the phrase contains more than one tokens, and "U" is for the phrases which are one-token long. If we use 4 entity types, there are 9 classes in total which each word can take using the BIO scheme and 17 classes using the BILOU labeling scheme. Note that most words have the "O" tag since relatively only a small amount of words are belong to named entity mentions. Using the BILOU scheme allows to learn a more expressive model and is shown to perform better than using the BIO scheme (Ratinov and Roth, 2009).

The typical machine learning models for NER include Hidden Markov Model (Rabiner, 1989), Conditional Random Fields (Lafferty et al., 2001), and sequential application of Perceptron or Winnow (Collins, 2002). Actually, the state-of-the-art result can be achieved by using a averaged perceptron with greedy decoding (Ratinov and Roth, 2009). The details of the standard features used in NER will be introduced in Chapter 4.1. Recently, neural network based sequence-to-sequence models (Lample et al., 2016) are shown to obtain comparable results on the well-studied datasets with minimal feature engineering.

Evaluation Methodology

Similar to the evaluation measure for WSD, the most common evaluation metric for NER is the phrase-level F1 score. If we represent each mention in the text as a tuple of three fields, (begin offset, end offset, entity type). Let P be the set of predicted mentions from a system and G be the set of ground truth mentions, we have

$$\text{precision} = \frac{|P \cap G|}{|P|}$$
$$\text{recall} = \frac{|P \cap G|}{|G|},$$

where $|\cdot|$ denotes the number of elements in the set. The F1 score (Eq. (2.1) of this system is the harmonic mean of precision and recall.

2.2 Related Techniques

In this section, we briefly introduce the machine learning approaches that will be used in this thesis. We start with supervised classification and ranking models, and then discuss the need to acquire supervision signals when there is no manually-annotated examples for the target task. Finally, we discuss the distributed representation for words.

2.2.1 Supervised Classification

In a binary classification problem, the output is a binary variable $y \in \{1, -1\}$ for each instance. Given a set of training data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{l}, \boldsymbol{x}_i \in R^n, y_i = \{1, -1\}$, where $\boldsymbol{x}_i$ is the feature vector of the i-th training example and y_i is the corresponding label, the goal of linear classification models is to learn a linear function,

$$y = sign(\boldsymbol{w}^T \boldsymbol{x}), \tag{2.2}$$

to predict the label y of an instance $\boldsymbol{x}$. That is, the linear function is parameterized by the weight vector $\boldsymbol{w} \in R^n$. If the dot product between the model $\boldsymbol{w}$ and the instance $\boldsymbol{x}$ is positive, the prediction is 1. Otherwise it outputs -1.

Several learning algorithms have been proposed and studied for estimating the weights $\boldsymbol{w}$ in Eq. (2.2) (e.g., (Minsky and Papert, 1969; Platt, 1999; Ng and Jordan, 2001)). In the following, we discuss a simple yet effective approach, averaged perceptron, which is used as the classifier in our NER model (Chapter 4.1).

The Perceptron Algorithm

The perceptron is a classic learning algorithm which is *online* and *error-driven*. *Online* means instead of using the entire training data to update the model, the algorithm looks at one training example at a time. That is, the algorithm updates the model only based on the training example which it is considering, and

Algorithm 1 The perceptron algorithm

```
Require: Training data $\mathcal{D}$
Ensure: The learned model $\boldsymbol{w}$
1: Initialize $\boldsymbol{w} \leftarrow 0$
2: for $t = 1, 2, \ldots,$ maxIteration do
3:     Randomly shuffle the data
4:     for $(\boldsymbol{x}, y) \in \mathcal{D}$ do
5:         if $y\boldsymbol{w}^T\boldsymbol{x} \leq 0$ then
6:             $\boldsymbol{w} \leftarrow \boldsymbol{w} + y\boldsymbol{x}$
7:         end if
8:     end for
9: end for
```

then moves on to another example. The pseudocode of the algorithm is shown in Algorithm 1. For each training instance $(\boldsymbol{x}, y)$, line 5 checks if the current model makes a mistake on the instance. If so, the weight vector $\boldsymbol{w}$ is updated by adding $y\boldsymbol{x}$.

Averaged Perceptron

The issue of the perceptron algorithm is that it emphasizes the later training instances more than the earlier ones. For example, if we have a good model which makes no mistake on the first 99 training instance, but it predicts wrongly on the 100th instance. The model will be changed according to the 100th instance without considering the impact on the first 99 instances.

A simple remedy to this issue is to give good models more weights than the models which make very few correct predictions. As the perceptron learns, it keeps a collection of weight vectors and the corresponding survival times: $\{(s^1, \boldsymbol{w}^1), (s^2, \boldsymbol{w}^2), \ldots, (s^k, \boldsymbol{w}^k)\}$, where s^k indicates the number of instances on which $\boldsymbol{w}^k$ predicts correctly. The final model is the weighted average of this sequence of models instead of simply taking the last model. That is, suppose there are K models during training, at test time, the prediction of an instance $\boldsymbol{x}$ is

$$y = sign((\sum_{i=1}^{K} s^i \boldsymbol{w}^i)^T \boldsymbol{x}),$$

where $\sum_{i=1}^{K} s^i \boldsymbol{w}^i$ can be viewed as the weighted average of the intermediate models during the training process.

2.2.2 Learning to Rank

Ranking is a central component in many information retrieval problem, such as document retrieval, collaborative filtering, and online advertising. Given a query and a set of instances, a ranking function assigns a score to each instance which indicates how relevant it is to the given query. A higher-scored instance is more relevant than an instance has a lower score, hence ranked higher. In the following, we discuss a popular ranking model, ranking support vector machines (SVM), which we use as the ranker in the wikification pipeline.

Linear Ranking SVM

Ranking SVM (Herbrich et al., 2000) is a pairwise approach which approximates the ranking problem by a classification problem. That is, it learns a binary classifier which can tell which instance is more relevant given a pair of instances. The goal is to minimize the losses induced by the inversions in ranking. More formally, given a set of training tuples $(q_i, y_i, \boldsymbol{x}_i)$, where q_i is the query associates with the i-th instance $\boldsymbol{x}_i \in R^n$, and $y_i \in R$ is the corresponding relevance score. By defining the set of preference pairs

$$P = \{(i, j) | q_i = q_j, y_i > y_j\},$$

L2-loss linear ranking SVM minimizes the following objective function:

$$\min_{\boldsymbol{w}} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C \sum_{(i,j)\in P} \max(0, 1 - \boldsymbol{w}^T(\boldsymbol{x}_i - \boldsymbol{x}_j))^2, \tag{2.3}$$

where the first term is the regularization term, the second term is the sum of training losses, and $C > 0$ is a parameter which balances the two terms. Since $\boldsymbol{x}_i$ is more relevant than $\boldsymbol{x}_j$, minimizing the loss function will find a $\boldsymbol{w}$ such that $\boldsymbol{w}^T\boldsymbol{x}_i - \boldsymbol{w}^T\boldsymbol{x}_j > 1$, otherwise, the positive loss will make the function value larger. For solving Eq.(2.3) efficiently, several works (Chapelle and Keerthi, 2010; Joachims, 2006; Sculley, 2009; Lee and Lin, 2014) have proposed different optimization techniques.

After obtaining the weight vector $\boldsymbol{w}$, at test time, an instance $\boldsymbol{x}$ which has a larger $\boldsymbol{w}^T\boldsymbol{x}$ should be ranked higher.

2.2.3 Word Embeddings

Word embeddings, also known as vector space representations or distributional representations, are real-valued vectors whose relative similarity correlates with the semantic similarity of the associated words. Such vectors are shown to be useful not only in computing similarity between terms, but also in several downstream NLP applications such as text classification, sentiment analysis, and named entity recognition.

Word embeddings are based on the idea that the semantic of a word can be determined by the contextual information (Harris, 1954; Firth, 1957). Earlier works use hand-crafted features from the context to represent words. More recently, approaches to generate contextual features automatically are proposed. One of the most influential model is Latent Semantic Analysis (LSA) (Deerwester et al., 1990), which was developed in the context of information retrieval. Different models may use different types of contextual information. In LSA, the entire document is used as the context, whereas in the recent neural network based models, the context of a target word usually only contains few neighboring words which are in the same sentence. In the following, we introduce a widely used word embeddings model, skip-gram model (Mikolov et al., 2013a), which serves as the base model for our cross-lingual embeddings in Chapter 4.3.

The Skip-Gram Model

The skip-gram model maximizes the following objective:

$$\sum_{(w,c)\in D} \log \frac{1}{1+e^{-v'_c \cdot v_w}} + \sum_{(w,c)\in D'} \log \frac{1}{1-e^{-v'_c \cdot v_w}},$$

where w is the target word, c is a context word within a window of w , v_w is the embedding of the target word w, v'_c is the embedding of c in the context, D is the set of training documents, and D' contains the randomly sampled token pairs which serve as negative examples. This objective is maximized with respect to variables v_w and v'_w for all w in the vocabulary. In this model, target words are used to predict the context word. The word pairs in the training documents are positive examples, and the randomly sampled pairs are negative examples.

Chapter 3

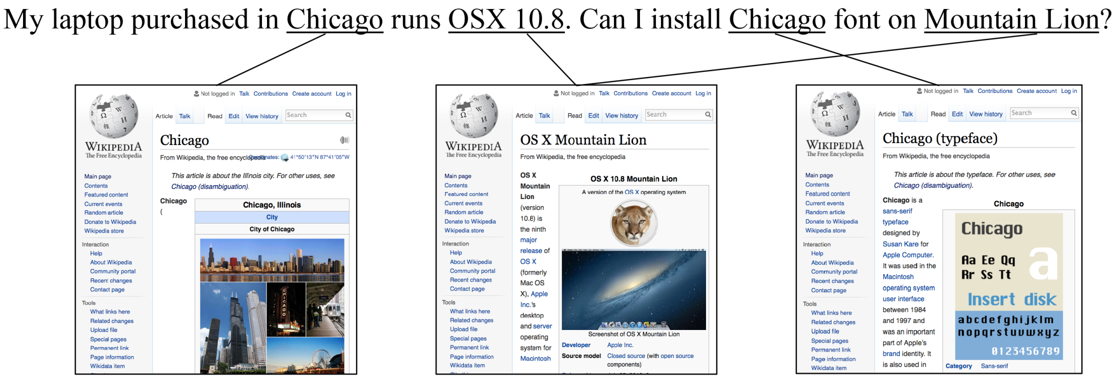
Wikification

As the number of entries in Wikipedia grows dramatically in recent years, Wikipedia has become an indispensable resource in knowledge acquisition and text understanding for both human beings and computers. The task of *Entity Linking* aims at extracting and disambiguating mentions (sub-strings in text) to the corresponding entries in a given knowledge base such as Wikipedia and FreeBase. If the target entity or concept does not exist in the given knowledge base (KB), "NIL" should be returned as the answer. When the target knowledge base is Wikipedia, this task is also referred as *Wikification*. In this chapter, we focus on introducing the Wikification problem since Wikipedia is the most widely used KB in the literature. In Chapter 5, we will discuss an extension of using multiple KBs other than Wikipedia.

An example of wikified text is shown in Figure 3.1. Given the input text, a Wikification system extracts the underlined mentions, and also links them to the corresponding Wikipedia titles (entries). We can see that the challenges in Wikification are due both to ambiguity and variability in expressing entities and concepts: a given mention in text, e.g., Chicago, may refer to different titles in Wikipedia (the city, the computer font, or many other entities), and a title can be expressed in the text in multiple ways, such as synonyms and nicknames (Mountain Lion is the name of Mac OS X version 10.8).

Wikification is closely related to WSD (Chapter 2.1.1). If the target mentions are common nouns, it is exactly the same as WSD using Wikipedia as the sense inventory, although Wikipedia might not be the most ideal sense inventory for all kinds of nouns. Wikification usually focuses on extracting and disambiguating named entities, because most entries in Wikipedia are entities. If the target mentions are named entities, there are two main differences between WSD and Wikification. First, the mention is complete in WSD, but in Wikification, the mention is potentially incomplete since an entity can be expressed in various ways, such as abbreviations or aliases. Therefore, the mention extraction step is more challenging in Wikification. Second, the candidate senses for a word are given in WSD. Due to potentially incomplete mentions and huge ambiguity of names, Wikification has an additional step to retrieve title candidates from millions of Wikipedia titles based on the mention surface strings.

As Wikification usually focuses on named entities, NER (Chapter 4.1.1) is commonly used as the mention



My laptop purchased in Chicago runs OSX 10.8. Can I install Chicago font on Mountain Lion?

Chicago

From Wikipedia, the free encyclopedia Coordinates: 41°50′13″N 87°41′05″W

This article is about the Illinois city. For other uses, see Chicago (disambiguation).

Chicago (

Chicago, Illinois

City

City of Chicago

OS X Mountain Lion

From Wikipedia, the free encyclopedia

OS X Mountain Lion (version 10.8) is the ninth major release of OS X (formerly Mac OS X), Apple Inc.'s desktop and server operating system for Macintosh

OS X 10.8 Mountain Lion

A version of the OS X operating system

Screenshot of OS X Mountain Lion

Developer Apple Inc.

Source model Closed source (with open source components)

Chicago (typeface)

From Wikipedia, the free encyclopedia

This article is about the typeface. For other uses, see Chicago (disambiguation).

Chicago is a sans-serif typeface designed by Susan Kare for Apple Computer. It was used in the Macintosh operating system user interface between 1984 and 1997 and was an important part of Apple's brand identity. It is also used in

Chicago

Category Sans-serif

Figure 3.1: An example of the Wikification problem.

extraction step for Wikification. That is, an NER model is applied on the given text to extract named entity mentions which are then grounded to Wikipedia. Nevertheless, Wikification can be viewed as an extremely fine-grained NER problem conceptually. Namely, instead of using few common entity types (E.g., PER, ORG, LOC, MISC) as in the traditional NER problem, Wikification uses millions of types (each entity entry in Wikipedia is a type). However, the techniques used in the NER problem is not suitable for solving Wikification since the label space is simply too large. The training data is impossible to cover all titles in Wikipedia.

Wikification has been studied extensively recently (Bunescu and Pasca, 2006; Cucerzan, 2007; Mihalcea and Csomai, 2007; Milne and Witten, 2008; Ratinov et al., 2011). A common framework for solving Wikification problem consists of four steps:

- **Mention Extraction:** Given the input text, the first step is to identify the mentions (phrases in text) which will be disambiguated in the later steps. Different applications may have different definitions of mention. For instance, mentions could be proper nouns, common nouns, or any keyphrases that are explained in Wikipedia.
- **Title Candidate Generation:** The second step is to generate a set of title candidates for each extracted mention. The goal is to quickly reduce the number of possible Wikipedia titles from millions to a manageable size, so that a more sophisticated and resource-hungry algorithm can be applied to disambiguate these candidates. There is a trade-off here: too many candidates may make the performance of the ranker in the next step suffer; on the other hand, the correct answer may not be included if we only retrieve a small number of candidates. The fundamental problem in this step is computing name similarity.
- **Title Candidate Ranking:** Given the extracted mentions and the corresponding title candidates, a ranking model assigns a score to each title candidate which indicates how relevant the title is to

the mention. The key challenge in this step is how to measure similarity between mentions and titles. That is, how to better represent the mentions in the input document and the entries in Wikipedia, so that the resulting similarity achieves disambiguation.

- **NIL Mention Identification:** This step determines if the top ranked title is actually the answer. If the target concept or entity does not have an entry in Wikipedia, the answer proposed by the ranker should be rejected. These unlinkable mentions are often referred as *NIL mentions*.

In the application of knowledge base population in which the goal is to enrich a knowledge base using the information extracted from text, there is an additional *NIL mention clustering* step. The idea is that these unlinkable mentions are potentially new entities which can be added into a knowledge base. In order to gather more accurate and comprehensive information (e.g., relations with other entities) of these NIL mentions, it is useful to cluster the NIL mentions which refer to the same entity. This problem is the co-reference resolution problem, which is out of the focus of this thesis.

In this chapter, we survey techniques and resources used in Wikification. Moro et al. (2014); Shen et al. (2015); Ling et al. (2015) have discussed several aspects of Wikification. We intend to provide a more comprehensive survey and also include recent progresses. This chapter is organized as follows: Section 3.1 introduces Wikipedia and the related knowledge bases which are widely used in solving the Wikification problem. Section 3.2, 3.3, 3.4, and 3.5 survey the techniques which have been proposed for the four main components of the Wikification pipeline. Finally, we discuss data sets and the evaluation metrics in Section 3.6.

3.1 Preliminaries

A knowledge base is the fundamental component for Entity Linking. The minimum requirement of a knowledge base is that it should contain unambiguous entries which refer to entities in the world or concepts in a field. A knowledge base usually also contains properties or attributes of each entry, for instance, definition or description, category information, or even relationships with other entries in the knowledge base. Since we are focusing on the Wikification task, in the following, we briefly introduce the KBs which are closely related to Wikipedia and are used in solving the Wikification problem.

3.1.1 Wikipedia

Wikipedia[1] is a free online multilingual encyclopedia with the aim to allow anyone to edit it. It is the most popular Internet encyclopedia in the world and is a very quickly growing resource. As in June 2017, there are 5.4 million entries in the English Wikipedia.

An entry in Wikipedia is an article which can be referred by an unique **title** (e.g., Barack_Obama[2]) or identifier (e.g., page id: 534366). The article describes important information of the entity. Moreover, phrases in this article could be linked to the corresponding entries in Wikipedia. For instance, in the first paragraph of Barack Obama, the phrase “African American” is linked to Wikipedia page “African_Americans”, and the phrase “U.S. Senate” is linked to the page “United_States_Senate”. These hyperlinked phrases are usually referred as **anchor text**. They are created by the users with intension to help readers to comprehend the article easier.

At the end of each Wikipedia article, there is a list of language names. Each of these language names links to the page of this entity in another language. This information is often referred as **inter-language links**, which are very useful in the cross-lingual applications (Chapter 4).

Every article in Wikipedia is required to have at least one **category**. For instance, some categories of Barack Obama page are “Presidents of the United Stats”, “Illinois Democrats”, and “University of Chicago Law School faculty”. Categories allow articles to be placed into one or more topics. These topics can be further categorized by one or more parent categories.

Another useful structure is the **disambiguation pages**, which are created for ambiguous names. For example, “CIA_(disambiguation)” is a disambiguation page which contains about 30 titles which can be referred by “CIA”. Some examples are “Cairo_International_Airport”, “California_Institute_of_the_Arts”, and “Certified_Internal_Auditor”.

There is a special kind of pages which is called **redirect**. A redirect page exists for each alternative name which can be used to refer to an entity in Wikipedia. For example, “America”, “US”, “U.S.”, and “USA” are some of the redirect pages for the Wikipedia title “United_States”. Using the redirect titles in URL will bring users to the page of the target entry (United_States). This resource is very useful in normalizing entity names in text.

[1] `https://en.wikipedia.org/`

[2] When expressing Wikipedia titles, we use underscores to represent white spaces.

3.1.2 DBpedia

DBpedia[3] is a multilingual knowledge base constructed by extracting structured information from Wikipedia such as infoboxes, categories, geo-coordinates, and the links to external web pages. The 2014 English version of DBpedia contains 4.58 million things (instances) and 68 million facts of these instances. Since DBpedia is organized in database structure, it is easier to obtain relations between entities using DBpedia than parsing the Wikipedia dump.

3.1.3 Freebase

Freebase is a large knowledge base which was collaboratively created mainly by its community members. It is an online collection of structured data harvested from many sources such as Wikipedia, NNDB, and MusicBrainz. As of 2014, Freebase had approximately 44 million instances and 2.4 billion facts. The typing information in Freebase is considered much cleaner than the categories in Wikipedia, hence people usually use the **Freebase types** of each Wikipedia title in the Wikification problem.

In 2015, Freebase was officially shut down, although the database dump is still available online[4]. It is replaced the Wikidata project[5]. We skip the introduction to Wikidata since it is a relatively new resource to the Wikification problem. However, we believe it will be very useful for acquiring additional information for the Wikipedia entries.

3.2 Mention Extraction

After knowing Wikipedia and the related resources, now we start to discuss the Wikification pipeline. The first step of the pipeline is to detect the phrases in text which we would like to disambiguate. Mention extraction is not a trivial problem and it is a critical step to Wikification since it is the first step of the pipeline. Different applications may need different types of mentions. Ling et al. (2015) discuss different annotation styles among different datasets. Many Wikification or Entity Linking works simply skip this step by taking the gold mentions in the datasets as inputs, and only focusing on disambiguating the gold mentions.

Mihalcea and Csomai (2007) view mention extraction as a keyword extraction problem, and propose an unsupervised approach. They construct a controlled vocabulary which contains all Wikipedia titles and frequent anchor texts. Given a piece of text, they first extract all *n*-grams which match some entries in the

[3] `http://wiki.dbpedia.org/`

[4] `https://developers.google.com/freebase/`

[5] `https://www.wikidata.org`

vocabulary. The matched n-gram candidates are then ranked by their "keyphraseness", which measures how often a term is hyperlinked (anchor text) in the entire collection of Wikipedia documents.

Milne and Witten (2008) learn a supervised classifier from Wikipedia anchor text to decide if an n-gram is a mention. They use the later steps of the pipeline to disambiguate every n-grams in order to generate better features for this mention classifier. Some of the features they use indicate the generality, location, and spread of the phrase. These features try to capture the ideas that the phrases mentioned in the first paragraph tend to be more important, and that how consistently the document discusses this phrase.

Cucerzan (2007) truecases the input text and then applies an named entity recognizer to extract mentions. Recently, named entity becomes the most popular definition of mentions for the Wikification task, since named entities are better defined and Wikipedia contains mostly named entities. In addition, many recent Wikification works are driven by several years of the Text Analysis Conference Knowledge Base Population shared tasks (McNamee and Dang, 2009; Ji et al., 2010, 2011, 2014, 2015, 2016) in which the target mentions are named entities.

Recently, several researchers propose to jointly perform NER and Entity Linking since the two tasks may reinforce each other. That is, the coarse named entity type of a mention may restrict possible Wikipedia titles, and the target Wikipedia entry may give hints on mention boundaries and types. Guo et al. (2013) formulate mention detection and disambiguation together as a structured prediction problem. Each n-gram that matches some anchor text becomes a mention candidate. The structural SVM model jointly grounds mention candidates to Wikipedia or rejects mention candidates. Since tweets are very short, the joint learning and inference is feasible. Similar to this idea, Sil and Yates (2013) also over-generates mention candidates and let the model to link or to reject the mention candidates simultaneously. However, since news articles are much longer than tweets, it is not tractable to consider all n-grams as mention candidates. Instead, they use the mentions detected by an NER, the noun phrases identified by a shallow parser, and some heuristic rules to expand the existing mention candidates. Furthermore, they partition mention candidates in a document into several groups based on how close they are to each other. The model only performs joint prediction on the mentions in the same group.

Luo et al. (2015) jointly model NER and Wikification by incorporating ranking features for disambiguation into a linear-chain NER model. The extended semi-CRF (Sarawagi and Cohen, 2004) not only directly models mention boundaries, but also considers entity distribution and mutual dependency over segmentations. Their model achieves very good results on the CoNLL-AIDA dataset.

Nguyen et al. (2016) also propose a model to jointly model NER and Wikification. The label for each token becomes a concatenation of the NER label and Wikipedia title (e.g., PER:Barack_Obama). Instead of the widely used linear-chain model, they propose a tree model in which the factor connections between tokens are based on the results of a dependency parser. In their experiment, the tree model outperforms the linear chain model on both NER and end-to-end Wikification.

3.3 Title Candidate Generation

Given an extracted or gold mention, it is not feasible to compare it with all titles in Wikipedia. Even simple string matching would take a lot of time, not to mention extracting sophisticated features based on the context. Therefore, the goal of this step is to quickly generate a small set of title candidates that the mention may refer to. To achieve quick retrieval, simple name similarity or matching is usually applied to compare the mention surface string with title strings. That is, no contextual information in the query document is used in this step. Candidate generation is very critical to the overall performance of Wikification (Hachey et al., 2013). There is a trade-off of the size of candidate sets. On one hand, if we only generate few candidates, the correct title may not be included. On the other hand, too many candidates will make the ranking problem in the next step harder. This also indicates the initial ranking of title candidates is important.

This step is mainly achieved by dictionary based methods (Bunescu and Pasca, 2006; Mihalcea and Csomai, 2007; Cucerzan, 2007; Kulkarni et al., 2009; Zhang et al., 2010; Zheng et al., 2010; Gottipati and Jiang, 2011; Ratinov et al., 2011; Han and Sun, 2011; Zhang et al., 2011; Shen et al., 2012; Guo et al., 2013; Gattani et al., 2013; Luo et al., 2015; Lazic et al., 2015; Globerson et al., 2016). A dictionary based method collects as many as possible names that each Wikipedia title may be referred to, and use this information to construct a dictionary. More specifically, each entry in the dictionary is a (*key*, *value*) pair, where *key* is a name and the corresponding *value* contains all the possible titles which can be referred by the key. For instance, the key "Chicago" may have the following Wikipedia titles as the values: "Chicago", "Chicago_(magazine)", "Chicago_(band)", "Chicago_Park,_California", and so on. The (key, value) pairs could be collected using the following Wikipedia structures:

- **Title:** The title itself is included as a key for the title. In addition, the first letters of the tokens in the title can form another key for this title.
- **Anchor text:** The hyperlinked phrases in Wikipedia articles could provide name variations for Wikipedia entries. Namely, the hyperlinked phrase is a key, and the target title is the value.
- **Redirects:** For example, if the string "us" is redirected to Wikipedia title United_States, "us" will be considered as one of the keys for United_States.

- **Disambiguation pages:** The titles contained in a disambiguation page are included in the value set of the target string which is being disambiguated.

- **Bold phrases:** In the first paragraph of Wikipedia articles, sometimes there are bold-faced phrases which are the aliases of the title. For instance, in the page of United_States, "United State of America", "USA", "United States", "U.S.", and "America" are the bold-faced phrases in the very first sentence. These phrases can be used as the keys for the title United_States.

Besides the features in Wikipedia, CrossWiki (Spitkovsky and Chang, 2012) is used in the candidate generation step in several recent works (Ling et al., 2015; Ganea et al., 2016). CrossWiki is built by crawling the web to find hyperlinked phrases which point to some Wikipedia pages. This dictionary contains more than 175 million unique keys (strings) along with the Wikipedia titles they may represent. Chisholm and Hachey (2015) also use hyperlinks on the web to gather this mention-to-titles mapping.

Given a mention, the simplest way to generate title candidates is by exactly matching the mention surface string with the keys in the dictionary. If there is a match, the Wikipedia titles in the corresponding value set are taken as the candidates. Besides exact match, some systems (Dredze et al., 2010; Tsai and Roth, 2016a,b) use partial match. For instance, the keys that are wholly contained in the mention, or the keys have strong string similarity scores with the mention. The similarity measures could be character Dice score, skip bigram Dice score, or Hamming distance. For all the matched keys, the merged value sets are used as the candidates.

Before looking up the dictionary, some approaches try to correct misspellings in the mentions. This is particularly useful for text which has not been carefully edited such as documents from discussion forums, weblogs, and Twitter. Zhang et al. (2010) use the "Did you mean" feature in Wikipedia search to correct misspellings. For example, if we search "Abbot Nutrition" in Wikipedia, the first sentence of the result page is "Did you mean: abbott nutrition", which corrects the misspelled "Abbot". Similarly, Zheng et al. (2010) use query spelling correction of Google search engine to correct the misspellings in the mentions.

Simple co-reference resolution or surface form expansion is sometimes applied to resolve the short mentions before querying the dictionary (Cucerzan, 2007; Zheng et al., 2010; Gottipati and Jiang, 2011; Zhang et al., 2011; Pershina et al., 2015). Since short mentions are usually more ambiguous (e.g., acronyms and last names), the idea is that the corresponding full names could be mentioned somewhere in the same document. Most approaches use heuristic rules to match adjacent entities (e.g., University of Illinois at Urbana-Champaign (UIUC)), or in the entire document (e.g., "George W. Bush" may be mentioned before "Bush"). Zhang et al. (2011) propose to learn a supervised classifier to decide if a mention is the acronym of another in more challenging cases. For example, "CCP" stands for "Communist Party of China", and

"MOD"/"MINDEF"/"MD" either of which can stand for "Ministry of Defense". They not only look at the mention string, but also consider neighboring words of the mentions. They show substantial improvement over rule-based methods.

Besides the dictionary-based candidate generation method, some systems (Zhang et al., 2010; Dredze et al., 2010) leverage a search engine to query candidates. Zhang et al. (2010) query Wikipedia search for the infrequent mentions. Dredze et al. (2010) use the mention string to query Google search and collect Wikipedia pages within the top 20 returned documents as the candidates.

The retrieved candidate set could be very large for some highly ambiguous mentions, say, contain more than 100 title candidates. Especially when the partial match methods are used. This large set of candidates may make the ranking problem difficult, and also make the training and inference inefficient for more expressive models. Therefore, most systems only keep top k candidates from the full candidates set, where k is usually less than 30. The initial ranking of candidates is based on some popularity-based measures. A common approach is to compute title prior, $Pr(title|mention)$, from Wikipedia anchor text:

$$Pr(title|mention) = \frac{\#\text{ times the mention is linked to the title}}{\#\text{ times the mention is hyperlinked}} \tag{3.1}$$

3.4 Title Candidate Ranking

In the previous two sections, we have described approaches to extract mentions and retrieve Wikipedia title candidates. The next challenge is to select the best title from the candidate set according to the meaning of the mention in the given context. In the candidate generation step, approaches are usually based on name similarity, that is, no contextual information is used since the goal is to quickly produce a small set of titles which includes the answer. In order to pick the answer out of the candidate set, contextual information should be taken into account.

This key step of the Wikification pipeline is usually viewed as a ranking problem. Namely, a mention is a query and the corresponding title candidates are the possible outcomes. A ranking model will assign a score to each candidate, which indicates how relevant it is to the mention. The candidates are then ranked by these relevance scores. We broadly classify candidate ranking models into three groups based on the required resources: supervised methods, unsupervised methods, and semi-supervised methods. In the following sub-sections, we first introduce the common features used in these machine learning models, and then discuss the three groups of models.

3.4.1 Features

The features used in the ranking models are designed to capture some similarity between (mention, candidate) pairs. Supervised methods usually use multiple features and learn a model to combine these features, whereas unsupervised methods may only consider few features. We divide features into two groups: context-independent and context-dependent features.

Context-Independent Features

The context-independent features only measure similarity between the mention surface string and title candidates. Due to this context-independent nature, these features are also commonly used for the initial ranking in the candidate generation step.

- **Entity popularity:** The most widely used entity popularity feature is the probability of the title given the mention (Eq. 3.1), which is estimated from the anchor texts in Wikipedia or documents on the web. Almost all wikification systems (Hoffart et al., 2011; Kulkarni et al., 2009; Ratinov et al., 2011; Shen et al., 2012; Ling et al., 2015; Luo et al., 2015; Nguyen et al., 2016; Yamada et al., 2016; Tsai and Roth, 2016b) use this feature since many mentions simply refer to the most popular entity. Besides this prior, Guo et al. (2013); Gattani et al. (2013) use view statistics of Wikipedia pages as features. Dredze et al. (2010) look at the number of Wikipedia pages links to and from a candidate title to represent the popularity of the candidate.

- **Entity type:** The match between entity types of mentions and candidate titles has been shown to be useful. Since NER is usually applied to extract mentions, each mention will be assigned an entity type (person, location, organization, and so on). For Wikipedia titles, Dredze et al. (2010) infer entity types from the information in the infobox, and Ling et al. (2015) get typing information from the corresponding entries in Freebase.

- **Name similarity:** This type of features computes string similarity between mentions and candidate titles. Some features are based on string similarity measures such as edit distance (Zheng et al., 2010; Liu et al., 2013), character Dice, and Hamming distance (Dredze et al., 2010). Other common name comparison features include: whether the candidate title matches the mention exactly, whether the candidate title starts or ends with the mention, and the number of identical words between the candidate title and the mention.

Context-Dependent Features

Context-dependent features measure similarity of (mention, candidate) pairs using other words or phrases in the given documents. The fundamental difference between different features in this category is how to represent mentions using context in the documents and how to represent Wikipedia titles, so that some similarity between the mention and title representations can lead to the right title. According to the types of context, context-dependent features can be broadly classified into two categories:

- **Textual context:** In this category, mentions are represented by the words in the documents. For instance, the context of a mention could be the words in the entire documents (Guo et al., 2013; Liu et al., 2013), words in a small window around the mention (Bunescu and Pasca, 2006; Kulkarni et al., 2009; Han et al., 2011; Ratinov et al., 2011), or other mentions in the document (Hoffart et al., 2011; Zhang et al., 2010; Dredze et al., 2010). For each Wikipedia title, the context could be the words in the entire Wikipedia page (Bunescu and Pasca, 2006; Kulkarni et al., 2009; Ratinov et al., 2011; Han et al., 2011), words in the first paragraph (Kulkarni et al., 2009), words around the anchor texts which point to the title, anchor texts in the title page, or words in the categories of the title. Besides bag-of-words representations, one could also weight each word by the TF-IDF scores (Guo et al., 2013; Ratinov et al., 2011). After mention and title representations are constructed, the similarity measure between them could be cosine similarity, dot product, word overlap, or Jaccard similarity.

  Recently neural network based models are proposed to generate distributional representations for mentions and titles. He et al. (2013) jointly optimize document and entity representations for a given similarity measure. They apply stacked denoising auto-encoders to generate document representations, and the anchor texts in Wikipedia are then used as supervision to fine tune the representations toward the similarity measure. Tsai and Roth (2016b); Yamada et al. (2016) learn word and title representations by applying skip-gram models (Mikolov et al., 2013a) on Wikipedia documents. By substituting anchor texts with the corresponding titles, words and titles are mapped into the same continuous vector space. Francis-Landau et al. (2016) use convolutional neural networks to learn representations for mentions and titles at several levels (mention surface string, neighboring words, and the entire document).

- **Entity context:** Besides words in the document, entities (Wikipedia titles) exist in the document are also important clues for selecting the correct candidate. The idea is that the correct title candidate should be more coherent or more related to other titles in the document than other candidates are. Therefore, the features in this category try to capture similarity or topical coherency between Wikipedia titles.

Cucerzan (2007) uses the overlap between Wikipedia categories and words in the Wikipedia pages to measure the coherency between two Wikipedia titles. Milne and Witten (2008) propose the link-based measure. The idea is that two Wikipedia titles are more semantically related if there are more Wikipedia articles that link to both of them. Besides the incoming links to the titles, Ratinov et al. (2011) also use outgoing links to measure similarity between two titles. Given the sets of incoming links or outgoing links of two titles, Milne and Witten (2008) use Normalized Google Distance for the similarity measure. Ratinov et al. (2011) further calculate PMI (Point-wise Mutual Information) between the two sets of links, whereas Guo et al. (2013) use Jaccard similarity. For the tail titles, there might not be enough incoming links, outgoing links, and categories for measuring robust similarity. Hoffart et al. (2012) propose to extract keyphrases in the Wikipedia article, and compute similarity between two titles based on the overlap of keyphrases. Recently, Globerson et al. (2016) measure similarity between two Wikipedia titles using the number of Freebase relations between them, the number of hyperlinks between the two Wikipedia pages (in either direction), and the number of common mentions in the two pages.

The key problem of using entity coherence features is that decisions for different mentions are interdependent. That is, generating features for one mention depends on the disambiguation results of other mentions, and vice versa. One solution for this issue is to view this problem as a structured prediction problem which assigns answers to all mentions simultaneously (Guo et al., 2013; Nguyen et al., 2016; Ganea et al., 2016). However, structured prediction is know to be slow and hard to train since the number of possible outcomes grows exponentially with the number of mentions. Several systems break this interdependency among mentions using some approximations. Instead of looking at the disambiguated titles of other mentions, Cucerzan (2007) uses candidates of all other mentions in the document to represent the topic of the document. Milne and Witten (2008) only use the unambiguous mentions in the document. This approach relies on the presence of unambiguous mentions with high disambiguation utility. Ratinov et al. (2011) propose a two-stage approach. In the first stage, the ranker excludes the features which rely on context entities. The predictions of the first stage are then used to generate entity coherence features in the second stage. Tsai and Roth (2016b) only take the titles from the previously disambiguated mentions to generate these entity coherence features. Similar to Ratinov et al. (2011)'s two-stage approach, Globerson et al. (2016) take the most "supportive" candidate from each mention in the document as the context. The support is measured by similarity between titles and a proposed attention model.

3.4.2 Supervised Methods

Supervised methods learn how to score candidates using training data which contain labeled examples. Labeled examples are documents containing (mention, Wikipedia title), which could guide the models to better combine various features. Supervised methods are usually considered expensive in NLP applications since human annotators need to be trained to understand the task, and doing manual annotation is time consuming. For wikification, users of Wikipedia have manually created tons of labeled examples (anchor texts) in Wikipedia articles, hence most wikification systems leverage this free resource and apply supervised models. However, we note that Wikipedia documents may not be the most ideal supervision for all kinds of models and text genres, since the format of Wikipedia articles are very formal, many anchor texts are not named entities, and many linkable mentions are not linked (only the first mention of an entity is labeled).

We categorize supervised models into four categories: binary classification, learning to rank, structured prediction, and probabilistic model. Most supervised models generate several features (Section 3.4.1) for each (mention, title candidate) pair, each of which measures some aspect of relevancy between the mention and the title candidate. Different models make different assumptions and optimize different objectives.

Binary Classification

Binary classification models view each (mention, title candidate) as a binary decision problem. Namely, weather the mention refers to the title candidate or not. In training, each labeled example provides a positive instance. Other candidates generated by the system could provide negative instances. Some classifiers that have been applied to the Wikification problem are support vector machines (SVMs) (Milne and Witten, 2008; Pilz and Paaß, 2011; Zhang et al., 2010), logistic regression (Sil and Yates, 2013), naïve Bayes (Milne and Witten, 2008), and decision trees (Milne and Witten, 2008).

One issue of binary classification approaches is that multiple title candidates could be classified as positive, since the classifier makes decision for each mention independently. Milne and Witten (2008) do not resolve this issue and simply annotate a mention with multiple titles if there are more than one positive candidates. Pilz and Paaß (2011) pick the candidate which has the highest decision value from the SVM classifier. Zhang et al. (2010) use a vector space model which is based on several features used in the binary classifier to break tie.

Learning to Rank

In contrast to binary classification methods which make decision on each candidate independently of other candidates, learning to rank approaches model the preferences between candidates. Therefore, ranking models usually perform much better than binary classification models (Zheng et al., 2010). At prediction time, a ranking model assigns a score to each candidate, and the candidates are ranked according to these scores. The candidate with the highest score will be chosen as the answer. When training a ranking model, an ordered list of outcomes is usually provided. However, the training examples of Wikification are (mention, title) pairs. Namely, instead of knowing the preferences of all pairs of candidates, the models only learn from knowing that the correct title is more preferable than other titles in the candidate set.

Most systems that apply learning to rank framework (Bunescu and Pasca, 2006; Kulkarni et al., 2009; Dredze et al., 2010; Chen and Ji, 2011; Zhang et al., 2011; Ratinov et al., 2011; Shen et al., 2012; Globerson et al., 2016; Chisholm and Hachey, 2015; Yamada et al., 2016) use the ranking SVM model (Herbrich et al., 2000; Joachims, 2002). Ranking SVM is a pairwise ranking approach which applies binary classification on each pair of candidates to decide which candidate is more preferable. Zheng et al. (2010); Chen and Ji (2011) also use ListNet (Cao et al., 2007), a listwise ranking approach, which directly optimizes the evaluation metric, averaged over all mentions in the training data.

Structured Prediction

Since the decisions for the mentions in a document may be interdependent, the results could be sub-optimal if we resolve each mention independently. Although several approximation methods (Section 3.4.1) are proposed to generate features from the candidates of other mentions, this information could still be noisy. Structured prediction models address this issue by making predictions for multiple mentions simultaneously. However, since the label space becomes exponentially larger, inference speed is usually much slower and more training instances may be needed for models to generalize well.

Guo et al. (2013) use structural SVM to jointly perform mention detection and disambiguation on tweets. Because there are second-oder features between mentions, they order the mentions from left to right, and use beam search algorithm to find the joint assignment approximately.

Nguyen et al. (2016) jointly model NER and Wikification by concatenating NER labels and Wikipedia titles. For instance, the modified label for a token can be “PER:Barck_Obama”, where “PER” is the named entity type and “Barack_Obama” is the corresponding Wikipedia title. For each sentence, besides the well-studied linear-chain model, they propose a tree model in which a factor is added between two tokens if they are connected according to a dependency parser. The exact inference is still tractable by variants of the Viterbi algorithm. In addition, they propose another global model in which mentions across sentences are linked if there is any common title candidate. Since the structure is not a tree now, exact inference is intractable. They apply Gibbs sampling (Finkel et al., 2005) to approximate the solution. In their experiments, the global model performs better than the tree model and the linear-chain model in both NER and end-to-end Wikification.

Ganea et al. (2016) use gold mentions and focus on disambiguating all mentions in a document jointly. They propose a probabilistic graphical model which is essentially a complete graph since each mention depends on all other mentions. They use heuristics to prune the number of variables and apply loopy belief propagation to perform approximate inference. In the experiments, their model achieves state-of-the-art performance on several datasets.

Probabilistic Model

Unlike the above feature-based models, the models in this section maximize the likelihood of observing mentions, the corresponding titles, and context words in the training documents. Each model makes different assumptions of how the labels are generated, that is, the dependency between variables and how the joint probability is decomposed.

Han and Sun (2011) propose an entity-mention model. For each document, the entities are first chosen according to some popularity knowledge which gives the likelihood of an entity appearing in a document. Based on these entities, the mention surface strings and the context around each mention are then generated. Han and Sun (2012) further incorporate the idea of topical coherence and propose a topic-mention model. This model modifies the topic models by adding entities and entity mentions into the document generation process. For each document, the model first draws a topic, and then entities (titles) are generated according to the topic. Finally, entity mentions and other words in the document are generated based on the drawn entities.

Lazic et al. (2015) propose a selective context model which assumes that most features that appear in the context of a mention are not discriminative. A latent variable is used to select a single contextual feature for each mention. The features are other name mentions and noun phrases in the document. The document generation story is that the mentions are first chosen according to some prior probability. For each mention, the corresponding entity is then generated. The latent variable selects a relevant context feature that fires for this entity. Finally, the remaining features are drawn from a background distribution.

3.4.3 Unsupervised Methods

Unsupervised methods do not require labeled examples to train the model. Although the anchor texts in Wikipedia provide free training data for Wikification, several researchers develop unsupervised approaches with aims to be robust on different domains or to enhance an supervised model. Nevertheless, if there is in-domain training data, supervised methods usually can achieve better performance. Note that we include approaches which use a development set (small amount of labeled examples) to tune few hyper-parameters in the model.

Cucerzan (2007) employs a vector space model, in which the vector representation of the document is compared with the vector representation of title candidates. The title which has the highest similarity will be selected as the answer. Wikipedia categories and contexts in the Wikipedia page are used to represent a title candidate. For the document representation, all other mentions and context words are used.

Pan et al. (2015) construct a knowledge network for each mention based on Abstract Meaning Representation (AMR) Banarescu et al. (2013) of the document. The key idea is that AMR can be used to select more important mentions in the context for the target mention, since AMR provides richer analysis on text such as entity typing, co-reference, and some semantic roles. This mention knowledge network is then compared with each candidate's knowledge network which is constructed based on the relations in Wikipedia, DBpedia, and Freebase. If a candidate's network has more overlap with the mention's knowledge network, it will be ranked higher. Similar to this idea, Wang et al. (2015) construct a graph for all mentions in a documents, and disambiguate mentions jointly. In this graph, two mentions are connected if they are in a proximity or if there is a coreference relation. For each combination of candidate assignment in the mention graph, a candidate graph is built based on the relations between the selected candidates in the KB. Finally, each candidate graph is compared with the mention graph in order to select the best candidate assignments.

One category of unsupervised methods is the graph-based approaches. These approaches construct a graph using mentions and title candidates as nodes, and the edges between two nodes are based on the existence of relationships or compatibility of the two nodes. Han et al. (2011) use n-grams which are more likely to be keyphrases as the mention nodes and initialize their scores based on the TF-IDF measure. A random walk with restart algorithm is proposed to propagate these initial scores to title candidate nodes. Moreover, these scores will be propagated proportional to the weights on the edges. The edge weights are computed from mention-entity similarity or entity-entity similarity features. Moro et al. (2014) build a graph for each document using BabelNet[6], a semantic network constructed from multilingual Wikipedia, WordNet, and other resources. Instead of directly using the relations in BabelNet, for each title candidate, they perform random walk on BabelNet, starts from the this title candidate. The result is used to gather a semantic signature for this candidate, which contains the strongly related entities. Given a document, the proposed graph consists of nodes from the candidates of mentions, and edges from the semantic signatures. They further propose a densest subgraph heuristic to reduce the level of ambiguity and select the final titles.

[6] `http://babelnet.org`

Besides pure unsupervised methods, some approaches add an additional unsupervised inference upon the results of a supervised model. In other words, the scores obtained by a supervised model are used in an unsupervised method.

Hoffart et al. (2011) construct a mention-entity graph in which nodes are mentions and candidates. Each mention-entity edge is weighted by a combination of entity popularity features and textual context similarity features, which are computed from the training data. Each entity-entity edge is weighted by the link-based similarity between two Wikipedia pages. Given this graph, they propose an algorithm to compute a dense subgraph that contains exactly one mention-entity edge for each mention.

Kulkarni et al. (2009) try to select the best candidate for each mention jointly via maximizing an objective which consists of a local compatibility score and a label relatedness score. The local compatibility score for each candidate is obtained from a ranking SVM model, and the label relatedness score comes from some similarity measures between two titles (Section 3.4.1). Since exact inference over all candidate assignments is intractable, they propose an approach based on local hill-climbing and rounding integer linear programs. Similarly, in order to choose the best title assignments for all mentions simultaneously, Cheng and Roth (2013) formulate a Constrained Conditional Model (Roth and tau Yih, 2004; Chang et al., 2012) using Integer Linear Programming. The model consists of two part. The first part contains ranking scores from Ratinov et al. (2011)'s supervised ranker, and the second part contains relational scores for each pair of candidates from two mentions.

Similar to the graph-based approach proposed in Han et al. (2011), Pershina et al. (2015) perform a variant of personalized PageRank algorithm on the graph in which nodes are title candidates in a document, and an edge exists if the two titles are linked in Wikipedia. Besides the graph construction is different from Han et al. (2011), they initialize the score of each node by a supervised classifier.

3.4.4 Semi-Supervised Methods

The goal of semi-supervised methods is to improve a supervised model using large amount of unlabeled data. The probabilistic model proposed by Lazic et al. (2015) (see probabilistic model in Section 3.4.2) can be applied to the semi-supervised setting naturally. In their model, there is a variable indicates the probability of a title given a mention. If a mention is not labeled in the training data, the EM algorithm will calculate the expected likelihood with respect to this variable. Otherwise, the probability mass is simply set to the ground truth title. In their experiments, they use Wikipedia anchor texts as the labeled data, and use a Web corpus of 50 million pages as the unlabeled data. The model parameters are initialized by training the Naïve Bayes model on the labeled data. When they only use the labeled data, the model gets 79.7 F1 on the CoNLL-AIDA dataset. Adding unlabeled data makes it achieve 86.4 F1.

3.5 NIL Mention Identification

In the previous section, we have discussed approaches for choosing the best title from the title candidates. However, in some cases, the target entity does not exist in Wikipedia, but the candidate set is non-empty. The goal of this step is therefore to identify these mentions and output "NIL" as the answer for them.

Several studies (Cucerzan, 2007; Kulkarni et al., 2009; Han et al., 2011; Pershina et al., 2015; Yamada et al., 2016) simply do not handle this issue and assume target entities are in Wikipedia. In this case, NIL mentions are the mentions which have zero title candidates.

Bunescu and Pasca (2006); Gottipati and Jiang (2011); Shen et al. (2012); Ferragina and Scaiella (2010); Pilz and Paaß (2011); Li et al. (2013); Han and Sun (2012) employ a threshold on the ranking scores. Namely, if the ranking score of the top title candidate is lower than a threshold, "NIL" will be returned. The threshold could be learned from training data or manually tuned.

Another popular approach is to learn a supervised binary classifier to decide if the top candidate should be rejected (Zheng et al., 2010; Zhang et al., 2011; Han and Sun, 2011; Ratinov et al., 2011). The features are mostly the ones we introduced in Section 3.4.1. Besides only generating features from the mention and the corresponding top candidate, Ratinov et al. (2011) design additional features based on the top candidate and the runner-up.

In addition, some systems (Dredze et al., 2010; Guo et al., 2013; Han and Sun, 2011; Luo et al., 2015; Nguyen et al., 2016) do not have an additional step for NIL mention identification. They incorporate the NIL mention detection problem into ranking models. The most common way is to add a "NIL" candidate for each mention. Dredze et al. (2010) add an additional "NIL" candidate for each mention, and design features which capture properties which indicate if the mention could be NIL. For example, whether any of the candidates match the mention, and the max, mean, and difference between the features for all candidates. Guo et al. (2013) also add NIL as one of the candidate for each mention candidate. In their structural SVM model, a special bias feature is assigned to each mention, and the learned weight of this bias term will be used as the threshold to cut-off mentions. In the probabilistic model proposed by Han and Sun (2011), they use the idea that if a mention refers to an specific entity, the probability of this mention generated by this entity's model should be much higher than the probability of it is generated by a general language model. They add a NIL entity into the knowledge base and assume that the NIL entity generates mentions according to the general language model.

3.6 Evaluation Methodology

In this section, we discuss the commonly used labeled data sets, and various evaluation measures for Wikification.

3.6.1 Data Sets

In the following, we list data sets that researchers have been used for training or evaluation. Note that different data sets may use different versions of Wikipedia. If the gold labels are Wikipedia titles instead of the *Page IDs*, it might be difficult to evaluate the true performance if a system uses a newer version of Wikipedia. Since Wikipedia titles may be changed over time, gold titles in these data sets may not exist in the latest Wikipedia.

- **AQUAINT** (Milne and Witten, 2008) contains 727 mentions from newswire text. It is annotated to mimic the hyperlink style of Wikipedia. Namely, only the first mentions of important titles are annotated.

- **MSNBC** (Cucerzan, 2007) contains 747 mentions from MSNBC news. The mentions are extracted by running an NER and co-reference resolution system. In this dataset, all detected mentions are annotated.

- **ACE** (Ratinov et al., 2011) contains 257 mentions. This is a subset of the ACE co-reference data set. They leverage the gold typing and co-reference information to annotate the corresponding Wikipedia titles using Amazon Mechanical Turk.

- **Wikipedia** (Ratinov et al., 2011) contains 928 mentions from randomly selected 10,000 paragraphs in Wikipedia. Since most anchor texts can be easily resolved by the popularity features, the authors try to make this data set more challenging by removing most of the easy mentions. Note that several works (Bunescu and Pasca, 2006; Milne and Witten, 2008; Cucerzan, 2007)have created their own evaluation data using Wikipedia. However, this data set is made publicly available and is used in several studies.

- **KORE50** (Hoffart et al., 2012) contains 144 mentions from 50 short English sentences. The idea is to build a data set with high ambiguity.

- **AIDA-CoNLL** (Hoffart et al., 2011) contains roughly 35k mentions (including training, development, and test sets) from Reuters news articles. This data set was originally created for CoNLL 2003 NER shared task (Tjong Kim Sang and De Meulder, 2003b). Hoffart et al. (2011) hand-annotated all mentions with corresponding entries in YAGO2, Freebase, and Wikipedia. This data set is suitable for evaluating both NER and Wikification.

- **TAC** (Text Analysis Conference – Knowledge Base Population) hosts entity linking shared tasks from 2009 to 2017 (McNamee and Dang, 2009; Ji et al., 2010, 2011, 2014, 2015, 2016). The number of annotated mentions are always few thousands, and documents are mainly from news, discussion forums, and weblogs. Besides Wikification annotations, the mentions are also labeled with coarse named entity types. Wikipedia is used as the target knowledge base until 2013. Starting from 2014, mentions are linked to Freebase. Starting from 2015, the named entity mentions are annotated exhaustively, therefore are suitable for evaluating mention extraction or NER performance.

- **IITB** (Kulkarni et al., 2009) contains about 17k mentions from 107 web pages which are in the domains of sports, entertainment, science and technology, and health. The goal is to have high recall annotations, therefore the human annotators were told to be as exhaustive as possible.

3.6.2 Evaluation Metrics

In the earlier studies (Milne and Witten, 2008; Ratinov et al., 2011), Bag-of-Title F1 (BOT) is used to evaluate Wikification systems when the mentions are given. Given a document, let the set of predicted titles be P, and the set of gold titles be G. We have

$$\text{precision} = \frac{|P \cap G|}{|P|}, \text{and recall} = \frac{|P \cap G|}{|G|}.$$

The F1 score is the harmonic mean of precision and recall Eq. (2.1). However, since BOT does not evaluate which mention belongs to which title, a system can achieve a high BOT score without any correct prediction. This design is due to the different objective in the earlier works.

More recently, end-to-end performance of a wikification system is evaluated by the NER-like F1 score, which is mainly used in the TAC shared tasks. Each predicted and gold mention can be represented as a tuple of four fields: (start offset, end offset, entity type, Wikipedia title or NIL), where start and end offsets specify the mention boundaries. A predicted mention is considered matched with a gold mention if all fields in the tuple are identical. Some fields may be ignored according to the purpose of evaluation. For instance, for the systems which do not use NER in mention extraction, entity type is omitted in evaluation.

For the systems do not perform mention extraction and only focus on disambiguating gold mentions, the "Wikipedia title or NIL" field is the only evaluated field. In this case, the F1 score will be identical to precision and recall, which is also known as the "precision at one" or "accuracy" measures. Moreover, instead of only taking the top candidate into account, the “precision at k” evaluates if the answer is within the top k candidates.

Chapter 4

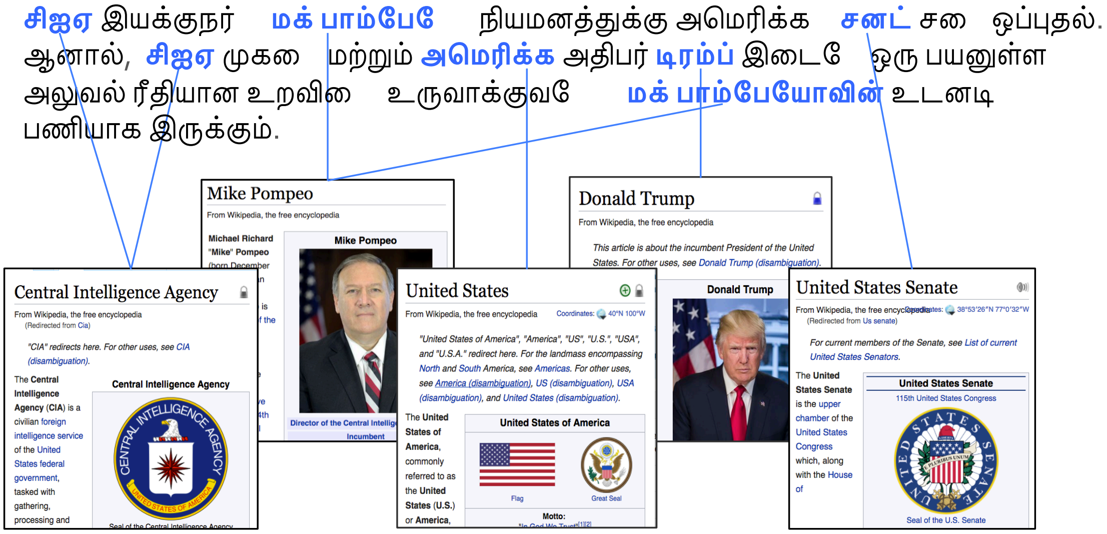
Cross-Lingual Wikification

Cross-lingual wikification is the problem of grounding entity mentions written in a foreign language to the English Wikipedia. That is, given a document written in a non-English language, the goal is to identify the entity mentions and also find their corresponding titles in the English Wikipedia. If the target entity does not exist in the English Wikipedia, "NIL" should be returned as the answer. Figure 4.1 shows an example. In this example, the input text is written in Tamil, and the goal of a cross-lingual wikification system is to extract the named entity mentions highlighted in blue and also find the English Wikipedia page for each mention.

This task is driven partly by the fact that a lot of information around the world is written in a foreign language for which there are limited linguistic resources and, specifically, no English translation technology. Instead of translating the whole document to English, grounding the important entity mentions in the English Wikipedia may be a good solution that could better capture the key message of the text, especially if it can be reliably achieved with fewer resources than those needed to develop a translation system. This task is mainly driven by the Text Analysis Conference (TAC) Knowledge Base Population (KBP) Entity Linking Tracks Ji et al. (2011, 2014, 2015, 2016), where the target languages are Spanish and Chinese.

The system pipeline of cross-lingual wikification is identical to the pipeline of the standard English wikification problem (Chapter 3), except that each step is much more challenging due to the multilingual and cross-lingual definition. For each section in this chapter, we discuss and address the key challenge for each component of the pipeline:

- **Foreign Mention Extraction (Section 4.1):** The first step is to identify the named entity mentions in the given non-English document. This is essentially the multilingual named entity recognition problem.
- **English Title Candidate Generation (Section 4.2):** Given a foreign mention, this step retrieves a set of English titles which could be referred by the mention. To achieve this, we need a mechanism to compare the foreign mention with English Wikipedia titles. One solution which we will discuss in



சிஐஏ இயக்குநர் மக் பாம்பேே நியமனத்துக்கு அமெரிக்க சனட் சபை ஒப்புதல். ஆனால், சிஐஏ முகமை மற்றும் அமெரிக்க அதிபர் டிரம்ப் இடைேய ஒரு பயனுள்ள அலுவல் ரீதியான உறவினை உருவாக்குவேத மக் பாம்பேயோவின் உடனடி பணியாக இருக்கும்.

Figure 4.1: An example of cross-lingual wikification. Given the Tamil text, the goal of cross-lingual wikification is to identify the highlighted named entity mentions, and also ground them to the English Wikipedia.

Section 4.2 is to translate foreign names into English.

- **English Title Candidate Ranking (Section 4.3):** In this step, the English title candidates will be ranked according to the relevancy to the input document. The key challenge of this step is to compute cross-lingual similarity between English Wikipedia titles and the contextual information in the foreign document.

We note that our goal is to solve the above challenges for all languages in Wikipedia. We do not develop language-specific models which rely on the resources or annotations only exist for a language. Rather, we explore the information exists in Wikipedia to facilitate our learning models. Based on the proposed techniques, we develop an end-to-end cross-lingual wikification system (demo[1] and source codes[2]).

[1] http://cogcomp.cs.illinois.edu/page/demo_view/xl_wikifier

[2] https://github.com/cttsai/illinois-cross-lingual-wikifier

4.1 Cross-Lingual Named Entity Recognition via Wikification

The first challenge in the cross-lingual wikification pipeline is to extract named entity mentions in the given foreign text (i.e., multilingual NER problem).

NER is successful for languages which have a large amount of annotated data, but for languages with little to no annotated data, this task becomes very challenging. There are two common approaches to address the lack of training data problem. The first approach is to automatically generate annotated training data in the target language from Wikipedia articles or from parallel corpora. The performance of this method depends on the quality of the generated data and how well the language-specific features are explored. The second approach is to train a model on another language which has abundant training data, and then apply the model directly on test documents in the target language. This direct transfer technique relies on developing language-independent features. Note that these two approaches are orthogonal and can be used together.

In this section, we focus on the second, direct transfer setting. We propose a cross-lingual NER model which is trained on annotated documents in one or multiple source languages, and can be applied to all languages in Wikipedia. The model depends on a cross-lingual grounding component (Section 4.3), which only requires multilingual Wikipedia, no sentence-aligned or word-aligned parallel text is needed.

The key contribution of this work is the development of a method that makes use of cross-lingual wikification and entity linking Tsai and Roth (2016a); Ji et al. (2015); Moro et al. (2014) to generate language-independent features for NER, and showing how useful this can be for training NER models with no annotation in the target language. Traditionally, wikification has been considered a downstream task of NER. That is, a named entity recognizer is first applied to identify mentions of interest, and then a wikifier is used to ground the extracted mentions to Wikipedia entries. In contrast to this traditional pipeline, we show that the ability to ground and disambiguate words is very useful to NER. By grounding every n-gram to the English Wikipedia, we obtain useful clues to NER, regardless of the target language.

Figure 4.2 shows an example of a German sentence. We use a cross-lingual wikifier to ground each word to the English Wikipedia. We can see that even though the disambiguation is not perfect, the FreeBase types still provide valuable information. That is, although “Albrecht Lehmann” is not an entry in Wikipedia, the wikifier still links “Albrecht” and “Lehmann” to people. Since words in any language are grounded to the English Wikipedia, the corresponding Wikipedia categories and Freebase types can be used as language-independent features.

The proposed model significantly outperforms comparable direct transfer methods on the Spanish, Dutch, and German CoNLL data. We also evaluate the model on five low-resource languages: Turkish, Tagalog, Yoruba, Bengali, and Tamil. Due to small sizes of Wikipedia, the overall performance is not as good as

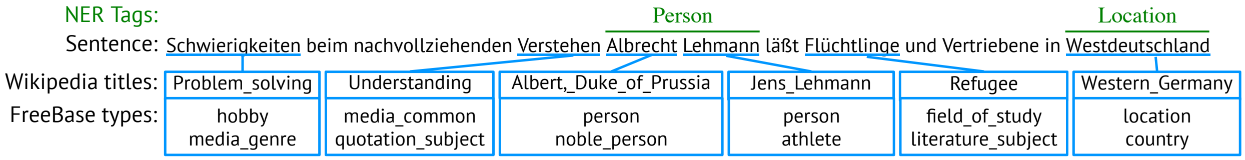


Figure 4.2: An example of a German sentence. We ground each word to the English Wikipedia using a cross-lingual wikifier. A word is not linked if it is a stop word or the wikifier returns NIL. We can see that the FreeBase types are strong signals to NER even with imperfect disambiguation.

the CoNLL experiments. Nevertheless, the wikifier features still give significant improvements, and the proposed direct transfer model outperforms the state of the art, which assumes parallel text and some interaction with a native speaker of the target language. In addition, we show that the proposed language-independent features not only perform well on the direct transfer scenario, but also improve monolingual models, which are trained on the target language. Another advantage of the proposed direct transfer model is that we can train on documents from multiple languages together, and further improve the results.

4.1.1 Named Entity Recognition Model

We use the state of the art English NER model of Ratinov and Roth (2009) as the base model. This model approaches NER as a multiclass classification problem with greedy decoding, using the BIO labeling scheme. The underlying classifier is averaged perceptron.

Table 4.1 summarizes the features used in our model. These can be divided into a base set of standard features which are included in Ratinov and Roth (2009), a set of gazetteer features which are based on titles in multilingual Wikipedia, and our novel cross-lingual wikifier features. The base set of features can be further divided into non-lexical and lexical categories.

Base Features

Non-Lexical Features Ratinov and Roth (2009) uses a small number of non-lexical features. For example, the previous tag feature is useful in predicting I- tags, because the previous tag should never be an O. The tag context feature looks in a 1000 word history and gathers statistics over tags assigned to words $[w_i, w_{i+1}, w_{i+2}]$. These features are included in all experiments.

In contrast with Täckström et al. (2012), we do not use POS tags as features. We could not get the universal POS tags for all languages in our experiments, and an earlier experiment indicated that adding POS tags does not improve the performance due to the accuracy of tagger.

Lexical Features Lexical features are very important for monolingual NER. In the direct transfer setting, lexical features are useful if the target language is close to the training language. We use a small number of

Base features	
Non-Lexical	
Previous Tags	(t_{i-1}, t_{i-2})
Tag Context	(distr. for $[w_i, w_{i+1}, w_{i+2}]$)
Lexical	
Forms	$(..., w_{i-1}, w_i, w_{i+1}, ...)$
Affixes	(prefixes and suffixes of w_i)
Capitalization	(w_i capitalized?)
Prev. Tag Pattern	(t_{i-2}, w_{i-1}, w_i)
Word type	(capital? digits? letter?)
Gazetteers	
Multilingual Wikipedia titles	
Cross-lingual Wikifier Features	
Freebase types of (w_{i-1}, w_i, w_{i+1})	
Wikipedia categories of (w_{i-1}, w_i, w_{i+1})	

Table 4.1: Feature groups. Base features are the features used by Ratinov and Roth (2009), the state of the art English NER model. Gazetteers and cross-lingual wikifier features are described in detail in Section 4.1.1.

simple features, including word forms, affixes, capitalization, and tag patterns. The latter feature considers a small window (at most 2 tokens) before the word in question. If there is a named entity in the window, it makes a feature out of NETag+$w_{i-2} + w_{i-1}$. Word type features simply indicate whether the word in question is all capitalized, is all digits, or is all letters.

Gazetteer Features

One of the larger performance improvements in Ratinov and Roth (2009) came from the use of (partial matches with) gazetteers. We include gazetteers also in our model, except we gather them in each language from Wikipedia. As in Ratinov and Roth (2009), we use the gazetteers as features. Specifically, we group them by topic, and use the name of the gazetteer file as the feature.

The method iteratively extends a short window to the right of the word in question. As the window increases in size, we search all gazetteers for occurrences of the phrase in the window. If we find a match, we add a feature to each word in the phrase according to its position in the phrase, either B for beginning, or I for inside.

This method generalizes gazetteers to unseen entities. For example, given the phrase “Bill and Melinda Gates Foundation”, “Bill” is marked as both B-PersonNames and B-Organizations, while “Foundation” is marked as I-Organizations. Imagine encountering at test time a fictional organization called “Dave and Sue Harris Foundation.” Although there is no gazetteer that contains this name, we have learned that “B-PersonName and B-PersonName B-PersonName Foundation” is a strong signal for an organization.

Cross-lingual Wikifier Features

As shown in Figure 4.2, disambiguating words to Wikipedia entries allows us to obtain useful information for NER from the corresponding FreeBase types and Wikipedia categories. A cross-lingual wikifier grounds words and phrases of non-English languages to the English Wikipedia, which provides language-independent features for transferring an NER model directly.

We use the system proposed in Tsai and Roth (2016a), which grounds input strings to the intersection of (the title spaces of) the English and the target language Wikipedias. The only requirement is a multilingual Wikipedia dump and it can be applied to all languages in Wikipedia.

Since we want to ground every n-gram ($n \leq 4$) in the document, deviating from the normal usage that only considers a few mentions of interest, we modify the system in the following two ways:

- The original candidate generation process queries the index by both whole input string and the individual tokens of the string. For the n-grams where $n > 1$, we generate title candidates only according to the whole string, not individual tokens. If we allow generating title candidates based on individual tokens then, for instance, the bigram "in Germany" will be linked to the title Germany thus wrongly considered as a named entity.

- The original ranking model includes the embeddings of other mentions in the document as features. It is clear that if we know what other important entities exist in the document, they provide useful clues to disambiguate a mention. However, if we want to wikify all n-grams, it makes no sense to include all of them as features, since the ranking model has already included features from TF-IDF weighted context words.

After wikifying every n-gram [3], we set the types of each n-gram as the coarse- and fine-grained FreeBase types and Wikipedia categories from the top 2 title candidates returned by wikifier. For each word w_i, we use the types of w_i, w_{i+1}, and w_{i-1}, and the types of the n-grams which contain w_i as features. Moreover, we also include wikifier's ranking features from the top candidate as features. This could serve as a linker Ratinov et al. (2011), which rejects the top prediction if it has a low confidence.

4.1.2 Experiments

We conduct experiments to validate and analyze the proposed NER model. First, we show that adding wikifier features improves results on monolingual NER. Second, we show that wikifier features are strong

[3] We set n to 4 in all our experiments.

	Latin Script							Non-Latin Script		
APPROACH	EN	NL	DE	ES	TR	TL	YO	BN	TA	AVG
Wiki size	5.1M	1.9M	1.9M	1.3M	269K	64K	31K	42K	85K	-
En. intersection	-	755K	964K	757K	169K	49K	30K	34K	51K	-
Gazetteer size	8.5M	579K	1M	943K	168K	54K	20K	29K	10K	-
Entities (train)	23.5K	18.8K	11.9K	13.3K	5.1K	4.6K	4.1K	8.8K	7.0K	-
Entities (test)	5.6K	3.6K	3.7K	3.9K	2.2K	3.4K	3.4K	3.5K	4.6K	-
Monolingual Experiments										
Wikifier only	71.57	57.02	49.74	60.13	52.84	51.02	29.35	47.78	38.05	50.83
Base Features	85.50	76.64	65.88	80.66	64.98	75.03	55.26	69.26	55.93	69.90
+Gazetteers	89.49	82.41	69.31	83.62	70.41	76.71	57.12	69.51	57.10	72.89
+Wikifier	**89.92**	**84.49**	**73.13**	**83.87**	**73.86**	**77.64**	**57.60**	**71.15**	**60.02**	**74.72**
Direct Transfer Experiments										
Wikifier only		40.44	39.83	43.82	41.79	42.11	27.91	**43.27**	**29.64**	38.01
Base Features		43.38	24.93	42.85	29.21	49.85	32.57	2.53	1.74	28.06
+Gazetteers		50.26	34.47	54.59	30.21	64.06	34.37	3.25	0.30	33.83
+Wikifier		**61.56**	**48.12**	**60.55**	**47.12**	**65.44**	**36.65**	18.18	5.65	**41.41**
Täckström baseline		48.4	23.5	45.6	-	-	-	-	-	-
Täckström bitext clusters		58.4	40.4	59.3	-	-	-	-	-	-
Zhang et al. (2016)		-	-	-	43.6	51.3	36.0	34.8	26.0	38.3

Table 4.2: Data sizes, monolingual experiments, and direct transfer experiments. Wiki size is the number of articles in Wikipedia. For monolingual experiments, we train the proposed model on the training data of the target languages. 'Wikifier only' uses the previous tags features also. For direct transfer experiments, all models are trained on CoNLL English training set. The rows marked Täckström come from Täckström et al. (2012), and are the baseline and clustering result. The plus signs (+) signify cumulative addition. EN: English, NL: Dutch, DE: German, ES: Spanish, TR: Turkish, TL: Tagalog, YO: Yoruba, BN: Bengali, TA: Tamil.

signals in direct transfer of a trained NER model across languages. Finally, we explore the importance of Wikipedia size to the quality of wikifier features and study the use of multiple source languages.

Datasets

We use data from CoNLL2002/2003 shared tasks Tjong Kim Sang (2002); Tjong Kim Sang and De Meulder (2003b). The 4 languages represented are English, German, Spanish, and Dutch, each annotated using the IOB1 labeling scheme, which we convert to the BIO labeling scheme. All training is on the train set, and testing is on the test set. The evaluation metric for all experiments is phrase level F1, as explained in Tjong Kim Sang (2002). In order to experiment on a broader range of languages, we also use data from the REFLEX Simpson et al. (2008) and LORELEI projects. From LORELEI, we use Turkish,[4] From REFLEX, we use Bengali, Tagalog, Tamil, and Yoruba.[5] While Turkish, Tagalog, and Yoruba each has a few non-Latin characters, Bengali and Tamil are with an entirely non-Latin script. This is a major reason for inclusion in

[4] LDC2014E115

[5] LDC2015E13,LDC2015E90,LDC2015E83,LDC2015E91

our experiments. We use the same set of test documents as used in Zhang et al. (2016). All other documents in the REFLEX and LORELEI packages are used as the training documents in our monolingual experiments. We refer to these five languages collectively as low-resource languages.

Besides PER, LOC, and ORG, some low-resource languages contain TIME tags and TTL tags, which represented titles in text, such as Secretary, President, or Minister. Since such words are not tagged in the CoNLL training data, we opted to simply remove these tags. On the other hand, there is no MISC tag in the low-resource languages. Instead, many MISC-tagged entities in the CoNLL datasets have LOC tags in the REFLEX and LORELEI packages, e.g., Italian and Chinese. We modify a MISC-tagged word to LOC tag if it is grounded to an entity with location as a FreeBase type, and remove all the other MISC tags in the training data. This process of changing MISC tags is only done when we train on CoNLL documents and test on low-resource languages.

The only requirement to build the cross-lingual wikifier model is a multilingual Wikipedia dump, and it can be trivially applied to all languages in Wikipedia. The top section of Table 4.2 lists Wikipedia sizes in terms of articles,[6] the number of titles linked to English titles, and the number of training and test mentions for each language.

Besides the English gazetteers used in Ratinov and Roth (2009), we collect gazetteers for each language using Wikipedia titles. A Wikipedia title is included in the list for person names if it contains FreeBase type person. Similarly, we also create a location list and an organization list for each language. The total number of names in the gazetteers of each language is listed in Table 4.2.

Monolingual Experiments

We begin by showing that wikifier features help when we train and test on the same language. The middle section of Table 4.2 shows these results.

In the ‘Wikifier only’ row, we use only wikifier features and previous tags features. This is intended to show the predictive power of wikifier features alone. Without using any lexical features, it gets good scores on the languages that have a large Wikipedia. These numbers represent the quality of the cross-lingual wikifier in that language, which in turn is correlated with the size of Wikipedia and size of the intersection with English Wikipedia.

The next row, ‘Base features’, shows that lexical features are always better than wikifier features only. This agrees with the common wisdom that lexical features are important for NER.

Adding gazetteers to the base features improves by more than 3 points for higher-resource languages.

[6] From `https://en.wikipedia.org/wiki/List_of_Wikipedias`, retrieved March 2016

This is because the low-resource languages have much smaller gazetteers which have lower coverage than other languages' gazetteers.

Finally, the '+Wikifier' row shows that our proposed features are valuable even in combination with strong features. It improves upon base features and gazetteer features for all 9 languages. These numbers may be less than state of the art because the features we use are designed for English, and may not capture lexical subtleties in every language. Nevertheless, they show that wikifier features have a non-trivial signal that has not been captured by other features.

Direct Transfer Experiments

We evaluate our direct transfer experiments by training on English and testing on the target language. The results from these experiments are shown in the bottom section of Table 4.2.

The 'Wikifier only' row shows that the wikifier features alone preserve a signal across languages. Interestingly, for both Bengali and Tamil, this is the strongest signal, and gets the highest score. If the lexical features are included when we train the English model, the learning algorithm will give them too much emphasis, thus decreasing the importance of the wikifier features. Since Bengali and Tamil use non-Latin scripts, no lexical feature in English will fire at test time. Thus, approaches that include base features perform poorly.

The results of 'Base features' can be viewed as a sort of language similarity to English, which, in this case, is related to lexical overlap and similarity between the scripts. Comparing to monolingual experiments, we can see that the lexical features become weak in the cross-lingual setting.

The gazetteer features are again shown to be very useful for almost all languages except Bengali and Tamil due to the reason explained in the monolingual experiment and to the inclusion of lexical features. For all other languages, the gain from adding gazetteers is even larger than it is in the monolingual setting.

For nearly every language, wikifier features help dramatically, which indicates that they are very good delexicalized features. Wikifier features add more than 10 points on Dutch, German, and Turkish.

The trend in Table 4.2 suggests the following strategy when we want to extract named entities in a new foreign language: It is better to include all features if the foreign language uses Latin script, since the names are likely to be mentioned similarly to the English names. Otherwise, using wikifier features only could be the best setting.

Täckström et al. (2012) also directly transfer an English NER model using the same setting as ours: train on the CoNLL English training set and predict on the test set of other three languages. We compare our baseline transfer model (Base Features) to the row denoted by "Täckström baseline". Even though we

FEATURES	SPANISH		GERMAN	
	#inter.	F1	#inter.	F1
Wikifier only	757K	43.82	964K	39.83
Wikifier−Freebase query	757K	34.69	964K	28.27
Wikifier−Freebase−50% intersection	379K	30.32	482K	27.24
Wikifier−Freebase−90% intersection	76K	29.44	96K	25.94

Table 4.3: The F1 scores of using only wikifier features with removing the support from FreeBase and varying the number of titles linked to the English Wikipedia. 'Wikifier−Freebase query' removes the component of querying FreeBase by the target language title from 'Wikifier only'. '$-X\%$ intersection' indicates removing $X\%$ of the inter-language links with English titles. The column #inter. shows the number of titles that intersect with English.

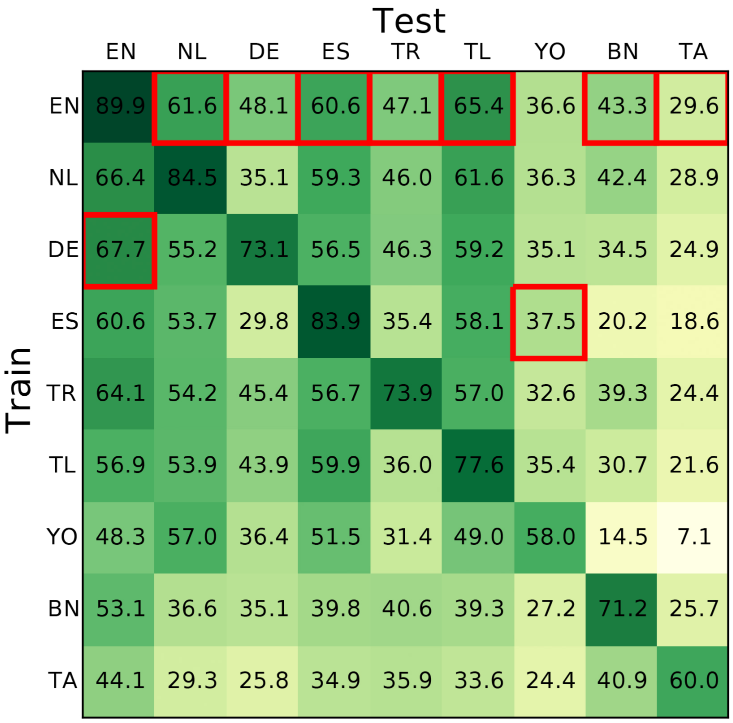
do not use gold POS tags, we see that our results are comparable. The second Täckström row uses parallel text to induce multilingual word clustering. While this approach is orthogonal to ours, and could be used in tandem to get even better scores, we compare against it for lack of a more closely aligned scenario. We see that for each language, our approach significantly outperforms their approach.

We note that our numbers are comparable to those reported for WIKI-2 in Nothman et al. (2012) for the CoNLL languages (with the exception of German, where their result is higher). However, they require language-specific heuristics to generate *silver-standard* training data from Wikipedia articles. What they gain for single languages, they likely lose in generalization to other languages. This approach is orthogonal to ours; we, too, can use their silver-standard data in training.

For the low-resource languages, we compare our direct transfer model with the expectation learning model proposed in Zhang et al. (2016). This model is not a direct transfer model, but it does not use any training data in the target languages either. Instead, for each target language, it generates patterns from parallel documents between English and the target language, a large monolingual corpus in the target language, and one-hour interaction with a native speaker of the target language. Note that they also use a cross-lingual wikifier, but only for refining the entity types. On the other hand, in our model, the features from the wikifier are used both in detecting entity mention boundaries and entity types. We can see that our approach performs better than their model on all five languages even though we assume much fewer resources. The difference is most significant on Turkish, Tagalog, and Bengali.

Quality of Wikifier Features

One immediate question is, why are wikifier features less helpful on the low-resource languages results than on the CoNLL languages? In this experiment, we show that smaller Wikipedia sizes result in worse Wikipedia features, which is the reason Yoruba has bad 'Wikifier only' results and then only small improvement from



Train \ Test	EN	NL	DE	ES	TR	TL	YO	BN	TA
EN	89.9	61.6	48.1	60.6	47.1	65.4	36.6	43.3	29.6
NL	66.4	84.5	35.1	59.3	46.0	61.6	36.3	42.4	28.9
DE	67.7	55.2	73.1	56.5	46.3	59.2	35.1	34.5	24.9
ES	60.6	53.7	29.8	83.9	35.4	58.1	37.5	20.2	18.6
TR	64.1	54.2	45.4	56.7	73.9	57.0	32.6	39.3	24.4
TL	56.9	53.9	43.9	59.9	36.0	77.6	35.4	30.7	21.6
YO	48.3	57.0	36.4	51.5	31.4	49.0	58.0	14.5	7.1
BN	53.1	36.6	35.1	39.8	40.6	39.3	27.2	71.2	25.7
TA	44.1	29.3	25.8	34.9	35.9	33.6	24.4	40.9	60.0

Figure 4.3: Different training/test language pairs. Scores shown are the F1 scores. The red boxes signify the best non-target training languages.

the wikifier features over base features.

The cross-lingual wikifier that we use in our system only grounds words to the intersection of the English and target language Wikipedia. Given a Wikipedia title in the target language, we first retrieve FreeBase IDs by querying the FreeBase API. If it fails, we find the corresponding English Wikipedia title via interlanguage links and then query the API with the English title. However, FreeBase does not contain entities in Yoruba, Bengali, and Tamil, so the first step will always fail for these three languages. We remove this step in the experiments of high-resource languages and the results are shown in the row 'W.−FB query ' of Table 4.3. We see that the performance drops significantly, because many words have no features from FreeBase types.

Next, we randomly remove 50% and 90% of the interlanguage links to English titles. This will not only reduce the number of fired features from Wikipedia categories, but also FreeBase types since English titles are used to query FreeBase IDs. When 90% of interlanguage links are removed, the scores of Spanish and German are closer to Yoruba's score (27.91).

Training Languages

In all previous experiments, the training language is always English. In order to test the efficacy of training with languages other than English, we create a train/test matrix with all combinations of languages, as seen

TRAINING LANG	TR	TL	YO	AVG
EN	47.12	65.44	36.65	49.74
EN+ES	44.85	66.61	37.57	49.68
EN+NL	48.34	66.09	36.87	50.43
EN+DE	49.47	64.10	35.14	49.57
EN+ES+NL+DE	49.00	66.37	**38.02**	51.13
ALL−Test Lang	**49.83**	**67.12**	37.56	**51.50**

Table 4.4: The F1 scores of the proposed direct transfer model on three low-resource languages using training data in multiple languages. The row "ALL−Test Lang" trains the model on all languages except the test language, Bengali, and Tamil. Bengali and Tamil are excluded since we use all features in this experiment.

in Figure 4.3.

The vertical axis represents training language, and the horizontal axis represents test language. A darker color signifies a higher score. For example, if we train on Spanish (ES) and test on Yoruba (YO), we get an F1 of 37.5. When the training or test language is Bengali (BN) or Tamil (TA), we only use wikifier features. For other settings, all features are included. Note that when the test language is one of the CoNLL languages (EN, NL, DE, ES) and the training language is a non-CoNLL language, we ignore all MISC tags in evaluation, since there is no MISC tag in the low-resource languages. The diagonals represent the monolingual setting in which we use all features for all languages. Since we are interested in transferring a model, we ignore the diagonals, and identify the best training language for a given test language as the largest off-diagonal in each column. These are demarcated with red boxes.

English is the best for most languages, with the only exception of Spanish being the best for Yoruba. It makes sense that high-resource languages are better training languages because 1) there are more annotated training instances, 2) larger Wikipedia creates denser wikifier features, therefore providing better estimation of the weights to these features.

Table 4.4 shows the results of training on multiple languages. We use all features in this experiment. The row "EN" only trains the model on the English training documents, and the results are identical to those shown in Table 4.2. Using all CoNLL languages (EN+ES+NL+DE) adds more than 1 point F1 in average comparing to using English only. Finally, training on all but the test languages further improves the results.

This experiment shows that we can augment training data from other languages' annotated documents. Although the performance only increases a little, it does not hurt most of the time.

Domain Adaptation

To improve the results of the monolingual experiments, we consider the domain adaptation setting where there is annotated data for both source and target domains. The question is whether training data from the

APPROACH	ES	NL	TR	TL	AVG
Target	83.87	84.49	73.86	77.64	79.96
Src+Tgt	**84.17**	**84.81**	**74.52**	**77.80**	**80.33**
FrustEasy	83.89	84.08	73.73	77.04	79.69

Table 4.5: The domain adaptation experiments. The source domain (English) training examples are used to improve the monolingual baseline model (*Target*) which is only trained on the target domain (Spanish, Dutch, Turkish, and Tagalog) training data. The numbers are the phrase-level F1 scores.

source domain can improve a model that is trained solely on the target-domain data. In this experiment, we use English as the source domain, and use Spanish, Dutch, Turkish, and Tagalog as four different target domains. We compare three approaches:

- **Target:** only uses the training data in the target domain. This is the setting of the monolingual experiments in Table 4.2.
- **Src+Tgt:** directly uses the training data from both source and target domains. This method is identical to the setting in our previous multi-source direct transfer experiments.
- **FrustEasy:** the "Frustratingly Easy" adaptation framework proposed by Daumé (2007).

All types of features are used in all settings. The results are shown in Table 4.5. We can see that although *Src+Tgt* is always the best approach, the improvement over the baseline, *Target*, is tiny. Interestingly, the *FrustEasy* framework does not help for most languages. This result is consistent with the analysis and observation in Chang et al. (2010a) that 1) when the source and target domains are very different, the baseline approach (*Target*) is very strong, and 2) when there are cross-domain clustering features (e.g., the wikifier features), *Src+Tgt* is better than *FrustEasy*. To further improve the monolingual baselines via adaptation from other languages, better cross-lingual or language-independent information may be needed.

4.1.3 Related Work

There are three main branches of work for extending NLP systems to many languages: projection across parallel data, Wikipedia-based approaches, and direct transfer. Projection and direct transfer take advantage of the success of NLP tools on high-resource languages. Wikipedia-based approaches exploit the fact that, by editing Wikipedia, thousands of people have made annotations in hundreds of languages.

Projection

Projection methods take a parallel corpus between source and target languages, annotate the source side, and push annotations across learned alignment edges. Assuming that source side annotations are of high quality, success depends largely on the quality of the alignments, which depends, in turn, on the size of the parallel data, and the difficulty of aligning with the target language.

There is work on projection for POS tagging Yarowsky et al. (2001); Das and Petrov (2011); Duong et al. (2014), NER Wang and Manning (2014); Kim et al. (2012); Ehrmann et al. (2011), and parsing Hwa et al. (2005); McDonald et al. (2011).

Wang and Manning (2014) show that projecting expectations of labels instead of hard labels can improve results. They experiment in two different settings: weakly-supervised, where only parallel data is available, and semi-supervised, where annotated training data is available along with unlabeled parallel data.

Using Wikipedia

Wikipedia has been used for a large number of NLP tasks, from use as a semantic space Gabrilovich and Markovitch (2007); Chang et al. (2008a); Song and Roth (2014), to generating parallel data Smith et al. (2010), to use in open information extraction Wu and Weld (2010). It has also been used to extract training data for NER, under the intuition that Wikipedia is already (partially) annotated with NER labels, in the form of links to pages. Nothman et al. (2012) generate *silver-standard* NER data from Wikipedia using link targets, and other heuristics. This can be gathered for any language in Wikipedia, but several of the heuristics depend on language-specific rules. Al-Rfou et al. (2015) generate training data from Wikipedia articles using a similar manner. The polyglot word embeddings Al-Rfou et al. (2013) are used as features in their NER model. Although the features are delexicalized, the embeddings are unique to each language, and so the model cannot transfer.

Kim et al. (2012) use Wikipedia to generate parallel sentences with NE annotations. They propose a semi-CRF model for aligning entities in parallel sentences. Results are very strong on Wikipedia data. This is a hybrid approach in that it is supervised projection using Wikipedia.

Our work is most closely related to Kazama and Torisawa (2007). They do NER using Wikipedia category features for each mention. However, their method for wikifying text is not robust to ambiguity, and they only do monolingual NER.

Sil and Yates (2013) create a joint model for NER and entity linking in English. They avoid the traditional pipeline by overgenerating mentions in the first stage and using NER features to rank candidates. While the results are promising, the model is not scalable to other languages because it requires both a trained NER and a NP chunker.

Direct Transfer

In direct transfer once trains a model in a high-resource setting using delexicalized features, that is, features that do not depend on word forms, and then directly applies it to text in a new language.

Täckström et al. (2012) experimented with direct transfer of dependency parsing and NER, and showed that using word cluster features can help, especially if the clusters are forced to conform across languages. The cross-lingual word clusters were induced using large parallel corpora.

Building on this work, Täckström (2012) focuses solely on NER, and includes experiments on self-training and multi-source transfer for NER. Their experiments are orthogonal to ours, and could be combined nicely. This work is closest to ours in terms of method, and therefore we compare against it in our experiments.

Our work falls under the umbrella of direct transfer methods combined with the use of Wikipedia. We introduce wikifier features, which are truly delexicalized, and use Wikipedia as a source of information for each language.

4.1.4 Conclusion

We propose a language-independent model for cross-lingual NER building on a cross-lingual wikifier. This model works on all languages in Wikipedia and the only requirement is a Wikipedia dump. We study a wide range of languages in both the monolingual and the cross-lingual settings, and show significant improvements over strong baselines. An analysis shows that the quality of the wikifier features depends on the Wikipedia size of the test language.

This work shows that if we can disambiguate words and phrases to the English Wikipedia, the typing information from Wikipedia categories and FreeBase are useful language-independent features for NER. However, there is additional information in Wikipedia that could be helpful and which we do not use, including words in the documents and relations between titles; this would require additional research.

In the future, we would like to experiment with combining our method with other techniques for multilingual NER (Section 4.1.3), including parallel projection and the automatic generation of training data from Wikipedia.

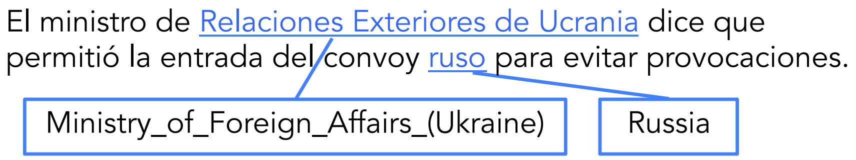

El ministro de Relaciones Exteriores de Ucrania dice que
permitió la entrada del convoy ruso para evitar provocaciones.
Ministry_of_Foreign_Affairs_(Ukraine)
Russia

Figure 4.4: An example of cross-lingual wikification. The Spanish mention "ruso" is grounded to the English Wikipedia title "Russia", and "Relaciones Exteriores de Ucrania" is grounded to the title "Ministry_of_Foreign_Affairs_(Ukraine)".

4.2 Learning Better Name Translation for Cross-Lingual Wikification

One of the key challenges for wikification is the *candidate generation* step – generating title candidates given a mention. Since there are millions of entries in the English Wikipedia, this step aims at quickly producing a manageable number of title candidates, so that a more sophisticated algorithm can be applied to rank them. The candidate generation step is typically done by indexing titles in Wikipedia using strings that could be used to refer to the titles.

In the cross-lingual setting, this problem becomes more challenging. There are two intuitive ways to retrieve English title candidates given a foreign mention: 1) Querying the English titles' index directly using a foreign mention. 2) Querying the foreign language titles' index using the foreign mention, and then converting the foreign titles to English using the inter-language links in Wikipedia. The first approach only works if the target language is very close to English, so that names in the two languages are expressed almost identically. The second approach depends heavily on the size of the foreign language Wikipedia. That is, this approach only works if the target entity exists in the foreign language Wikipedia, and there is a link pointing to the corresponding English page. In Figure 4.4, the first approach works for none of the two mentions since they are expressed differently in English. The second approach will work on the mention "ruso", because we can get the target entity Russia in the Spanish Wikipedia based on the given mention "ruso", and further reach Russia's English page via the inter-language links in Wikipedia. However, since Ukraine's Ministry of Foreign Affair does not exist in the Spanish Wikipedia, this approach will fail to find the correct English title for another mention "Relaciones Exteriores de Ucrania". Table 4.6 shows an upper bound on the coverage of the second approach. These numbers are estimated from the anchor texts in Wikipedia articles. For example, only 20.82% of the Bengali mentions are linked to English titles. In our experiments, we show that our model can retrieve the target English title for 65.18% of the Bengali mentions.

In this section, we focus on cases which cannot be addressed by these two approaches. One solution to this

	Current upper bound	Proposed method
Spanish	40.44%	64.62%
Turkish	35.06%	63.47%
Bengali	20.82%	65.18%
Tagalog	16.23%	73.23%

Table 4.6: Improved title candidate generation coverage. The left column gives the fraction of Wikipedia titles that are covered by the inter-language links. This is the *upper bound for current cross-lingual wikification methods*. The right column gives the fraction of these mentions that have the correct English title in the candidate set using the method developed in this section.

problem is to use a transliteration or translation model. We can translate foreign names into English and then use it to query the index of the English titles. However, the use of standard transliteration and translation models in this context suffers from two problems. First, the traditional setting of transliteration focuses only on single-token names of people or locations, but for wikification, the entities that we want to ground are often longer (e.g., names of organizations). Since multi-token names of locations and organizations typically require a mixture of translation and transliteration, they are excluded from "standard" transliteration studies. Second, the transliteration models are usually learned from word pairs, which could be manually created or mined from large amounts of parallel text. These are also required to train machine translation models. However, with the goal of solving cross-lingual wikification for all languages in Wikipedia, including many low resource languages, we cannot assume large amounts of high quality parallel data.

We propose a probabilistic model that is used to learn name translation from Wikipedia title pairs. Using the inter-language links in Wikipedia, we can obtain foreign-to-English title pairs for different types of entities and for all languages in Wikipedia. Since we learn from phrase pairs rather than word pairs, we extend a transliteration model to jointly model word alignment and word-to-word transliteration. It is clear that if we can align words in a phrase pair well, we can learn word transliteration better. On the other hand, a good transliteration model can help to improve word alignment performance, because the transliteration model may provide better word generation probability if the word pair appears infrequently in the training data, but the sub-words pairs in the word pair are frequent enough.

We compare the proposed model with six strong approaches from the literature, including 4 transliteration models and 2 character-based neural machine translation models. When these models are trained on Wikipedia title pairs of 8 languages, We show that our model outperforms these approaches not only on the standard string similarity-based metric, but also on the candidate generation performance of cross-lingual wikification. Finally, we show that our model improves an end-to-end cross-lingual wikification system on the TAC 2016 EDL dataset.

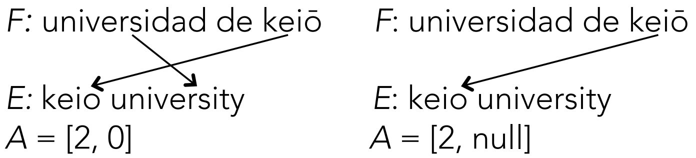


Figure 4.5: Examples of the word alignment variable A in Eq. (4.1).

4.2.1 The Joint Model

In this section, we present our model for learning name translation from Wikipedia title pairs.

Given a title pair (F, E), where F is the foreign title and E is the target English title, let the number of words in F and E be m and l respectively, we model the title generation probability as

$$P(F, A|E, m) = P(A|m) \prod_{(f,e)\in A} P(f|e), \tag{4.1}$$

where A is an alignment assignment of words $f \in F$ and $e \in E$. The alignment A is a list of size $|E| = l$, where $A[j]$ could either be null, or the index of the word in F which is aligned with the j-th target word. Figure 4.5 shows two examples. Given a Spanish-English title pair ("universidad de keiō", "keio university"), the word alignments variable A in the left example is $[2, 0]$. The 2 in the first position means that "keio" is aligned with "keiō", and the 0 indicates "university" is aligned with "universidad". In the right example, since the word "university" is not aligned with any source word, the second element in A becomes *null*. In order to reduce the number of possible A, we assume that a source word (in F) can be aligned with up to one target word, and no two source words can be aligned with the same target word. Note that in contrast to word alignment models in machine translation which usually have independence assumption between words, our model jointly determines the alignment of all words in the pair of strings. Training and inference in our model are tractable since the number of words in a title is usually small.

The last term of Eq. (4.1) is the word generation probability given the word alignment, where $(f, e) \in A$ is a word pair according to the alignment A. In the left example of Figure 4.5, there are two word pairs, (universidad, university) and (keiō, keio), therefore $\prod_{(f,e)\in A} P(f|e) = P(\text{universidad}|\text{university})P(\text{keiō}|\text{keio})$. This is where we use a transliteration model to better estimate the word generation probability. We adopt the model proposed by Pasternack and Roth (2009) in which the word generation probability is modeled as

$$P(f, a|e) = \prod_{(u,v)\in a} P(v|u), \tag{4.2}$$

where a indicates sub-word alignment between the foreign word f and the English word e. For instance, given $(f, e) = (\text{keiō}, \text{keio})$, one possible sub-word alignment is (ke-iō, ke-io), namely, "ke" is aligned with "ke" and "iō" is aligned with "io", therefore $P(f, a|e) = P(\text{ke}|\text{ke})P(\text{iō}|\text{io})$. This model will try all sub-word alignments that segment both source and target words into the same number of sub-words. Using Eq. (4.2), we can compute the word generation probability by:

$$P(f|e) = \sum_{a} \prod_{(u,v)\in a} P(v|u), \tag{4.3}$$

which sums over all possible sub-word alignments between the word pair (f, e). Combining Eq. (4.1) and (4.2), we have our final likelihood:

$$\begin{aligned} &P(F, A, a_A|E, m) \\ =&P(A|m) \prod_{(f,e)\in A} \prod_{(u,v)\in a_f^e} P(v|u). \end{aligned} \tag{4.4}$$

We use a_A to represent sub-word alignments of all word pairs according to the word alignment A, and use a_f^e to represent the sub-word alignments between a particular word pair (f, e).

To summarize, given a title pair (F, E), our model contains latent variable A which indicates word alignments between the two titles and a_A which describes sub-word alignments. We use the EM algorithm (Dempster et al., 1977) to maximize the likelihood of training pairs, and update the parameters $P(A|m)$ and $P(v|u)$ iteratively.

After training the model, we can generate the target English phrase given a foreign phrase F using Bayes rule:

$$\begin{aligned} E^* &= \arg\max_E P(E|F) = \arg\max_E \frac{P(F|E)P(E)}{P(F)} \\ &= \arg\max_E \sum_A \sum_{a_A} P(F, A, a_A|E, m)P(E), \end{aligned}$$

where $P(F, A, a_A|E)$ is from Eq. (4.4) and $P(E)$ is obtained from an English language model.

Expectation Maximization

Given training title pairs (F_i, E_i), $i = 1, \cdots, n$, the expected log likelihood is

$$\begin{aligned} & E_A E_{a_A}[\sum_{i=1}^{n}(\log P(A_i|m_i) + \sum_{(f,e)\in A_i} \sum_{(u,v)\in a_f^e} \log P(v|u))] \\ =& E_A[\sum_{i=1}^{n} \log P(A_i|m_i)] + E_A E_{a_A}[\sum_{i=1}^{n} \sum_{(f,e)\in A_i} \sum_{(u,v)\in a_f^e} \log P(v|u)] \end{aligned} \tag{4.5}$$

The expectation with respect to word alignment A and sub-word alignment a_A are computed using the current parameters, which are updated in the previous iteration. The first term of Eq. (4.5) can be expanded as:

$$\begin{aligned} & E_A[\sum_{i=1}^{n} \log P(A_i|m_i)] \\ =& \sum_{i=1}^{n} \sum_{A_i} P(A_i|F_i, E_i) \log P(A_i|m_i), \end{aligned}$$

Adding the constraints that $\sum_A P(A|m) = 1, \forall m$ using the Lagrangian multipliers method, we have the following objective

$$\sum_{i=1}^{n} \sum_{A_i} P(A_i|F_i, E_i) \log P(A_i|m_i) - \sum_{m} \alpha_m (\sum_{A} P(A|m) - 1).$$

To maximize this function, we take the partial derivative with respect to $P(\bar{A}|\bar{m})$, a particular word alignments $\bar{A}$ given $\bar{m}$ source tokens, and set the result to 0.

$$\sum_{i:|F_i|=\bar{m}} \frac{P(\bar{A}|F_i, E_i)}{P(\bar{A}|\bar{m})} - \alpha_{\bar{m}} = 0$$

The update rule of $P(\bar{A}|\bar{m})$ is

$$P(\bar{A}|\bar{m}) = \frac{\sum_{i:|F_i|=\bar{m}} P(\bar{A}|F_i, E_i)}{\sum_{i:|F_i|=\bar{m}} \sum_{A} P(A|F_i, E_i)}, \tag{4.6}$$

Where $P(A|F_i, E_i)$ can be computed from the current parameters:

$$\begin{aligned}
P(A|F_i, E_i) &= \frac{P(A, F_i|E_i)}{P(F_i|E_i)} \\
&= \frac{\sum_{a_A} P(A, F_i, a_A|E_i, m_i)}{\sum_A \sum_{a_A} P(A, F_i, a_A|E_i, m_i)} \\
&= \frac{P(A|m_i) \prod_{(f,e)\in A} \sum_{a_f^e} \prod_{(u,v)\in a_f^e} P(v|u)}{\sum_A P(A|m_i) \prod_{(f,e)\in A} \sum_{a_f^e} \prod_{(u,v)\in a_f^e} P(v|u)} \\
&= \frac{P(A|m_i) \prod_{(f,e)\in A} P(f|e)}{\sum_A P(A|m_i) \prod_{(f,e)\in A} P(f|e)}
\end{aligned}$$

The last equation is derived by using

$$P(f|e) = \sum_a \prod_{(u,v)\in a} P(v|u).$$

For the second term in Eq. (4.5),

$$\begin{aligned}
&E_A E_{a_A}[\sum_{i=1}^{n} \sum_{(f,e)\in A_i} \sum_{(u,v)\in a_f^e} \log P(v|u)] \\
=&\sum_{i=1}^{n} \sum_{A_i} P(A_i|F_i, E_i) \times \sum_{(f,e)\in A_i} \sum_{a_f^e} P(a_f^e) \sum_{(u,v)\in a_f^e} \log P(v|u).
\end{aligned}$$

Adding the constraints that $\sum_v P(v|u) = 1, \forall u$ using the Lagrangian multiplier method, we obtain

$$\sum_{i=1}^{n} \sum_{A_i} P(A_i|F_i, E_i) \times \sum_{(f,e)\in A_i} \sum_{a_f^e} P(a_f^e) \times \sum_{(u,v)\in a_f^e} \log P(v|u) - \sum_u \beta_u (\sum_v P(v|u) - 1).$$

To maximize this objective, we take the partial derivative with respect to the generation probability of a particular pair of sub-words $P(\bar{v}|\bar{u})$, and set the result to 0.

$$\sum_{i=1}^{n} \sum_{A_i} P(A_i|F_i, E_i) \sum_{a_f^e \in A_i} \frac{n_{\bar{u},\bar{v}|a_f^e} P(a_f^e)}{P(\bar{v}|\bar{u})} - \beta_u = 0,$$

where $n_{\bar{u},\bar{v}|a_f^e}$ is number of times the sub-word $\bar{v}$ is aligned with $\bar{u}$ under the word alignment A_i and the sub-word alignment a_f^e. The update rule of $P(\bar{v}|\bar{u})$ becomes

$$P(\bar{v}|\bar{u}) = \sum_{i=1}^{n} \sum_{A_i} P(A_i|F_i, E_i) \sum_{a_f^e \in A_i} \frac{n_{\bar{u},\bar{v}|a_f^e}}{n_{\bar{u},*|a_f^e}} P(a_f^e), \tag{4.7}$$

where $n_{\bar{u},*|a_f^e}$ is the number of times the sub-word $\bar{u}$ occurs in any target word under the word alignment A_i and the sub-word alignment a_f^e. The term $P(a_f^e)$ (the shorthand for $P(f, a|e)$) can be obtained from the current parameters using Eq. (4.2).

We have derived the update rules Eq. (4.6) and (4.7) for the parameters in our model (Eq. (4.4)).

Note that for the frequent words in the training data, we memorize their translation by taking the most probable alignment in each iteration. These word pairs are excluded in updating sub-word generation probabilities (Eq. (4.7)), since these words are usually translated instead of transliterated. More specifically, when we iterate through word pairs in the third summation of Eq. (4.7), we simply skip the frequent word pairs. For example, in Turkish, "ili" means prefecture and "adaları" means islands. Since the sub-word alignments of these word pairs are very different from the words that are transliterated, including these word pairs in Eq. (4.7) may result in worse estimation of sub-word generation probabilities.

4.2.2 Experiments

We compare our model with six other approaches. The first four approaches are the standard transliteration models which are designed to learn from transliteration word pairs:

- **DirecTL+** (Jiampojamarn et al., 2008) is a discriminative string transduction tool, which was successfully applied to transliteration in the NEWS shared tasks. Given sub-word aligned word pairs, DirecTL+ views transliteration problem as a sequence tagging problem. We use m2m-aligner (Jiampojamarn et al., 2007) to segment and align the input word pairs.

- **Sequitur** (Bisani and Ney, 2008) is a probabilistic model for grapheme-to-phoneme conversion. Unlike DirecTL+, which requires sub-word alignment, Sequitur directly trains a joint n-gram model from unaligned word pairs. Higher order n-gram models are trained iteratively from lower order models. We train up to 5-gram models.

- **P&R** (Pasternack and Roth, 2009) is the model which our model is based on. The probability of the source word given the target word is modeled as in Eq. (4.2), where sub-word alignments are described by the latent variable a.

- **JANUS** (Liu et al., 2016) trains a character-based left-to-right and a right-to-left LSTM model on the input word pairs. The prediction is based on the agreement between the outputs of these two models. We use 500 dimensional embeddings and 100 training epochs.

We also compare with the following two character-level neural machine translation models:

- **NMT-bpe** (Chung et al., 2016) segments the source words into sub-words using byte pair encoding (Sennrich et al., 2015), and encodes these sub-words by gated recurrent units (GRUs). When decoding, a newly designed character-level bi-scale recurrent neural network is applied.
- **NMT-char** [7] is inspired by NMT-bpe. This model not only decodes at character-level, but also encodes the source side at character-level. When encoding, the model applies a series of convolutional, pooling, and highway layers. The results are fed into a bi-directional GRU. At the decoding stage, a single feed-forward neural network is used to compute attention scores of every source segments. A two-layer character-level decoder is applied to predict the target characters.

For the methods which are designed for transliterating word pairs, we apply two word alignment methods to make word pairs from the title pairs.

- **p-align** takes title pairs that contain same number of words on each side, and aligns the words by their position. That is, the *i*-th source word is aligned to the *i*-th word of the target phrase.
- **f-align** applies a word alignment (Dyer et al., 2013) model on the training title pairs to produce word pairs.

At test time, after translating each word in the test phrase, we apply a bigram language model to reorder the predicted words. The language model is trained on all articles in the English Wikipedia.

Name Translation Performance

The first experiment evaluates the performance of each model by a standard transliteration metric: fuzzy F1 of the top-1 prediction. This metric is based on the longest common subsequence between the gold and generated names, and is used in several years of NEWS transliteration workshops (Li et al., 2009, 2010; Banchs et al., 2015).

We create training, development, and test title pairs from the inter-language links in Wikipedia. For a test language L, we take all the titles in L's Wikipedia which have a link pointing to the corresponding English page, and then use FreeBase types to classify them into one of the three entity types: person, location, and organization, or discard a title if it is non of the three types. Since different types of entities may be translated differently, we find that it is better to train a model for each entity type separately. For each entity type, we take at most 10k pairs for training and 5k pairs for both development and test. The numbers of title pairs for each language are shown in the column "#Title Pairs" of Table 4.7.

[7] `https://github.com/nyu-dl/dl4mt-c2c`

		#Title Pairs			#Mentions
Lang.	Type	Train	Dev	Test	
ES	LOC	10,000	5,000	5,000	4,953
	ORG	4,120	1,640	2,471	1,311
	PER	10,000	5,000	5,000	2,094
TR	LOC	7,738	3,084	4,644	2,569
	ORG	1,556	642	962	1,539
	PER	3,646	1,451	2,181	2,011
TL	LOC	1,538	609	903	931
	ORG	132	48	73	117
	PER	845	334	501	282
BN	LOC	3,151	1,262	1,893	2,229
	ORG	634	248	379	337
	PER	3,999	1,597	2,388	1,684
HE	LOC	8,861	3,557	5,000	5,891
	ORG	2,909	1,131	1,747	2,996
	PER	10,000	5,000	5,000	9,768
FR	LOC	10,000	5,000	5,000	5,271
	ORG	6,318	2,517	3,739	1,805
	PER	10,000	5,000	5,000	2,305
IT	LOC	10,000	5,000	5,000	4,405
	ORG	3,861	1,502	2,302	1,423
	PER	10,000	5,000	5,000	2,702
AR	LOC	10,000	5,000	5,000	4,743
	ORG	4,820	1,920	2,894	1,051
	PER	10,000	5,000	5,000	2,796

Table 4.7: Statistics of the data we use in Section 4.2.2 and 4.2.2. ES: Spanish, TR: Turkish, TL: Tagalog, BN: Bengali, HE: Hebrew, FR: French, IT: Italian, AR: Arabic.

The results are listed in Table 4.8. The last block of rows shows the average performance of all 8 languages. The bold-faced numbers are the highest numbers of each row, and the underlined numbers are the second highest.

For the four transliteration models, using a word alignment model (f-align) to preprocess title pairs is better than aligning words according to their position in the titles (p-align). However, for person names, sometimes the performance of using f-align is worse than using p-align. Since there is not much word reordering in person names across languages, using f-align may create more incorrect training word pairs than using p-align.

Liu et al. (2016) report JANUS performs very well on transliterating between Japanese and English names, but it is not as strong on our dataset. The reason could be that this model is not so robust with noisy input, since it is designed for training on clean word pairs.

For the neural machine translation methods, the full character-based model (NMT-char) performs much

		DirecTL+		Sequitur		P&R		JANUS		NMT	NMT	Ours
		p-align	f-align	p-align	f-align	p-align	f-align	p-align	f-align	-bpe	-char	
ES	LOC	50.85	61.29	52.28	64.59	52.41	65.76	48.14	55.84	73.24	**73.56**	71.72
	ORG	56.35	62.14	61.57	67.46	61.69	68.23	55.09	60.13	58.75	61.27	**73.51†**
	PER	80.61	80.39	80.37	80.86	81.34	81.61	78.84	77.03	67.87	75.81	**81.85†**
	Avg	63.87	69.12	65.38	71.68	65.85	72.60	61.83	65.19	68.22	72.03	**76.14†**
TR	LOC	74.22	73.56	79.15	78.12	78.93	78.72	75.61	73.28	68.42	70.87	**80.33†**
	ORG	72.95	72.92	74.88	74.37	75.32	75.23	68.61	65.61	59.33	59.97	**76.34†**
	PER	80.51	80.09	80.03	79.69	81.30	81.42	76.01	75.37	60.10	66.10	**81.59†**
	Avg	75.83	75.31	78.87	78.09	79.15	79.05	74.85	72.92	64.97	68.19	**80.19†**
TL	LOC	64.10	62.17	64.26	65.06	65.76	66.65	57.86	57.25	56.84	59.63	**74.41†**
	ORG	54.79	53.76	57.02	56.25	55.78	56.54	55.81	55.60	55.89	56.49	**71.63†**
	PER	84.24	81.97	83.36	82.66	83.37	82.78	77.30	74.51	61.37	64.09	**85.88†**
	Avg	70.47	68.47	70.38	70.59	71.24	71.62	64.35	63.02	58.33	60.99	**78.16†**
BN	LOC	80.70	79.69	89.69	89.10	89.20	89.15	86.45	83.88	68.26	72.02	**90.02†**
	ORG	75.71	76.38	85.93	85.14	84.93	84.39	79.28	76.86	57.35	57.52	**86.37**
	PER	84.50	85.22	90.49	90.13	89.86	89.74	88.73	88.32	70.43	73.45	**90.87†**
	Avg	82.24	82.26	89.80	89.31	89.19	89.07	87.04	85.59	68.48	71.57	**90.16†**
HE	LOC	66.21	65.38	68.84	69.52	66.71	67.78	64.01	62.71	60.96	61.46	**71.20†**
	ORG	63.12	62.64	64.73	65.03	63.43	65.05	56.98	56.87	54.57	56.22	**68.02†**
	PER	77.61	79.43	**88.08†**	87.88	86.68	86.51	87.60	84.75	77.77	80.00	87.59
	Avg	70.60	70.95	76.42	76.66	74.72	75.35	73.01	71.22	67.17	68.57	**77.70†**
FR	LOC	54.37	59.56	52.22	62.13	57.11	63.96	46.11	54.68	69.79	**71.15†**	70.25
	ORG	58.15	62.46	61.85	67.51	65.64	68.78	54.94	62.80	60.89	65.46	**71.73†**
	PER	81.34	80.73	81.21	81.09	82.22	82.23	77.03	80.54	66.60	73.50	**82.42†**
	Avg	65.21	68.05	65.40	70.49	68.57	71.92	59.77	66.30	66.21	70.46	**75.08†**
IT	LOC	55.37	61.12	55.57	63.90	55.57	64.92	45.88	60.61	72.79	**74.12†**	73.45
	ORG	58.96	62.96	62.45	67.77	63.98	68.65	54.22	59.46	56.61	59.38	**71.09†**
	PER	80.26	80.10	80.06	80.27	81.09	81.16	78.72	78.87	67.54	76.71	**81.80†**
	Avg	66.16	69.18	66.81	71.28	67.51	72.22	60.79	67.82	67.63	72.41	**76.40†**
AR	LOC	65.16	65.18	68.44	68.07	66.86	68.14	63.34	64.51	64.22	66.19	**69.78†**
	ORG	58.34	60.95	64.67	68.19	61.80	66.70	58.50	62.84	57.81	59.86	**69.54†**
	PER	81.03	81.28	87.54	87.35	86.68	86.51	87.25	86.38	76.35	81.21	**87.56**
	Avg	69.78	70.47	75.00	75.57	73.41	74.94	71.53	72.61	67.49	70.59	**76.62†**
Avg	LOC	63.87	65.99	66.31	70.06	66.57	70.64	60.92	64.09	66.81	68.62	**75.14†**
	ORG	62.30	64.28	66.64	68.97	66.57	69.20	60.43	62.52	57.65	59.52	**73.53†**
	PER	81.26	81.15	83.89	83.74	84.07	84.00	81.44	80.72	68.50	73.86	**84.94†**
	Avg	70.52	71.73	73.51	75.46	73.70	75.85	69.15	70.58	66.06	69.35	**78.81†**

Table 4.8: Wikipedia title translation results. Given a Wikipedia title in a foreign language, we translate it into English using various models. The numbers are fuzzy F1 scores between the top-1 translation and the gold English title. The highest number of each row is bold-faced and the second highest is underlined. A bold-faced number with a dagger indicates the difference between it and the runner-up is statistically significant. We use approximate randomization (Noreen, 1989) with p-value < 0.05.

better than NMT-bpe which only decodes at the character-level. NMT models tend to perform better on European languages which the models are developed on and have more training pairs. From their low performance on lower-resource languages (TL and BN), it can be concluded that they may need more training data in order to generalize better. We have tried to use a state-of-the-art NMT-char model which is trained on a MT corpus, but the performance is worse than using the model trained on our name pairs.

Our model outperforms all other approaches in most cases, especially on LOC and ORG where word

		DirecTL+		Sequitur		P&R		JANUS		NMT	NMT	Ours
		P-align	f-align	p-align	f-align	p-align	f-align	p-align	f-align	-bpe	-char	
ES	LOC	4.93	27.05	20.51	37.03	17.34	61.72	2.40	15.06	26.43	27.86	**65.90†**
	ORG	8.39	22.43	21.28	31.05	34.25	58.05	5.34	13.20	7.40	7.70	**61.25†**
	PER	31.18	31.66	35.24	35.24	63.80	**63.94**	30.28	28.99	11.22	23.26	63.71
	Avg	12.05	27.48	24.32	35.64	31.63	61.70	9.85	18.26	19.63	23.55	**64.62†**
TR	LOC	49.94	47.92	51.50	50.60	60.65	61.35	38.07	42.62	21.29	24.45	**63.60†**
	ORG	43.92	41.33	39.18	39.64	56.27	57.76	9.23	10.98	3.96	4.35	**61.99†**
	PER	33.71	33.71	31.92	33.76	62.31	**64.69**	23.92	25.61	6.02	7.26	63.85
	Avg	43.10	41.59	41.97	42.31	60.09	61.55	26.16	29.07	11.91	13.74	**63.28†**
TL	LOC	38.45	30.61	42.53	43.39	68.42	71.00	13.96	17.51	8.27	4.73	**78.09†**
	ORG	0.85	4.27	12.82	8.55	11.97	19.66	0.00	2.56	1.71	1.71	**35.04†**
	PER	32.98	33.69	35.11	33.33	64.89	67.73	15.60	18.44	5.67	6.74	**73.05†**
	Avg	33.98	28.95	38.35	38.20	62.71	65.79	13.08	16.39	7.14	4.89	**73.23†**
BN	LOC	39.03	36.11	64.87	64.87	66.67	66.67	51.77	48.77	18.26	19.25	**68.37†**
	ORG	18.69	17.21	39.76	39.47	54.90	**56.08**	24.93	24.93	9.20	5.64	54.60
	PER	24.58	25.71	39.31	39.61	62.29	61.70	35.04	37.05	15.62	17.46	**63.66†**
	Avg	31.69	30.49	52.75	52.85	64.00	63.86	43.01	42.24	16.49	17.46	**65.41†**
HE	LOC	11.54	10.25	23.60	27.75	27.28	32.54	16.28	17.64	11.19	11.17	**42.32†**
	ORG	6.78	4.91	14.85	18.86	20.09	28.81	7.51	12.45	5.94	5.61	**36.95†**
	PER	6.70	6.82	25.66	26.10	47.85	49.09	23.88	21.71	25.57	25.26	**50.60†**
	Avg	8.24	7.60	23.27	25.46	36.90	40.61	18.85	18.94	17.88	17.65	**45.66†**
FR	LOC	16.35	30.73	24.78	39.50	39.39	63.33	2.49	22.82	20.72	22.75	**64.24†**
	ORG	13.68	27.92	24.60	36.23	48.31	61.94	5.26	21.88	11.41	13.07	**64.10†**
	PER	37.96	37.27	39.13	40.39	65.64	**66.46**	29.20	37.18	10.54	20.87	65.77
	Avg	21.15	31.80	28.27	39.09	47.55	63.83	9.58	26.17	16.43	20.42	**64.59†**
IT	LOC	11.06	27.83	24.36	41.00	25.24	62.29	2.29	32.85	26.88	30.85	**67.51†**
	ORG	24.31	39.63	29.52	43.85	46.31	63.25	4.08	20.31	6.04	4.01	**69.08†**
	PER	36.27	35.31	39.93	39.56	61.40	59.88	33.49	35.05	10.99	25.39	**63.06†**
	Avg	21.25	32.17	30.15	41.02	40.21	61.69	12.47	31.45	18.37	24.64	**66.37†**
AR	LOC	13.92	14.34	18.09	22.35	22.14	27.26	10.58	16.21	7.82	8.92	**31.86†**
	ORG	3.52	5.14	13.42	21.12	21.31	36.16	8.18	16.37	15.13	12.08	**40.72†**
	PER	8.91	9.66	24.50	25.61	42.92	44.81	24.07	23.86	20.71	25.43	**45.06**
	Avg	11.01	11.69	19.60	23.26	28.80	34.06	14.68	18.72	12.91	14.68	**37.24†**
Avg	LOC	23.15	28.11	33.78	40.81	40.89	55.77	17.23	26.69	17.61	18.75	**60.24†**
	ORG	15.02	20.36	24.43	29.85	36.68	47.71	8.07	15.34	7.60	6.77	**52.97†**
	PER	26.54	26.73	33.85	34.20	58.89	59.79	26.93	28.49	13.29	18.96	**61.09†**
	Avg	22.81	26.47	32.34	37.23	46.49	56.64	18.46	25.15	15.09	17.13	**60.05†**

Table 4.9: Wikipedia title candidate generation experiment. Given a mention, we translate it into English using different models, and then a candidate generation algorithm is applied to the translated names. The numbers indicate percentage of mentions that have the gold English title in the candidate set.

alignment is required. P&R with f-align often gets the second highest numbers, and Sequitur is usually slightly worse than P&R.

Candidate Generation Performance

The fuzzy F1 score in the previous section evaluates string similarity between the predicted name and the gold translation. It does not directly show the ability of retrieving the target English title given a foreign mention. In this experiment, we use the translated English names to generate English title candidates, and evaluate how often a model can produce the correct English title in the candidate set.

Following the way that Tsai and Roth (2016b) create a dataset for cross-lingual wikification, we use articles in Wikipedia to make a dataset which only contains named entity mentions. For each anchor text (hyperlinked string) in Wikipedia articles, we get its entity type (or non-entity) based on the FreeBase types of its target title, and only keep the mentions that are belong to one of the three types (PER, ORG, and LOC). We use 30,000 articles for Turkish, Tagalog, and Bengali, and 10,000 articles for the other languages. Note that we exclude the mentions which appear in the training pairs, and the mentions which are identical to the target title. For example, if a Spanish mention "Barack Obama" is linked to the English title "Barack_Obama", we will exclude this mention. Since this trivial case will be handled without a name translation model in practice. The number of test mentions for each language is listed in the column "#Mentions" of Table 4.7.

The title candidate generation algorithm is as follows. We collect all anchor text and its corresponding title from all articles in the English Wikipedia. We then build three dictionaries from this collection. The first one simply maps each anchor text (the entire string) to all possible titles. The second dictionary breaks each anchor text into words and maps each word to all possible titles. The third dictionary further breaks words into character 4-grams and maps each character 4-gram to all possible titles. In other words, the first dictionary has the highest precision but lowest recall. In contrast, in the third dictionary, each character 4-gram is likely to be mapped to many titles, thus has the highest recall. We sort the titles by $P(\text{title}|\text{key})$ in each dictionary, where "key" is the key of each dictionary (phrases, words, or character 4-grams).

For each mention (translated English name), we will generate at most 30 candidate titles. We query the first dictionary by the entire mention string to retrieve the top 30 titles. If there are less than 30 titles, we then query the second dictionary by each word in the mention. The third dictionary is used in a similar way if the total number of candidates is still less than 30. It is true that generating more than 30 candidates can make the coverage higher for all models. However, as Tsai and Roth (2016a) pointed out, generating too many candidates will result in worse ranking performance in the later steps of the wikification pipeline.

The results are shown in Table 4.9. The numbers indicate the percentage of mentions which have gold English title in the candidate set. Note that we use the same candidate generation algorithm for all models, and we do not evaluate the ranking performance in this experiment since this problem is not the focus of this work. To compare different transliteration and translation models, we only want to see if the name translation can help to retrieve the target English title.

We can see that although the relative performance between models is similar to the trend shown in Table 4.8, the range of the numbers in Table 4.9 is much wider. This indicates that even if two strings are pretty

	Spanish		
	Precision	Recall	F1
Base system	56.33	51.33	53.71
+Proposed model	**63.62**	**57.98**	**60.67†**
	Chinese		
Base system	69.95	58.53	63.62
+Proposed model	**72.05**	**60.10**	**65.53†**

Table 4.10: End-to-end wikification performance on TAC 2016 EDL task. Incorporating the proposed name translation model into the base system improves the overall performance for both languages.

similar in terms of fuzzy F1 score, they could generate very different set of candidates. More importantly, a string which has a higher fuzzy F1 score does not always retrieve the correct title. For example, NMT-char gets the best score on Spanish location names in Table 4.8, but it only ranked the fourth in generating candidates. We notice that NMT-char tend to generate tokens of popular location names in the training pairs. This behavior may not hurt the fuzzy F1 score much, but when generate candidates, it will generate candidates which are totally unrelated to the mention. On the other hand, Sequitur fails to transliterate several tokens of the test mentions, so the predictions tend to be short. Again, although the fuzzy F1 score of Sequitur looks good, it fails to generate the correct title since some key words in the foreign mentions are not translated.

End-to-end Wikification Performance

To evaluate the impact of using the proposed model in a cross-lingual wikification system, we add our name translation model to one of the top systems (Tsai et al., 2016)[8] in the TAC 2016 Entity Discovery and Linking shared task (Ji et al., 2016) in which the two target languages are Chinese and Spanish. This system simply uses the approach which relies on the inter-language links to generate English title candidates. As discussed in Section 1, if the target entity does not exist in the target-language Wikipedia or it is not linked to the corresponding English page, this approach will fail to retrieve the correct title.

We augment this base system in the following way: if the base system does not generate any candidate for a mention, we use our model to translate the mention into English and then query the English title index. Note that the test documents were written after 2011, and many entities were added into Wikipedia after the events happened. To simulate a more challenging and real-time situation, we remove the entities in the target-language Wikipedia which were created after 2011 in this experiment. The results are shown in Table 4.10. A predicted mention is considered correct if and only if the mention boundary, entity type,

[8] `https://github.com/cttsai/illinois-cross-lingual-wikifier`.

and the FreeBase ID (which can be derived from Wikipedia titles) are all identical to a gold mention. We can see that the scores on both languages have improved significantly by incorporating the proposed model. The smaller improvement on Chinese indicates that Chinese-to-English name translation is harder than Spanish-to-English, but the smaller gap is also due to the fact that the naive candidate generation approach works better on Chinese.

4.2.3 Related Work

Besides the transliteration models that we introduced in the experiment section, Irvine et al. (2010) mines training word pairs from inter-language links in Wikipedia. Although they only work on person names in which the words can be easily aligned, they conduct careful analysis on 13 languages and show the effect of the amount of training data on transliteration performance.

Tao et al. (2006); Yoon et al. (2007); Klementiev and Roth (2008); Goldwasser et al. (2009) propose to discover name transliteration from comparable corpora or temporally aligned documents. Although these resources may not be available for low-resource languages, these methods could be used for generating more training phrase pairs for our model. In TAC EDL (Ji et al., 2016), some teams also try to mine name translation pairs from comparable corpora in order to improve cross-lingual wikification performance.

4.2.4 Conclusion

We propose a probabilistic model to learn name translation from Wikipedia titles. Using inter-language links in Wikipedia, we can collect training title pairs for more than 250 languages. The proposed model jointly considers word alignments and word transliteration, therefore it has advantage in learning location and organization names in which words are ordered differently across languages. We show that our model outperforms 6 other transliteration and translation models not only on a string similarity metric, but also on the ability of generating title candidates for the cross-lingual wikification problem.

4.3 Cross-lingual Wikification Using Multilingual Embeddings

After getting a list of English title candidates for each foreign mention, the next challenge in the cross-lingual wikification pipeline is choosing the most relevant title based on the contextual information in the given document. In other words, there is a need to compute cross-lingual similarity between foreign words in the documents and the English title candidates.

In this section, we address this problem by using multilingual title and word embeddings. We represent words and Wikipedia titles in both the foreign language and in English in the same continuous vector space, which allows us to compute meaningful similarity between mentions in the foreign language and titles in English. We show that learning these embeddings only requires Wikipedia documents and language links between the titles across different languages, which are quite common in Wikipedia. Therefore, we can learn embeddings for all languages in Wikipedia without any additional annotation or supervision.

For evaluation purposes, we focus in this section on mentions that have corresponding titles in both the English and the foreign language Wikipedia, and concentrate on disambiguating titles across languages. This allows us to evaluate on a large number of Wikipedia documents. Note that under this setting, a natural approach is to do wikification on the foreign language and then follow the language links to obtain the corresponding English titles. However, this approach requires developing a separate wikifier for each foreign language if it uses language-specific features, while our approach is generic and only requires using the appropriate embeddings. Importantly, the aforementioned approach will also not generalize to the cases where the target titles only exist in the English Wikipedia while ours does.

We create a challenging Wikipedia dataset for 12 foreign languages and show that the proposed approach, WikiME (**Wiki**fication using **M**ultilingual **E**mbeddings), consistently outperforms various baselines. Moreover, the results on the TAC KBP 2015 Entity Linking dataset show that our approach compares favorably with the best Spanish system and the best Chinese system despite using significantly weaker resources (no need for translation). We note that the need for translation would have prevented the wikification of 12 languages used in the experiment section.

4.3.1 Multilingual Entity and Word Embeddings

In this section, we describe how we generate a vector representation for each word and Wikipedia title in any language.

Monolingual Embeddings

The first step is to train monolingual embeddings for each language separately. We adopt the "Alignment by Wikipedia Anchors" model proposed in Wang et al. (2014). For each language, we take all documents in Wikipedia and replace the hyperlinked text with the corresponding Wikipedia title. For example, consider the following Wikipedia sentence: "It is led by and mainly composed of **Sunni** Arabs from **Iraq** and **Syria**.", where the three bold faced mentions are linked to some Wikipedia titles. We replace those mentions and the sentence becomes "It is led by and mainly composed of `en/Sunni_Islam` Arabs from `en/Iraq` and `en/Syria`." We then learn the skip-gram model Mikolov et al. (2013a,b) on this newly generated text. Since a title appears as a token in the transformed text, we will obtain an embedding for each word and title from the model.

The skip-gram model maximizes the following objective:

$$\sum_{(w,c)\in D} \log \frac{1}{1+e^{-v'_c \cdot v_w}} + \sum_{(w,c)\in D'} \log \frac{1}{1-e^{-v'_c \cdot v_w}},$$

where w is the target token (word or title), c is a context token within a window of w , v_w is the target embedding represents w, v'_c is the embedding of c in context, D is the set of training documents, and D' contains the sampled token pairs which serve as negative examples. This objective is maximized with respect to variables v_w's and v'_w's. In this model, the target token is used to predict the tokens in the context. The token pairs in the training documents are positive examples, and the randomly sampled pairs are negative examples.

Multilingual Embeddings

After getting monolingual embeddings, we adopt the model proposed in Faruqui and Dyer (2014) to project the embeddings of a foreign language and English to the same space. The requirement of this model is a dictionary which maps the words in English to the words in the foreign language. Note that there is no need to have this mapping for every word. The aligned words are used to learn the projection matrices, and the matrices can later be applied to the embeddings of each word to obtain the enhanced new embeddings. Faruqui and Dyer (2014) obtain this dictionary by picking the most frequent translated word from a parallel corpus. However, there is a limited or no parallel corpus for many languages. Since our monolingual embedding model consists also of title embeddings, we can use the Wikipedia title alignments between two languages as the dictionary.

Let $A_{en} \in R^{a\times k_1}$ and $A_{fo} \in R^{a\times k_2}$ be the matrices containing the embeddings of the aligned English and foreign language titles, where a is the number of aligned titles and k_1 and k_2 are the dimensionality of English embeddings and foreign language embeddings respectively (i.e., each row is a title embedding). Canonical correlation analysis (CCA) Hotelling (1936) is applied to these two matrices:

$$P_{en}, P_{fo} = CCA(A_{en}, A_{fo}),$$

where $P_{en} \in R^{k_1 \times d}$ and $P_{fo} \in R^{k_2 \times d}$ are the projection matrices for English and foreign language embeddings, and d is the dimensionality of the projected vectors, which is a parameter in CCA.

Let $E_{en} \in R^{n_1 \times k_1}$ be the matrix containing the monolingual embeddings for all words and titles in English, where the number of words and titles is n_1, We obtain the multilingual embeddings of English words and titles by

$$E'_{en} = E_{en} P_{en} \in R^{n_1 \times d}.$$

Similarly, the multilingual embeddings of the foreign words and titles are stored in the rows of

$$E'_{fo} = E_{fo} P_{fo} \in R^{n_2 \times d},$$

where there are n_2 words and titles in the foreign language. The rows of E'_{en} and E'_{fo} are the representations of words and titles that we use to create the similarity features in the ranker.

Faruqui and Dyer (2014) show that the multilingual embeddings perform better than monolingual embeddings on various English word similarity datasets. Since synonyms in English may be translated into the same word in a foreign language, the CCA model could bring the synonyms in English closer in the embedding space. In this section, we further show that projecting the embeddings of the two languages into the same space helps us computing better similarity between the words and titles across languages and that a bilingual dictionary consisting of pairs of Wikipedia titles is sufficient to induce these embeddings.

4.3.2 Cross-lingual Wikification Pipeline

We now describe the algorithm for finding the English title given a foreign mention. This algorithm is used in all experiments in the next section.

FEATURE TYPE	DESCRIPTIONS
Basic	$Pr(c\|m)$ and $Pr(m\|c)$, the fraction of times the title candidate c is the target page given the mention m, and the fraction of times c is referred by m
Other Mentions	Cosine similarity of $e(c)$ and the average of vectors in *other-mentions*(m) The max and mini cosine similarity of the vectors in *other-mentions*(m) and $e(c)$
Local Context	Cosine similarity of $e(c)$ and $context_j(m)$, for $j = 30, 100$, and 200
Previous Titles	Cosine similarity of $e(c)$ and the average of vectors in *previous-titles*(m) The max and min cosine similarity of the vectors in *previous-titles*(m) and $e(c)$

Table 4.11: Features for measuring similarity of an English title candidate c and a mention m in the foreign language, where $e(c)$ is the English title embedding of c. *other-mentions*(m), *previous-titles*(m), and $context_j(m)$ are defined in Section 4.3.2.

Candidate Generation

Given a mention m, the first step is to select a set of English title candidates C_m, a subset of all titles in the English Wikipedia. Ideally the correct title is included in this set. The goal is to produce a manageable number of candidates so that a more sophisticated algorithm can be applied to disambiguate them.

Since we focus on the titles in the intersection of English and the foreign language Wikipedia, we can build indices from the anchor texts in the foreign language Wikipedia. More specifically, we create two dictionaries and apply a two-step approach. The first dictionary maps each hyperlinked mention string in the text to the corresponding English titles. We simply lookup this dictionary by using the query mention m to retrieve all possible titles. The title candidates are initially sorted by $Pr(title|mention)$, the fraction of times the title is the target page of the given mention. This probability is estimated from all Wikipedia documents. The top k title candidates are then returned.

If the first high-precision dictionary fails to generate any candidate, we then lookup the second dictionary. We break each hyperlinked mention string into tokens, and create a dictionary which maps tokens to English titles. The tokens of m are used to query this dictionary. Similarly, the candidates are sorted by $Pr(title|token)$ and the top k candidates are returned.

Candidate Ranking

Given a mention m and a set of title candidates C_m, we compute a score for each title in C_m which indicates how relevant the title is to m. For a candidate $c \in C_m$, we define the relevance as:

$$s(m, c) = \sum_i w_i \phi_i(m, c), \tag{4.8}$$

LANGUAGE	#TOKENS	#ALIGN. TITLES
German	616,347,668	960,624
Spanish	460,984,251	754,740
French	357,553,957	1,088,660
Italian	342,038,537	836,154
Chinese	179,637,674	469,982
Hebrew	75,076,391	137,821
Thai	68,991,911	72,072
Arabic	67,954,771	255,935
Turkish	47,712,534	162,677
Tamil	12,665,312	50,570
Tagalog	4,925,785	48,725
Urdu	3,802,679	83,665

Table 4.12: The number of tokens used in training the skip-gram model and the number of titles which can be aligned to the corresponding English titles via the language links in Wikipedia.

a weighted sum of the features, ϕ_i, which are based on multilingual title and word embeddings. We represent the mention m by the following contextual clues and use these representation to compute feature values:

- $context_j(m)$: use the tokens within j characters of m to compute the TF-IDF weighted average of their embeddings in the foreign language.
- *other-mentions*(m): a set of vectors that represent other mentions. For each mention in the document other than m, we represent it by averaging the embeddings of the tokens in the mention surface string.
- *previous-titles*(m): a set of vectors that represent previous entities. For each mention before m, we represent it by the English embedding of the disambiguated title.

Let $e(c)$ be the English embedding of the title candidate c. The features used in Eq. (4.8) are shown in Table 4.11. We train a linear ranking SVM model with the proposed features to obtain the weights, w_i, in Eq. (4.8). Finally, the title which has the highest relevant score is chosen as the answer to m.

4.3.3 Experiments

We evaluate the proposed method on the Wikipedia dataset of 12 langugaes and the TAC'15 Entity Linking dataset.

For all experiments, we use the Word2Vec implementation in Gensim[9] to learn the skip-gram model with dimensionality 500 for each language. The CCA code for projecting mono-lingual embeddings is from Faruqui and Dyer (2014)[10] in which the ratio parameter is set to 0.5 (i.e., the resulting multilingual embeddings have dimensionality 250).

[9] `https://radimrehurek.com/gensim/`

[10] `https://github.com/mfaruqui/crosslingual-cca`

language	#training	#test (#hard)
German	23,124	9,798 (3,266)
Spanish	30,471	12,153 (4,051)
French	37,860	14,358 (4,786)
Italian	34,185	12,775 (4,254)
Chinese	44,246	11,394 (3,798)
Hebrew	20,223	16,146 (5,382)
Thai	16,819	11,381 (3,792)
Arabic	22,711	10,646 (3,549)
Turkish	12,942	13,798 (4,598)
Tamil	21,373	11,346 (3,776)
Tagalog	4,835	1,074 (358)
Urdu	1,413	1,389 (463)

Table 4.13: The number of training and test mentions of the Wikipedia dataset. The mentions are from the hyperlinked text in randomly selected Wikipedia documents. We ensure that there are at least one-third of test mentions are hard (cannot be solved by the most common title given the mention).

We use Stanford Word Segmenter Chang et al. (2008b) for tokenizing Chinese, and use the Java built-in BreakIterator for Thai. For all other languages, tokenization is based on whitespaces. The number of tokens we use to learn the skip-gram model and the number of title alignments used by the CCA are given in Table 4.12. For learning the weights in Eq. (5.2), we use the implementation of linear ranking SVM in Lee and Lin (2014). Parameter selection and feature engineering are done by conducting cross-validation on the training data of Spanish Wikipedia dataset.

Wikipedia Dataset

We create this dataset from the documents in Wikipedia by taking the anchors (hyperlinked texts) as the query mentions and the corresponding English Wikipedia titles as the answers. Note that we only keep the mentions for which we can get the corresponding English Wikipedia titles by the language links. As observed in previous work Ratinov et al. (2011), most of the mentions in Wikipedia documents are easy, that is, the baseline of simply choosing the title that maximizes $Pr(title|mention)$, the most frequent title given the mention surface string, performs quite well. In order to create a more challenging dataset, we randomly select mentions such that the number of easy mentions is about twice the number of hard mentions (those mentions for which the most common title is not the correct title). This generation process is inspired by (and close to) the distribution generated in the TAC KBP2015 Entity Linking Track. Another problem that occurs when creating a dataset from Wikipedia documents is that even though training documents are different from test documents, many mentions and titles actually overlap. To test that the algorithms really generalize from training examples, we ensure that no (mention, title) pair in the test set appear in the training set. Table 4.13 shows the number of training mentions, test mentions, and hard mentions in the

LANGUAGE	METHOD	HARD	EASY	TOTAL
German	MonoEmb	35.18	**96.92**	76.34
	WordAlign	52.39	95.32	81.01
	WikiME	**53.28**	95.53	**81.45**
	Ceiling	90.20	100	96.73
Spanish	EsWikifier	40.11	**99.28**	79.56
	MonoEmb	38.46	96.12	76.90
	WordAlign	48.75	95.78	80.10
	WikiME	**54.46**	94.83	**81.37**
	Ceiling	93.46	100	97.69
French	MonoEmb	23.17	**97.16**	72.50
	WordAlign	41.70	96.08	77.96
	WikiME	**47.51**	95.72	**79.65**
	Ceiling	89.41	100	96.47
Italian	MonoEmb	32.68	**97.48**	75.90
	WikiME	**48.28**	95.52	**79.79**
	Ceiling	87.99	100	96.00
Chinese	MonoEmb	43.73	97.85	79.81
	WikiME	**57.61**	**98.03**	**84.55**
	Ceiling	94.29	100	98.10
Hebrew	MonoEmb	42.59	**98.16**	79.64
	WikiME	**56.67**	97.71	**84.03**
	Ceiling	96.84	100	98.95
Thai	MonoEmb	53.43	99.08	83.87
	WikiME	**70.02**	**99.17**	**89.46**
	Ceiling	94.49	100	98.16
Arabic	MonoEmb	39.81	**98.99**	79.26
	WikiME	**62.05**	98.17	**86.13**
	Ceiling	93.27	100	97.76
Turkish	MonoEmb	40.47	**98.15**	78.93
	WikiME	**60.18**	97.55	**85.10**
	Ceiling	94.08	100	98.03
Tamil	MonoEmb	34.51	98.65	77.30
	WikiME	**54.13**	**99.13**	**84.15**
	Ceiling	95.60	100	98.54
Tagalog	MonoEmb	35.47	**99.44**	78.12
	WikiME	**56.70**	98.46	**84.54**
	Ceiling	90.78	100	96.93
Urdu	MonoEmb	63.71	98.81	87.11
	WikiME	**74.51**	**99.35**	**91.07**
	Ceiling	90.06	100	96.69

Table 4.14: Ranking performance (Precision@1) of different approaches on various languages. Since about one-third of the test mentions are non-trivial, a baseline is 66.67 for all languages, if we pick the most common title given the mention. **Bold** signifies highest score for each column.

test set of each language. This dataset is publicly available at `http://bilbo.cs.illinois.edu/~ctsai12/xlwikifier-wikidata.zip`.

The performance of the proposed method (WikiME) is shown in Table 4.14 along with the following

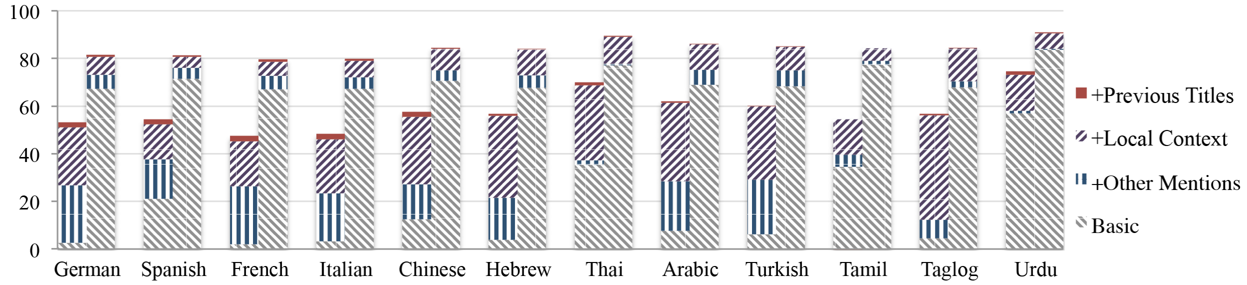


Figure 4.6: Feature ablation study of WikiME. The left bar of each language shows the performance on hard mentions, whereas the right bar corresponds to the performance of all mentions. The descriptions of feature types are listed in Table 4.11.

approaches:

- **MonoEmb**: In this method, we use the monolingual embeddings before applying CCA while all the other settings are the same as in WikiME. Since the monolingual embeddings are learnt separately for each language, calculating the cosine similarity of the word embedding in the foreign language and an English title embedding does not produce a good similarity function. The ranker, though, learns that the most important feature is $Pr(title|mention)$, and, consequently, performs well on easy mentions but has poor performance on hard mentions.

- **WordAlign**: Instead of using the aligned Wikipedia titles in generating multilingual embeddings, the CCA model operates on the word alignments as originally proposed in Faruqui and Dyer (2014). We use the word alignments provided by Faruqui and Dyer (2014), which are obtained from the parallel news commentary corpora combined with the Europarl corpus for English to German, France, and Spanish. The number of aligned words for German, France, and Spanish are 37,484, 37,582, and 37,554 respectively. WikiME performs statistically significantly better than WordAlign on all three languages.

- **EsWikifier**: We use Illinois Wikifier Ratinov et al. (2011); Cheng and Roth (2013) on a Spanish Wikipedia dump and train its ranker on the same set of documents that are used in WikiME.

- **Ceiling**: These rows show the performance of title candidate generation. That is, the numbers indicate the percentage of mentions that have the gold title in its candidate set, therefore upper-bounds the ranking performance.

In sum, WikiME can disambiguate the hard mentions much better than other methods without sacrificing the performance on the easy mentions much. Comparing across different languages, it is important to note

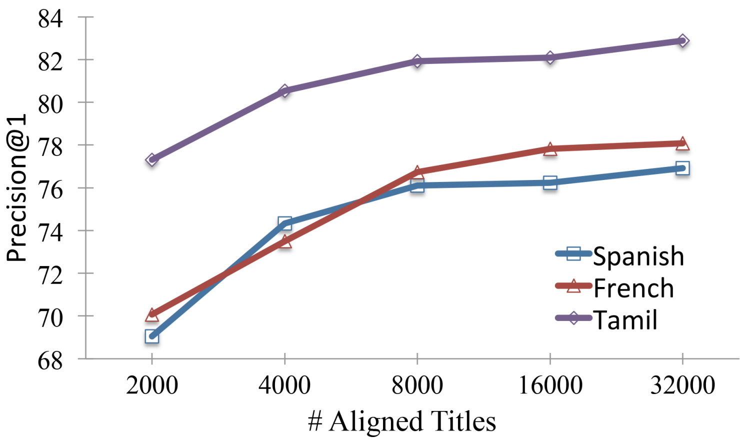


Figure 4.7: The number of aligned titles used in generating multilingual embeddings versus the performance of WikiME.

that languages which have a smaller size Wikipedia tend to have better performance, despite the degradation in the quality of the embeddings (see below). This is due to the difficulty of the datasets. That is, there is less ambiguity because the number of articles in the corresponding Wikipedia is small.

Figure 4.6 shows the feature ablation study of WikiME. For each language, we show results on hard mentions (the left bar) and all mentions (the right bar). We do not show the performance on easy mentions since it always stays high and does not change much. We can see that *Local Context* and *Other Mentions* are very effective for most of the languages. In particular, on hard mentions, the performance gain of the three feature groups is from almost 0 to around 50. For the easier dataset such as Urdu, *Basic features* alone work quite well.

Figure 4.7 shows the performance of WikiME when we vary the number of aligned titles in generating multilingual embeddings. The performance drops a lot when there are only few aligned titles, especially for Spanish and French, where the results are even worse than MonoEmb when only 2000 titles are aligned. This indicates that the CCA method needs enough aligned pairs in order to produce good embeddings. The performance does not change much when there are more than 16,000 aligned titles.

TAC KBP 2015 Entity Linking

To evaluate our system on documents outside Wikipedia, we conduct an experiment on the evaluation documents in TAC KBP2015 Tri-Lingual Entity Linking Track. In this dataset, there are 166 Chinese documents (84 news and 82 discussion forum articles) and 167 Spanish documents (84 news and 83 discussion forum articles). The mentions in this dataset are all named entities of five types: Person, Geo-political Entity, Organization, Location, and Facility.

APPROACH	SPANISH	CHINESE
Translation + EnWikifier	79.35	N/A
EsWikifier	79.04	N/A
WikiME	**82.43**	**85.07**
+Typing		
Top TAC'15 System	80.4	83.1
WikiME	**80.93**	**83.63**

Table 4.15: TAC KBP2015 Entity Linking dataset. All results use gold mentions and the metric is precision@1. The top section only evaluates the linked FreeBase ID. To compare with the best systems in TAC, we also classify each mention into the five entity types. The results which evaluate both FreeBase IDs and entity types are shown in the bottom section.

Table 4.15 shows the results. Besides the Spanish Wikifier (EsWikifier) that we used in the previous experiment, we implemented another baseline for Spanish Wikification. In this method, we use Google Translate to translate the whole documents from Spanish to English, and then the English Illinois Wikifier is applied to disambiguate the English gold mentions. Note that the target Knowledge Base of this dataset is FreeBase, therefore we use the FreeBase API to map the resulting English or Spanish Wikipedia titles to the corresponding FreeBase ID. If this conversion fails to find the corresponding FreeBase ID, "NIL" is returned instead.

The ranker models used in all three systems are trained on Wikipedia documents. We can see that WikiME outperforms both baselines significantly on Spanish. It is interesting to see that the translation-based baseline performs slightly better than the Spanish Wikifier, which indicates that the machine translation between Spanish and English is quite reliable. Note that this translation-based baseline got the highest score in this shared task when the mention boundaries were not given.

The row "Top TAC'15 System" lists the best scores of the diagnostic setting in which mention boundaries are given Ji et al. (2015). Since the official evaluation metric considers not only the linked FreeBase IDs but also the entity types, namely, an answer is counted as correct only if the FreeBase ID and the entity type are both correct, we built two simple 5-class classifiers to classify each mention into the five entity types so that we can compare with the state of the art. One classifier uses FreeBase types of the linked FreeBase ID as features, and this classifier is only applied to mentions that are linked to some entry in FreeBase. For NIL mentions, another classifier which uses word form features (words in the mention, previous word, and next word) is applied. Both classifiers are trained on the training data of this task. From the last two rows of Table 4.15, we can see that WikiME achieves better results than the best TAC participants.

4.3.4 Related Work

Wikification on English documents has been studied extensively. Earlier works Bunescu and Pasca (2006); Mihalcea and Csomai (2007) focus on local features which compare context words with the content of candidate Wikipedia pages. Later, several works Cucerzan (2007); Milne and Witten (2008); Han and Zhao (2009); Ferragina and Scaiella (2010); Ratinov et al. (2011) proposed to explore global features, trying to capture coherence among titles that appear in the text. In our method, we compute local and global features based on multilingual embeddings, which allow us to capture better similarity between words and Wikipedia titles across languages.

The annual TAC KBP Entity Linking Track has used the cross-lingual setting since 2011 Ji et al. (2011, 2014, 2015), where the target foreign languages are Spanish and Chinese. To our best knowledge, most of the participants use one of the following two approaches: (1) Do entity linking in the foreign language, and then find the corresponding English titles from the resulting foreign language titles; and (2) Translate the query documents to English and do English entity linking. The first approach relies on a large enough Knowledge Base in the foreign language, whereas the second depends on a good machine translation system. The approach developed in this work makes significantly simpler assumptions on the availability of such resources, and therefore can scale also to lower-resource languages, while doing very well also on high-resource languages.

Wang et al. (2015) proposed an unsupervised method which matches a knowledge graph with a graph constructed from mentions and the corresponding candidates of the query document. This approach performs well on the Chinese dataset of TAC'13, but falls into the category (1). Moro et al. (2014) proposed another graph-based approach which uses Wikipedia and WordNet in multiple languages as lexical resources. However, they only focus on English Wikification.

McNamee et al. (2011) aims at the same cross-lingual Wikification setting as we do, where the challenge is in comparing foreign language words with English titles. They treat this problem as a cross-lingual information retrieval problem. That is, given the context words of the target mention in the foreign language, retrieve the most relevant English Wikipedia page. However, their approach requires parallel text to estimate word translation probabilities. In contrast, our method only needs Wikipedia documents and the interlanguage links.

Besides the CCA-based multilingual word embeddings Faruqui and Dyer (2014) that we extend in Section 4.3.1, several other methods also try to embed words in different languages into the same space. Hermann and Blunsom (2014) use a sentence aligned corpus to learn bilingual word vectors. The intuition behind the model is that representations of aligned sentences should be similar. Unlike the CCA-based method which learns monolingual word embeddings first, this model directly learns the cross-lingual embeddings. Luong et al. (2015) propose Bilingual Skip-Gram which extends the monolingual skip-gram model and learns bilingual embeddings using a parallel copora and word alignments. The model jointly considers within language co-occurrence and meaning equivalence across languages. That is, the monolingual objective for each language is also included in their learning objective. Several recent approaches Gouws et al. (2014); Coulmance et al. (2015); Shi et al. (2015); Soyer et al. (2015) also require a sentence aligned parallel corpus to learn multilingual embeddings. Unlike other approaches, Vulić and Moens (2015) propose a model that only requires comparable corpora in two languages to induce cross-lingual vectors. Similar to our proposed approach, this model can also be applied to all languages in Wikipedia if we treat documents across two Wikipedia languages as a comparable corpus. However, the quality and quantity of this comparable corpus for low-resource languages will be low, we believe.

We choose the CCA-based model because we can obtain multilingual word and title embeddings for all languages in Wikipedia without any additional data beyond Wikipedia. In addition, by decoupling the training of the monolingual embeddings from the cross-lingual alignment we make it easier to improve the quality of the embeddings by getting more text in the target language or a better dictionary between English and the target language. Nevertheless, as cross-lingul wikification provides another testbed for multilingual embeddings, it would be very interesting to compare these recent models on Wikipedia languages.

4.3.5 Conclusion

We propose a new, low-resource, approach to Wikification across multiple languages. Our first step is to train multilingual word and title embeddings jointly using title alignments across Wikipedia collections in different languages. We then show that using features based on these multilingual embeddings, our wikification ranking model performs very well on a newly constructed dataset in 12 languages, and achieves state of the art also on the TAC'15 Entity Linking dataset.

An immediate future direction following our work is to improve the title candidate generation process so that it can handle the case where the corresponding titles only exist in the English Wikipedia. This only requires augmenting our method with a transliteration tool and, together with the proposed disambiguation approach across languages, this will be a very useful tool for low-resource languages which have a small number of articles in Wikipedia.

Chapter 5

Concept Grounding to Multiple Knowledge Bases via Indirect Supervision

Grounding entities and concepts appearing in text documents to a knowledge base (KB) has become a popular method for contextually disambiguating them and can be used also for focused knowledge acquisition. It has been shown a valuable component for several natural language processing and information extraction tasks across different domains. In the news domain, the task is often called *Wikification* or *Entity Linking* and has been studied extensively recently Bunescu and Pasca (2006); Cucerzan (2007); Mihalcea and Csomai (2007); Ratinov et al. (2011); Cheng and Roth (2013). Wikipedia is widely used as the target KB due to its broad coverage and detailed information of concepts. While Wikipedia is an excellent general purpose encyclopedic resource, when the text is domain specific, it may not be the single ideal resource; the text could be better "covered" by multiple ontological or encyclopedic resources.

This is clearly the case for scientific text which is often covered by multiple ontologies, each addressing some aspects of the domain. For example, in the biological domain there are multiple ontologies: Entrez Gene Lu and Wilbur (2010) focuses on genomes that have been completely sequenced; Gene Ontology Ashburner et al. (2000) more broadly describes gene product characteristics; and ChEBI, is a dictionary of molecular entities focused on "small" chemical compounds. The ontologies provide complementary information, but they overlap and, in these cases, make use of different vocabulary and provide different relevant information.

In this chapter, we consider the problem of grounding concepts appearing in documents to multiple KBs. We use the biomedical domain as our application domain, both due to its importance and to the fact that thousands of person-years have been spent on putting together a large number of relevant KBs. We discuss other potential applications in the end of the chapter. The challenges in this problem are due both to ambiguity and variability in expressing concepts: a given mention in text can be used to express different concepts in the KBs, and a KB (ontology) concept may be expressed in text in multiple ways, such as synonyms or nicknames. In the case of using multiple KBs, an additional challenge is due to the overlap between KBs: a mention can refer to multiple concepts in different KBs and we want to ground the mention to all of them. Figure 4.2 shows an example of concept annotations from the CRAFT corpus Bada et al.

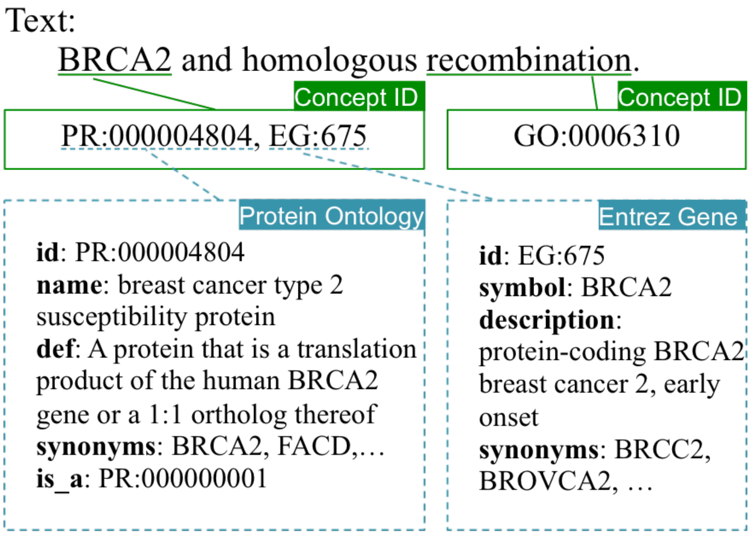

Text:
BRCA2 and homologous recombination.
Concept ID
PR:000004804, EG:675
Concept ID
GO:0006310
Protein Ontology
id: PR:000004804
name: breast cancer type 2 susceptibility protein
def: A protein that is a translation product of the human BRCA2 gene or a 1:1 ortholog thereof
synonyms: BRCA2, FACD,…
is_a: PR:000000001
Entrez Gene
id: EG:675
symbol: BRCA2
description: protein-coding BRCA2 breast cancer 2, early onset
synonyms: BRCC2, BROVCA2, …

Figure 5.1: An example of concept annotations in the CRAFT dataset and of common attributes of concepts in the KBs.

(2012). The mention *BRCA2* refers both to “breast cancer type 2 susceptibility protein (PR:000004804)” from the Protein Ontology and to “BRCA2 (EG:675)” from the Entrez Gene database, which has more than one hundred genes across different species that can be referred to as *BRCA2*.

In the context of Wikification, people often train a ranking model to score how relevant a KB concept is to a mention. It is straightforward to use Wikipedia to supervise this model, since the hyperlink structure in Wikipedia text indicates which title a mention refers to. However, other KBs may not have such useful information. An entry in a typical biological KB only consists of a name, a few sentences of definition, synonyms, and a few relations (Figure 5.1). In addition, it is relatively difficult to obtain human annotations in the biomedical domain due to the high level of expertise required and to highly ambiguous concepts.

Our key contribution in this work is to show that, by exploring the overlap and the relationship between KBs, we can obtain high quality indirect supervision signals for sufficiently many examples, and thus train a ranking model. Without using any document in training and no annotated supervision, our approach achieves better ranking results than all previous approaches tried on this problem.

We then explore another advantage of using multiple KBs; we show that, since concepts are represented in different ways in different KBs, there are some natural constraints between these representations. In the above example, if we determine that a gene mention is relevant to the human genome and should therefore be grounded to human concepts in the NCBI Taxonomy, we can easily rule out all the candidate genes from other species, which are not mentioned in the document; we can develop these constraints since genes in the Entrez Gene KB have NCBI Taxonomy IDs as species attributes. If we do not use the NCBI Taxonomy as

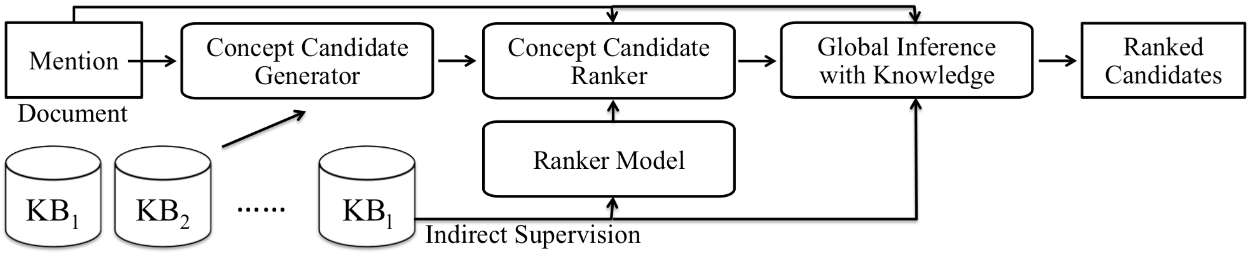


Figure 5.2: Algorithmic components of our system.

a knowledge source to ground concepts but rather only focus on disambiguating gene names in a document, we may lose this valuable information. Our final model combines this kind of prior knowledge with our ranker scores using a Constrained Conditional Model (CCM) Roth and tau Yih (2004); Chang et al. (2012) to enforce a coherent global mapping of all mentions in a given document to their corresponding concepts. The proposed system, CCMIS (CCM with Indirect Supervision), performs significantly better than the best unsupervised baseline and is competitive with a directly supervised model we use to assess the quality of the automatically generated indirect supervision.

5.1 Task Definition and Model Overview

We formalize the problem as follows. We are given a document d with a set of mentions $M = \{m_1, \ldots, m_n\}$, and l KBs, $k_1, \ldots, k_l$. Each KB k_j is a graph (T_j, R_j), where a concept $t \in T_j$ represents a node and a relation $r \in R_j$ between two concepts is an undirected edge. For each mention in the document, our goal is to retrieve a set of concepts $\mathcal{C} \subseteq T_1 \cup \cdots \cup T_l$ that the mention refers to. Note that a mention may refer to multiple concepts in a single or multiple KBs.

Figure 5.2 shows the algorithmic components in our system. The first step is concept candidates generation. Given a mention m_i from a document, it produces a candidate set $C_{m_i} \subseteq T_1 \cup \cdots \cup T_l$. That is, C_{m_i} is a subset of all concepts in the KBs. We only look at the surface string of the mention in this step, no contextual information is examined. The goal of this step is to quickly produce a short list of concepts which includes the correct answers. We call these concept candidates "grounding candidates" for mention m_i.

The second and key step is the ranking step where, given a concept mention in text, we assign a score to each of its potential KB grounding candidates, which indicates how relevant it is to the given mention. We train a linear ranking SVM model using the information in the KBs. Note that unlike the Wikification problem in which it is possible to use the Wikipedia structure to learn a ranker, most other KBs do not have text with hyperlinks. To overcome this problem, we propose a novel method to utilize the redundancy and relationship between KBs as indirect supervision. Specifically, if two concepts in different KBs are determined to be the same, we can assume that one is the "gold label" for the other, and extract textual and relational features between them, making this pair an approximation of the real grounding instance. This method only requires information from the KBs hence no annotated document is needed.

We would like to use multiple sources of knowledge in order to train a robust ranking function over a set of candidates. Some of these can be captured via features of the ranking function (e.g., textual similarity between the context of a mention and the description of a concept in a KB); some, are better captured as constraints over the ranking produced by a ranking function. For example, if we choose to map a mention into a node in the KB that is species-specific, we insist on the species being mentioned in the context of the target mention. We will not link otherwise. As a way to combine statistical information and such declarative constraints we formulate our problem using Constrained Conditional Model (CCM) Roth and tau Yih (2004); Chang et al. (2012). We formulate the following Integer Linear Program (ILP) objective function to enforce a coherent global solution of all mentions in a document:

$$\begin{aligned} \arg\max_{\mathbf{e}} \quad & \sum_{i=1}^{n} \sum_{j=1}^{|C_{m_i}|} s_i^j e_i^j - \sum_k \rho_k \gamma_k(\mathbf{e}) \\ s.t. \quad & e_i^j \in \{0,1\}, \quad \forall i,j \end{aligned} \tag{5.1}$$

where e_i^j is a boolean variable that indicates if we choose the j-th candidate of the i-th mention, $\mathbf{e}$ is a vector that contains all the e_i^j's, and s_i^j is the ranker score, capturing the relatedness of mention m_i and its j-th candidate. In the second component, γ_k is a boolean variable that indicates whether the k-th constraint is violated, and ρ_k is the pre-defined penalty for violating the k-th constraint. Constraints are often defined on a subset of variables from our prior knowledge. For example, one constraint used in our system states that genes of a species can only be selected if that species is mentioned somewhere in the document. This ILP problem can be solved quickly by an off-the-shelf ILP package since only a small number of variables are constrained. In the end, all the selected concept candidates are ranked according to s_i^j, and a list of ranked concepts is returned for each mention.

In the following sections, we describe each component in detail.

5.2 Concept Candidate Generation

In the first step of our system, given a mention m from a document, we produce a set of concept candidates C_m which is a subset of all concepts in the KBs. We want to reduce the number of concept candidates from millions to a manageable size, so that a more sophisticated and resource-hungry algorithm can be applied to disambiguate them. There is a trade-off here: we want C_m to contain all the answers that correspond to m, but if there are too many candidates, the performance of the ranker may suffer. Therefore, we first do synonym matching, which gives a high precision candidate list. Word matching is then applied only if synonym matching fails to generate any candidate. We describe the synonym matching and word matching procedures in the following sections.

5.2.1 Synonym Matching

We construct a dictionary from all synonyms and name fields across all KBs. This dictionary maps a string (synonym or name) to all possible concepts. In order to handle variations of words, we use the SPECIALIST Lexical Tools [1] to normalize each token of synonyms and the given input mention. Doing exact matching between the mention and all the names in KBs gives a high precision candidate list.

5.2.2 Word Matching

After synonym matching, many mentions may still have an empty candidate list because the KBs do not cover all possible ways to express a concept. If no candidate is generated after applying the first dictionary lookup method, we compare words in the mention with words in the KBs and their synonyms. We use as candidates all those concepts that match in this process. Note that this strategy may return a large number of concepts, therefore we only keep the top k concepts to maintain feasibility. We use the score from the PageRank algorithm (Section 5.6.4) to rank concepts initially.

5.3 Concept Candidate Ranking

This section describes how we obtain the relevance scores s_i^j in Eq. (5.1) for each (mention, concept) pair. Given a mention m and a concept candidate $c \in C_m$, we define the *relevance* of c to m as:

$$s(m, c) = \phi(m, c) + \psi(c, \Gamma_m). \tag{5.2}$$

[1] `http://lexsrv3.nlm.nih.gov/LexSysGroup/Projects/lvg/current/web/index.html`

The first component $\phi(m, c)$ measures the local compatibility between the mention and the concept candidate. It uses text-based features to capture the intuition that a given concept c is more likely to be referred to by the mention m if the entry of c in a KB has high textual similarity to the text around m. We model it as a linear combination of a set of local features ϕ_i:

$$\phi(m, c) = \sum_i w_i \phi_i(m, c).$$

The second component of the scoring function (5.2), $\psi(c, \Gamma_m)$ is a global component that captures how well does the concept c fit into the disambiguation context Γ_m of the mention m. The disambiguation context consists of other concepts in the document or close to the mention. Of course, we do not know what the concepts that correspond to other mentions in the document are, and different ways to construct disambiguation context have been proposed Cucerzan (2007); Milne and Witten (2008); Ratinov et al. (2011). Since in our case (in difference from the standard Wikification) a mention may refer to multiple concepts, using the current top ranked concept candidate from other mentions Ratinov et al. (2011) may lose some useful information. Therefore, we develop an approach that is similar to Cucerzan (2007). Instead of considering all the ambiguous mentions in the document, we take all concept candidates from mentions in nearby sentences as our disambiguation context. Although some irrelevant concepts are included, we rely on a high precision candidate generation process to reduce errors. Similar to the local score model, we design a set of global features ψ_j across multiple KBs and define:

$$\psi(m, c) = \sum_j w_j \psi_j(c, \Gamma_m).$$

In addition to local and global features, we use the PageRank score of the concept candidates as a baseline feature. To rank concept candidates $c \in C_m$ of a mention m, we use a linear ranking SVM to learn the weights w_i and w_j of the local and global features, respectively. The features used in our experiments are listed in the following sections.

5.3.1 Local Features

The local features used in our system are calculated from $context(m)$ and $def(c)$, where $context(m)$ represents the bag of words from p sentences before and after the mention m, and $def(c)$ is the bag of words from the definition of c in a KB. Words are lowercased and stemmed.

- $|context(m) \cap def(c)|$. The total number of common words in the context of m and c.

- Cosine similarity between the tf-idf vectors of $context(m)$ and $def(c)$. The i-th component in the vector is the tf-idf score of the i-th word in the vocabulary. The document frequency of words is calculated from all definitions in KBs, each definition representing a document.

- Common words in $context(m)$ and $def(c)$. This is a sparse boolean vector with length that is the size of vocabulary. The i-th feature is on if the i-th word in the vocabulary exists in both $context(m)$ and $def(c)$. Instead of using tf-idf vectors to capture the importance of each word, we use this feature to learn a weight for each word.

5.3.2 Global Features

Global features are defined on $neighbor(c)$ and Γ_m, where $neighbor(c)$ is the set of concepts which have relations with c in any of the KBs, and Γ_m is a set of candidate concepts from other mentions in the context of mention m. We consider all the mentions in p sentences before and after m in our experiments.

- $|neighbor(c) \cap \Gamma_m|$. The total number of common concepts in $neighbor(c)$ and Γ_m. We also split this number according to different KBs, and keep a feature that indicates the total number of common concepts from each KB.

- Common concepts in $neighbor(c)$ and Γ_m. This is a sparse boolean vector with length that is the total number of concepts. The i-th feature is on if the i-th concept exists in both $neightbor(c)$ and Γ_m.

5.4 Indirect Supervision

One of our key contributions in this work is a way to train the model described above, without any supervision and no information (such as hypelinks) from the documents. To accomplish that, we devise an indirect supervision method that explores the redundancy of information in the KBs and the relationship between KBs to construct training examples, so that we can train a ranking SVM model without any annotated document.

We make the assumption that if two concepts from different KBs have the same cross reference field, they are, in fact, the same concept. For instance, the concept named *chromosome* is in the Gene Ontology (GO:0005694) and the Sequence Ontology (SO:0000340). These two entries both have an attribute “xref: Wikipedia:Chromosome”[2], which points to the Wikipedia page of chromosome thus indicating that they are the same concept. This redundancy allows us to generate an “annotated” example as follows: we make the

[2] Besides Wikipedia, other knowledge bases can also be in the cross-reference field.

definition of GO:0005694 the context of an ambiguous mention, and annotate it as the concept SO:0000340. This way, we can exploit the fact that definitions (of concepts) and related concepts are described differently in different KBs, to learn the importance of words and neighboring concepts, facilitating generalization. Another resource that we leverage is the "has participant" relationship. For example, *fructose metabolic process* (GO:0006000) in the Gene Ontology has a participant *fructose* (CHEBI:28757) from the Chemical Entities ontology. This allows us to generate another "annotated" example, where we annotate the *fructose* in *fructose metabolic process* with the concept CHEBI:28757. Note that while this information usually exists across multiple KBs, it is also possible to apply this method on a single KB.

Next we describe the indirect supervision process in some more details. The first step of constructing our training examples is to cluster concepts in the KBs by the cross reference attributes and also extract all pairs of concepts that have "has participant" relations. In each concept cluster, we randomly pick one concept as the fake "mention" and the rest of concepts as the gold annotations to this mention.

5.4.1 Negative Concept Candidates

After the clustering step, we have obtained several positive concept candidates. To generate negative candidates, we apply our candidate generation method on the name of the concept which is treated as the mention, and also uniformly sample 200 concepts from all KBs. However, there is no guarantee that these candidates are really negative. Instead of using binary relevance score to train a linear ranking SVM, we take the number of common ancestors between a candidate and the positive candidates as the relevance score for the candidate. This way, if we missed a gold concept in the cluster, we won't assign it a completely irrelevant score if it has close proximity in the hierarchy of KB with other golds.

5.4.2 Feature Extraction

The local and global features we designed to capture the relatedness between a concept candidate and a mention are defined on $context(m)$ and Γ_m, which are the textual clues around the ambiguous mention m. The indirect supervision examples we have are not from any document, so there is no contextual clues. To approximate the features used at prediction time for the concept m which is treated as the mention, we use $def(m)$ to replace $context(m)$ and Γ_m is replaced by $neighbor(m)$, the neighboring concepts in the KB. By doing these approximations, we can generate features for a pair of concepts to facilitate training a ranker.

5.5 Constraints for Global Inference

At this point, we only use two types of hard constraints in Eq. (5.1) to enforce the consistency between concepts of different mentions. More specifically, a gene can be selected only if it is from a species mentioned somewhere in the document. We first form a species candidate set by gathering all concept candidates from NCBI Taxonomy[3] in a document. The assumption is that the genes mentioned in this document should be from at least one of the species in the species candidate set. Some concepts from the Protein Ontology and all concepts from the Entrez Gene Database have attributes that indicate the corresponding species, thus we design the following two constraints:

- A concept from the Entrez Gene Database must have an NCBI Taxonomy ID in the species candidate set of the document.
- If a concept from the Protein Ontology and a concept from NCBI Taxonomy have a relation "only in taxon", the concept from the Protein Ontology will be picked only if the concept from NCBI Taxonomy is in the species candidate set.

These two constraints are defined only on a single concept candidate, and we set the penalty ρ_k of them to be infinity to make these constraints hard constraints. Therefore, if a concept violates any of these two constraints, it will be excluded from the final concept list.

5.6 Experiments

In this section, we compare the proposed CCMIS with five other approaches on the CRAFT dataset. In addition, we present experimental analysis designed to evaluate the candidate generation method, features of the ranking model, the quality of indirect supervision, and the benefit of using multiple KBs.

5.6.1 Dataset

The Colorado Richly Annotated Full-Text (CRAFT) corpus Bada et al. (2012) is the largest gold standard corpus with high-quality annotations from multiple KBs: the Cell Type Ontology (CL), the Chemical Entities of Biological Interest ontology (CHEBI), the NCBI Taxonomy (NCBITaxon), the Protein Ontology (PR), the Sequence Ontology (SO), the Entrez Gene database (EG), and the Gene Ontology (GO). It identifies nearly all concepts from 67 full text of biomedical journal articles. We use the ontologies released along

[3] NCBI Taxonomy Database is a curated classification and nomenclature for all of the organisms in the public sequence databases (`http://www.ncbi.nlm.nih.gov/taxonomy`)

Ontology	#Concepts	#Anno.	#Uniq. Anno.
PR	26,879	15,593	889
NCBITaxon	789,509	7,449	149
GO	25,471	29,443	1,235
CHEBI	19,633	8,137	553
EG	17,097,474	12,266	1,021
SO	1,704	21,284	259
CL	857	5,760	155
Total	17,961,527	99,932	4,261

Table 5.1: Statistics of the concepts in the ontologies and the CRAFT corpus. We use "concepts" to refer to the entires in the ontologies, and "annotations" are concepts which are associated with mentions in the text. The second column shows the total number of concepts in each ontology. The third and fourth columns show the number of annotations and unique annotations of each ontology in the CRAFT corpus.

with the annotated documents in CRAFT-1.0 except EG which is not included in the package. We use the version which was available on October 30th, 2014. The CRAFT corpus consists of 82,634 concept mentions in total. The total number of concepts and unique concepts from each ontology is shown in Table 5.1. Note that the total number of gold annotations (99,932 the last row of the third column) is larger than the number of mentions which indicates that a mention may refer to more than one concepts across multiple ontologies. The interannotator-agreement of concept annotations is above 90% F1 score for all the ontologies Bada et al. (2012).

5.6.2 Evaluation Metrics

We mainly use the mean area under the precision-recall curve (AUC of PR-curve) (Agarwal et al., 2005) as the evaluation metric. Each mention has a ranked concept list as an output, and a set of gold concepts. We calculate the precision and recall at every ranking position, forming a PR-curve. Note that the recall is calculated using the total number of gold concepts, not just the total number of golds in the output list. This way we ensure this metric reflects the fact that some gold concepts are missing in the output. The AUC of the PR-curves of all mentions are averaged to get a final single number. We also report a hierarchical version of the AUC. The intuition is that if a concept is the parent or child of the gold concept, it should be penalized less than a concept which is far away from the gold in the hierarchy. We calculate hierarchical precision and recall using the method proposed in Kiritchenko et al. (2005), which replaces each concept by its ancestors (including itself), and then calculates the precision and recall at every ranking position by matching the ancestors of a concept candidate with the ancestors of the gold concepts. If a predicted concept has more common ancestors with the gold concepts, the score will be higher.

5.6.3 Baselines

We compare our proposed method with the following unsupervised methods.

- **TF-IDF** Cosine similarity between the TF-IDF vectors of mention context and the concept candidate's definition.
- **PageRank** Brin and Page (1998) We run the PageRank algorithm on the graph constructed from all the KBs with damping factor 0.85. This method doesn't consider any context of the ambiguous mentions at all, so a candidate concept always gets the same score, regardless of the mention it is a candidate for.
- **CollectiveInf** Zheng et al. (2015) In this method, the initial score for each concept is calculated by a modified PageRank algorithm, in which the entropy of relations are used as the edges' weights. The final score of a concept candidate is further adjusted by the matching between neighboring concepts in the KB and the concept candidates around the mention. That is, if a neighbor concept in the KBs also appears in the context of the mention, the score of the concept candidate is increased according to the initial score of the matched neighbor concept.
- **Ppr** Agirre and Soroa (2009) The Personalized PageRank algorithm implemented in the UKB package[4]. This method first inserts the context mentions into the graph as nodes, and links them with directed edges to the corresponding concept candidates. The PageRank algorithm is then applied by concentrating the initial probability mass uniformly over the mention nodes. We take a window of 30 mentions as the context. Note that in order to have a fair comparison of the disambiguation ability, we use the proposed candidate generation method in Section 5.2 to produce the confusion set for each mention.
- **Ppr_w2w** This is another variant of the Personalized PageRank algorithm and it has the best performance in Agirre and Soroa (2009). It builds a graph for each target mention and concentrates the initial probability mass in the concept candidates of the target mention. We also directly use the implementation released in the UKB package and take 30 mentions around the target mention as the context. As discussed in Agirre and Soroa (2009), the drawback of this method is its slow running time, since it performs a PageRank algorithm on the whole graph for each mention. Given the large number of mentions and the huge graph in the CRAFT dataset, we set the number of iteration of PageRank to 3 in order to make the running time tractable. It takes around 3 days on a machine with 3.0GHz CPU, whereas other approaches only need less than one hour.

[4] `http://ixa2.si.ehu.es/ukb/`

Approach	Mean AUC	Mean hAUC
TF-IDF	40.44	48.50
PageRank	42.78	50.04
CollectiveInf	35.67	42.93
Ppr	43.39	51.88
Ppr_w2w	46.51	55.46
CCMIS	**48.58**	**57.37**

Table 5.2: A comparison of CCMIS and other five unsupervised approaches on the CRAFT corpus. The evaluation metrics are mean AUC of PR-curve and its hierarchical version. CCMIS outperforms other methods significantly in both metrics (using bootstrapped t-test with p-value < 0.05

Feature	Mean AUC	Mean hAUC
PageRank	42.78	50.04
+ Local Features	45.58	54.53
+ Global Features	46.64	56.00
+ Constraints	48.58	57.37

Table 5.3: Feature ablation study of the proposed method, CCMIS. The initial ranking of candidates is according to the PageRank score. Training an indirectly supervised ranker with local and global features improves the performance by 3.3 points of mean AUC. Doing global inference with constraints improves almost 2 points overall.

5.6.4 Experimental Results

We use a public linear ranking SVM package Lee and Lin (2014) with default parameters to learn the ranking model. Feature engineering is done by doing cross validation on the indirect supervision examples, therefore, we can use all documents as the test set for all approaches. Table 5.2 shows the overall performance of each approach. The results of graph-based approaches are consistent with the results in Agirre and Soroa (2009): Ppr_w2w performs better than Ppr, and these two Personalized PageRank approaches outperform the static PageRank method. However, given the large size of KBs and the number of mentions in the CRAFT corpus, Ppr_w2w requires two days to run on a 3.0GHz CPU, whereas Ppr only takes two hours and other methods can be done within an hour. TF-IDF does not performs well since the definitions of concepts are very short and concise, which makes them hard to be matched with any context words. Our algorithm CCMIS gets 2 points higher than Ppr_w2w in terms of mean AUC, even though no additional external information is being used; specifically, no annotated document is needed to train the ranking model. Regarding the relaxed metric, mean hierarchical AUC (hAUC), the relative performance is the same but the gaps between hAUC and AUC indicate that many concept candidates are the ancestors or descendants of the gold concept, which might be proven good enough in practice.

Table 5.3 shows a feature ablation study of CCMIS. The initial ranking of candidates is according to the

Approach	$k = 0$	$k = 10$	$k = 20$	$k = 30$
TF-IDF	38.60	21.30	17.58	15.12
PageRank	40.09	21.78	20.23	19.73
CollectiveInf	33.93	16.6	12.74	11.17
Ppr	40.91	20.71	20.40	19.74
Ppr_w2w	42.58	24.56	23.13	21.21
CCMIS	**45.95**	**29.32**	**26.46**	**24.48**
Gold coverage	62.70	68.92	70.02	70.62

Table 5.4: Comparing ranking performance by changing the parameter in the candidate generation algorithm. Besides synonym matching, we use word matching to make sure each mention has at least k candidates. Note that in the setting of Table 5.2, word matching is only applied if synonym matching fails to generate any candidates. The gold coverage is the percentage of gold annotations included in the candidate list, a performance upper bound. The metric is mean AUC.

Approach	Mean AUC	Mean hAUC
CCMIS	48.58	57.34
Gold Clusters	50.85	56.50
Direct Supervision	58.98	62.59

Table 5.5: Evaluating the quality of indirectly supervised examples. The only difference between these three approaches is the way we obtain training examples. That is, only the ranking model is changed. Concept candidates, features, and learning algorithm are stay the same.

static PageRank score. Training an indirectly supervised ranker with local features adds almost 3 points of mean AUC. Without adding the two constraints to enforce species coherence, the ranking scores from our ranker already perform better than other approaches. Using these constraints adds about 1.9 points of mean AUC overall.

The candidate generation method plays an important role in getting good ranking performance. In CCMIS, we include the top 10 candidates from word matching only when synonym matching fails to generate any candidate since candidates generated by word matching are nosier. This way covers 68.11% of the gold concepts, which indicates that the ceiling of the ranking performance is close to 68.11. To show how the candidate generation method affects ranking performance we add candidates from word matching to mentions so that each mention has at least k concept candidates. The results are shown in Table 5.4. The last row of Table 5.4 shows the percentage of gold concepts included as candidates. We can see that after $k = 20$, the gold coverage merely increases. This indicates that lexical level matching is not sufficient for generating more gold concepts into the candidate set. From $k = 0$ to 10, the performance of each approach drops a lot. CCMIS is more robust when there are more irrelevant candidates. It is also interesting to see that simply increasing the gold coverage may result in worse overall performance. We need a more powerful ranking algorithm to handle larger number of candidates.

5.6.5 Quality of Indirect Supervision

We assess the quality of our indirect supervision training examples by comparing CCMIS's performance with two other approaches. These two approaches only change the way CCMIS constructs training examples for linear ranking SVM. That is, only the ranking model is changed while other components (concept candidates, features, and learning algorithm) of the system are identical.

Instead of finding concept clusters using cross reference fields and has_participant relations as in our proposed method, the first approach used the gold clusters from the mentions which have more than one gold annotation in the CRAFT corpus. Each mention forms a concept cluster in which members are the gold annotations. We conduct 5-fold cross validation on the CRAFT corpus, where gold clusters are extracted from the training documents. Note that the features of the training examples are generated in the same way as in our indirect supervision method, that is, although concept clusters are taken from documents, no text is used to generate features. This way we can focus on comparing the quality of the concept clusters obtained from the KBs with human annotation. This approach is named Gold Clusters in Table 5.5. Interestingly, its performance is better than CCMIS in terms of mean AUC but slightly worse in mean hAUC, which indicates that the concept clusters obtained by the proposed method have as good a quality as the gold annotations.

The second comparison is against direct supervision; here we use gold annotations from the CRAFT corpus itself and generate features of training examples in the same way used at prediction time. The results of 5-fold cross validation are listed in the third row of Table 5.5. Apparently, but not surprisingly, training with gold achieves about 10 points better than CCMIS. Note that in this case, the test examples, which are generated given the text documents, are expected to be more similar to the training examples, which are also generated from the text documents, in difference from the training examples used by CCMIS. This gap indicates how well the indirect supervision method approximates the distribution of the features in the test data, without using any document to obtain a good model.

5.6.6 Using KBs Individually v.s. Jointly

We compare the ranking performance of using KBs individually versus jointly. The joint case is exactly the setup of our task: grounding a given mention to multiple KBs, where the information from multiple KBs is used together. In the individual case, we only use the information from a single KB to ground concepts to this target KB, and do this for each KB. We create a dataset for each KB by splitting the annotations in the CRAFT dataset. For example, when we create the dataset for the Gene Ontology (GO), we only keep the mentions that have at least one gold annotation from GO, and remove all the other annotations. Approaches applied on this dataset can only access the information in GO. Note that we keep the candidate

Approach	Individual KBs							Joint Approach (on individual KBs)						
	PR	GO	NC	EG	CH	SO	CL	PR	GO	NC	EG	CH	SO	CL
PageRank	83.1	36.9	44.7	11.6	56.4	57.9	74.3	83.4	41.7	45.2	37.9	71.4	56.9	77.8
CollectInf	83.4	35.7	44.7	11.6	56.3	57.7	74.2	84.0	39.2	45.0	45.5	60.5	57.2	80.5
Ppr	83.2	36.5	44.7	11.6	58.0	57.8	76.1	82.9	41.3	45.1	53.9	71.2	57.2	78.9
Ppr_w2w	84.5	35.7	44.3	23.2	70.6	56.9	76.6	84.5	40.3	44.9	32.3	71.2	57.5	77.6
CCMIS	84.4	35.5	43.7	25.9	68.6	57.0	76.5	83.9	42.3	45.1	38.5	70.7	57.7	78.0

Table 5.6: A comparison between linking to each ontology individually and jointly. The evaluation metric is mean AUC of PR-curve. Note that the numbers are not directly comparable with the ones in the previous tables since the mentions in the CRAFT corpus are split into different datasets according to the annotations. For each approach and ontology, jointly using multiple KBs yields better results in most cases. The averaged performance over all datasets is summarized in Table 5.7.

generation process the same as what we do in the joint case, but only keep the concept candidates from the target ontology. Hence, we can see how does ranking performance changes given the same set of candidates. The evaluation is done on each ontology's dataset separately. We also evaluate the joint methods on each ontology separately by splitting the final ranked concepts according to their knowledge source. It allows for a fair comparison of the performance of the joint method with the individual methods. Note that the performance numbers in this section are not directly comparable to the ones in previous sections as the dataset has changed.

The results of applying different approaches to each KB's dataset are shown in Table 5.6. We can see that for each pair of (approach, ontology), using multiple KBs jointly usually yields a better result than using each KB individually. The improvement is more obvious for some ontologies, for instance, EG and CH, which contain sparse relations. In these cases, using multiple KBs together provides more information of the neighbor concepts thus may have a better match with the context of the ambiguous mention.

Table 5.7 shows the overall performance by averaging the AUC score of each mention in each ontology's dataset. CCMIS outperforms other approaches in the joint case, but has the largest gap between the individual and joint cases. The reason is that the concept clusters used to generate our training examples have worse quality and quantity within a single KB. In addition, using a single KB makes the global features sparser. This result indicates that CCMIS leverages the information across multiple knowledge bases well to achieve the best overall performance. Interestingly, Ppr_w2w achieves the best performance in the individual case. It seems that approach performs relatively well in a homogeneous network. It would be interesting to see if we can combine the power of Ppr_w2w as a feature in our ranking model (while avoiding its unrealistic computational cost).

Approach	Individual KBs	Joint Approach
PageRank	49.85	55.74
CollectiveInf	49.46	55.42
Ppr	52.12	54.88
Ppr_w2w	**52.23**	56.18
CCMIS	49.93	**57.65**

Table 5.7: The overall performance of using KBs individually and jointly. Note that the numbers are averaged AUC of mentions across different KBs' datasets, a different evaluation metric from Table 5.2. Using multiple KBs jointly always yields a better result and the gain of CCMIS is the largest.

5.7 Related Work

In the news domain, many researchers have studied ways to train a model to disambiguate concepts by directly using hyperlinks in Wikipedia documents as supervision. Earlier works Bunescu and Pasca (2006); Mihalcea and Csomai (2007) focus on local features which compare context words with the content of candidate Wikipedia pages. Later, several works Cucerzan (2007); Milne and Witten (2008); Han and Zhao (2009); Ferragina and Scaiella (2010); Ratinov et al. (2011) explore global features, trying to capture coherence among concepts that appear in close proximity in the text. Shen et al. (2012) and Dredze et al. (2010) train their model on a small manually created data set to handle documents in different domains. Cheng and Roth (2013) use relations between entities as constraints to support global inference with ranker scores, and show substantial improvement on several datasets. The main difference between our method and these Wikification approaches is that we train a ranking model by constructing indirect supervision signals from multiple KBs without using any annotated documents.

Concept Grounding and Word Sense Disambiguation (WSD) are closely related tasks as they both address the lexical ambiguity of language. Recently, several works try to relate the two by incorporating the lexical resources used in these tasks. Cholakov et al. (2014) disambiguate verbs to the senses in WordNet by creating semantic patterns from multiple lexical KBs, i.e., Wikipedia, Wikitionary, WordNet, FrameNet, and VerbNet, and also for each verb mention in the text. Moro et al. (2014) propose a graph-based approach which uses Wikipedia and WordNet as lexical resources. Their unified approach can achieve state of the art results on 6 Wikification and WSD datasets. The observation from these two papers are consistent with our conclusion that using multiple KBs jointly can improve individual tasks. Matuschek and Gurevych (2014) try to align different lexical resources (WordNet, Wikitionary, and Wikipedia in different languages). This approach is related to our construction of indirect supervision, and it would be interesting to see if the alignments could improve the quality of the indirect supervision and thus the quality of the disambiguation.

In the biomedical domain, the extensively studied word sense disambiguation problem Weeber et al.

(2001) focuses on disambiguating mentions to UMLS (Unified Medical Language System) Metathesaurus Bodenreider (2004). The main difference from our problem is that the WSD problem only addresses a small number of terms and the candidate concepts for each ambiguous mention are provided as part of the input. Researchers have developed various unsupervised methods that make use of information in the KB. McInnes (2008) compared the context words of the ambiguous mention to a profile built from UMLS concepts. Viewing the KB as a graph and adding context information into the graph, Agirre et al. (2010) compared the original PageRank algorithm with a personalized version. Jimeno-Yepes and Aronson (2010) automatically built training examples for each sense by retrieving documents from a large corpus. This approach is infeasible for our problem because we have a large amount of candidate concepts. The popular system MetaMap Aronson and Lang (2010) disambiguates mentions to semantic categories in UMLS using journal descriptor indexing. It is designed specifically for UMLS and it does not disambiguate two candidates if they are classified into the same semantic category. However, Jimeno-Yepes and Aronson (2010) showed that most of the unsupervised methods cannot even outperform the maximum frequency baseline and are not as good as the supervised methods Joshi et al. (2005); Leroy and Rindflesch (2005).

Recently, there has been a series of BioCreative challenges on gene normalization Morgan et al. (2008); Lu and Wilbur (2010); Mao et al. (2013) and chemical document indexing Krallinger et al. (2013). These tasks are closer to the problem of automatic indexing of biomedical literature, however, all these studies focus on a single KB or even a subset of it.

In our experiments we make use of the CRAFT dataset that has been studied extensively; however, most of these studies focus on mention extraction rather than disambiguating mentions. Funk et al. (2014) comprehensively compared three dictionary-based systems: MetaMap, NCBO Annotator Jonquet et al. (2009), and ConceptMapper Tanenblatt et al. (2010) and shows that the latter has the best performance. However, it only applies various string matching strategies on the surface string of the mention and the concept names in KBs, and does not attempt any disambiguation based on the context of the mentions.

5.8 Conclusion

This work studied the concept grounding problem where the target knowledge bases do not contain rich textual and structural information. We showed that we can achieve better performance than existing methods by leveraging the relations between multiple KBs. The proposed approach of constructing indirect supervision examples enables us to apply the well-studied statistical learning model even when there is no direct supervision. Inducing simple constraints to enforce solution consistency across related KBs was shown to further improve the ranking results. This work and the analysis shown suggest a range of questions from how to combine other resources to obtain higher quality of supervision, to issues of handling feature sparsity and improving the crucially important candidate generation precision.

An immediate question that follows from our work is whether (and what) other tasks can be benefit from the proposed technique. The proposed method of constructing indirect supervision examples is based on (1) Redundant information between multiple knowledge bases. The fact that duplicated concepts with different descriptions/relations appear in different KBs allows the algorithm to figure out what is important in the concept descriptions and thus provides a way to distinguish among concepts. (2) The features extracted from concept-concept pairs. These, as we show, approximate well the features of mention-concept pairs at test time. If (1) is satisfied, that is, there are multiple KBs and enough entries in them that can be aligned, then the proposed method can be applied. However, the performance of this method highly depends on (2), the quality of features and how well the indirect supervision examples approximate the text at prediction time.

An application which fits this setting is the verb sense disambiguation problem, where there are multiple sense inventories (e.g., VerbNet Kipper et al. (2000), FrameNet Baker et al. (1998), and PropBank Palmer et al. (2005)) and many of the senses are aligned by different resources (e.g., UBY Gurevych et al. (2012) and Unified Verb Index[5]). There are corpora which have annotations from one or more these verb sense inventories available, such as OntoNotes Pradhan et al. (2007) and MASC[6]. However, unlike the biomedical ontologies which have many common attributes and relatively uniform structure, different verb sense inventories vary in format and content: some resources have descriptions or example sentences of the senses, but others only have the names of semantic roles; some have relations between senses but some do not. Therefore, the question of what features would be useful in this case could be very different from those proposed in this chapter and would require additional research.

In one of the related works we mentioned in the previous section, Cholakov et al. (2014) actually utilized multiple verb sense inventories to link verbs to VerbNet. They generate a “semantic pattern” for each sense using the connections between different sense inventories, and those to each verb mention in the text. The prediction is based on the similarity between semantic patterns. Although this unsupervised method is very different from our indirect supervision approach, they confirm that using links between different sense inventories improves the performance. It would be very interesting to try our method on this problem.

[5] http://verbs.colorado.edu/verb-index/index.php

[6] https://catalog.ldc.upenn.edu/LDC2013T12

Chapter 6

Conclusion and Discussion

In this thesis, we investigate and discuss the problem of grounding entities and concepts mentioned in text to one or multiple knowledge bases, which is an essential step toward understanding natural languages for both human beings and computers. We show that this problem is very challenging especially when it is extended to the cross-lingual setting and to use the knowledge bases which are not as rich as Wikipedia is. More importantly, we show that the existing information in the knowledge bases or the datasets we already have can give us very useful clues for solving these problems. By developing algorithms and models which harvest from these incidental signals, we can achieve better performance than the existing methods which may require even more resources.

In Chapter 4, we propose models to address the three main challenges in cross-lingual wikification problem. For the first problem, multilingual NER (Chapter 4.1), we propose a language-independent model for cross-lingual NER building on a cross-lingual wikifier. We study a wide range of languages in both the monolingual and the cross-lingual settings, and show significant improvements over strong baselines. This work shows that if we can disambiguate words and phrases to the English Wikipedia, the typing information from Wikipedia categories and FreeBase are useful language-independent features for NER.

However, there is additional information in Wikipedia that could be helpful, including words in the documents and relations between titles; this would require additional research.In the future, we would like to investigate other techniques, including using comparable or parallel text and the automatic generation of training data from Wikipedia.

We show that the title mapping across languages in Wikipedia provides strong signals for both translating named entities and capturing cross-lingual similarity between words and Wikipedia titles. In Chapter 4.2, we propose a probabilistic model to learn name translation from Wikipedia titles. Using inter-language links in Wikipedia, we can collect training title pairs for more than 250 languages. The proposed model jointly considers word alignments and word transliteration, therefore it has advantage in learning location and organization names in which words are ordered differently across languages. We show that our model outperforms 6 other transliteration and translation models not only on a string similarity metric, but also on the ability of generating title candidates for the cross-lingual wikification problem.

For the smaller languages in Wikipedia, the inter-language links may not provide enough title pairs for training. In the future, we would like to investigate the techniques which discover name pairs from a collection of comparable or temporally aligned documents. These techniques could provide more examples for our name translation model thus improve the quality of the model.

In Chapter 4.3, we propose a low-resource approach to train cross-lingual word and title embeddings jointly using title alignments across Wikipedia collections in different languages. We then show that using features based on these multilingual embeddings, our wikification ranking model performs very well on a newly constructed dataset in 12 languages, and achieves state of the art also on the TAC 2015 Entity Linking dataset.

In the proposed cross-lingual embedding model, we only use the neighboring words which co-occur with the title mentions as the signals to represent the Wikipedia titles. Since Wikipedia contains rich information for each entry, we would like to investigate ways to generate better title representations using additional information such as the relationships between the titles, and words and entities which are used to introduce the titles.

The last part of this thesis (Chapter 5) studies the concept grounding problem where the target knowledge bases do not contain rich textual and structural information. We show that we can achieve better performance than existing methods by leveraging the relations between multiple KBs. The proposed approach of constructing indirect supervision examples enables us to apply the well-studied statistical learning model even when there is no direct supervision. Inducing simple constraints to enforce solution consistency across related KBs was shown to further improve the ranking results.

An immediate follow-up direction is to extend this technique to other KBs. Whether the proposed technique works well on the KBs which have even less structural information requires further research.

Bibliography

Agarwal, S., Graepel, T., Herbrich, R., and Roth, D. (2005). A large deviation bound for the area under the roc curve. In *Proc. of the Conference on Neural Information Processing Systems (NIPS)*, pages 9–16.

Agirre, E. and Soroa, A. (2009). Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–41.

Agirre, E., Soroa, A., and Stevenson, M. (2010). Graph-based word sense disambiguation of biomedical documents. *Bioinformatics*, 26(22):2889–2896.

Al-Rfou, R., Kulkarni, V., Perozzi, B., and Skiena, S. (2015). Polyglot-NER: Massive multilingual named entity recognition. In *Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, British Columbia, Canada*. SIAM.

Al-Rfou, R., Perozzi, B., and Skiena, S. (2013). Polyglot: Distributed word representations for multilingual NLP. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria. Association for Computational Linguistics.

Aronson, A. R. and Lang, F.-M. (2010). An overview of MetaMap: Historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (2000). Gene Ontology: Tool for the unification of biology. *Nature Genetics*, 25(1):25–29.

Bada, M., Eckert, M., Evans, D., Garcia, K., Shipley, K., Sitnikov, D., Baumgartner, W. A., Cohen, K. B., Verspoor, K., Blake, J. A., and Hunter, L. E. (2012). Concept annotation in the CRAFT corpus. *BMC Bioinformatics*, 13(1):161.

Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The Berkeley FrameNet project. In *Proceedings of the 17th International Conference on Computational Linguistics*, pages 86–90.

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., and Schneider, N. (2013). Abstract meaning representation for sembanking. In *Linguistic Annotation Workshop at ACL*.

Banchs, R. E., Zhang, M., Duan, X., Li, H., and Kumaran, A. (2015). Report of NEWS 2015 machine transliteration shared task. In *Proceedings of NEWS 2015 The Fifth Named Entities Workshop*, page 10.

Berant, J., Chou, A., Frostig, R., and Liang, P. (2013). Semantic parsing on freebase from question-answer pairs. In *EMNLP*, volume 2, page 6.

Bisani, M. and Ney, H. (2008). Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451.

Black, E. (1988). An experiment in computational discrimination of english word senses. *IBM Journal of research and development*, 32(2):185–194.

Bodenreider, O. (2004). The unified medical language system (UMLS): Integrating biomedical terminology. *Nucleic Acids Research*, 32:D267–D270.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1):107–117.

Bunescu, R. and Pasca, M. (2006). Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the European Chapter of the ACL (EACL)*.

Cao, Z., Qin, T., Liu, T., Tsai, M., and Li, H. (2007). Learning to rank: from pairwise approach to listwise approach. In Ghahramani, Z., editor, *Proceedings of the International Conference on Machine Learning (ICML)*, pages 129–136. Omnipress.

Chang, M.-W., Connor, M., and Roth, D. (2010a). The necessity of combining adaptation methods. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Massachusetts, USA.

Chang, M.-W., Ratinov, L., and Roth, D. (2012). Structured learning with constrained conditional models. *Machine Learning*, 88(3):399–431.

Chang, M.-W., Ratinov, L., Roth, D., and Srikumar, V. (2008a). Importance of semantic representation: Dataless classification. In *Proc. of the Conference on Artificial Intelligence (AAAI)*.

Chang, M.-W., Srikumar, V., Goldwasser, D., and Roth, D. (2010b). Structured output learning with indirect supervision. In *Proc. of the International Conference on Machine Learning (ICML)*.

Chang, P.-C., Galley, M., and Manning, C. D. (2008b). Optimizing Chinese word segmentation for machine translation performance. In *Proceedings of the third workshop on statistical machine translation, Association for Computational Linguistics*.

Chapelle, O. and Keerthi, S. S. (2010). Efficient algorithms for ranking with SVMs. *Information Retrieval*, 13(3):201–215.

Chen, Z. and Ji, H. (2011). Collaborative ranking: A case study on entity linking. In *EMNLP*, pages 771–781.

Cheng, X. and Roth, D. (2013). Relational inference for wikification. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Chisholm, A. and Hachey, B. (2015). Entity disambiguation with web links. *TACL*, 3:145–156.

Cholakov, K., Eckle-Kohler, J., and Gurevych, I. (2014). Automated verb sense labelling based on linked lexical resources. In *EACL*, pages 68–77.

Chung, J., Cho, K., and Bengio, Y. (2016). A character-level decoder without explicit segmentation for neural machine translation. *arXiv preprint arXiv:1603.06147*.

Clarke, J., Goldwasser, D., Chang, M.-W., and Roth, D. (2010). Driving semantic parsing from the world's response. In *Proc. of the Conference on Computational Natural Language Learning (CoNLL)*.

Collins, M. (2002). Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*.

Cottrell, G. W. (1985). A connectionist approach to word sense disambiguation.

Coulmance, J., Marty, J.-M., Wenzek, G., and Benhalloum, A. (2015). Trans-gram, fast cross-lingual word-embeddings. In *Proceedings of EMNLP*.

Cucerzan, S. (2007). Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of EMNLP*, pages 708–716.

Das, D. and Petrov, S. (2011). Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Portland, OR. Association for Computational Linguistics.

Daumé, H. (2007). Frustratingly easy domain adaptation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 256–263, Prague, Czech Republic. Association for Computational Linguistics.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39:1–38.

Dredze, M., McNamee, P., Rao, D., Gerber, A., and Finin, T. (2010). Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 277–285.

Duong, L., Cohn, T., Verspoor, K., Bird, S., and Cook, P. (2014). What can we get from 1000 tokens? A case study of multilingual POS tagging for resource-poor languages. In *EMNLP*, pages 886–897. Citeseer.

Dyer, C., Chahuneau, V., and Smith, N. A. (2013). A simple, fast, and effective reparameterization of ibm model 2. Association for Computational Linguistics.

Ehrmann, M., Turchi, M., and Steinberger, R. (2011). Building a multilingual named entity-annotated corpus using annotation projection. In *RANLP*.

Escudero, G., Màrquez, L., Rigau, G., and Salgado, J. G. (2000). On the portability and tuning of supervised word sense disambiguation systems.

Faruqui, M. and Dyer, C. (2014). Improving vector space word representations using multilingual correlation. Association for Computational Linguistics.

Ferragina, P. and Scaiella, U. (2010). TAGME: On-the-fly annotation of short text fragments (by Wikipedia entities). In *Proceedings of CIKM*, pages 1625–1628.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 363–370, Morristown, NJ, USA. Association for Computational Linguistics.

Firth, J. R. (1957). A synopsis of linguistic theory 1930-1955. pages 1–32.

Francis-Landau, M., Durrett, G., and Klein, D. (2016). Capturing semantic similarity for entity linking with convolutional neural networks. *NAACL*.

Funk, C., Baumgartner, W., Garcia, B., Roeder, C., Bada, M., Cohen, K. B., Hunter, L. E., and Verspoor, K. (2014). Large-scale biomedical concept recognition: An evaluation of current automatic annotators and their parameters. *BMC Bioinformatics*, 15(1):59.

Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *IJCAI*.

Ganea, O.-E., Ganea, M., Lucchi, A., Eickhoff, C., and Hofmann, T. (2016). Probabilistic bag-of-hyperlinks model for entity linking. In *WWW*, pages 927–938.

Gattani, A., Lamba, D. S., Garera, N., Tiwari, M., Chai, X., Das, S., Subramaniam, S., Rajaraman, A., Harinarayan, V., and Doan, A. (2013). Entity extraction, linking, classification, and tagging for social media: a wikipedia-based approach. *VLDB Endowment*, 6(11):1126–1137.

Globerson, A., Lazic, N., Chakrabarti, S., Subramanya, A., Ringgaard, M., and Pereira, F. (2016). Collective entity resolution with multi-focal attention. pages 621–631.

Goldwasser, D., Chang, M.-W., Tu, Y., and Roth, D. (2009). Constraint driven transliteration discovery. In Nicolov, N., editor, *Proc. of the Conference on Recent Advances in Natural Language Processing*. John Benjamins.

Gottipati, S. and Jiang, J. (2011). Linking entities to a knowledge base with query expansion. In *EMNLP*, pages 804–813. Association for Computational Linguistics.

Gouws, S., Bengio, Y., and Corrado, G. (2014). BILBOWA: Fast bilingual distributed representations without word alignments. In *Deep Learning Workshop, NIPS*.

Grishman, R. and Sundheim, B. (1996). Message understanding conference-6: A brief history. In *Coling*, volume 96, pages 466–471.

Guo, S., Chang, M.-W., and Kiciman, E. (2013). To link or not to link? A study on end-to-end tweet entity linking. In *NAACL*, pages 1020–1030.

Gurevych, I., Eckle-Kohler, J., Hartmann, S., Matuschek, M., Meyer, C. M., and Wirth, C. (2012). UBY: A large-scale unified lexical-semantic resource based on LMF. In *Proceedings of EACL*, pages 580–590.

Guthrie, J. A., Guthrie, L., Wilks, Y., and Aidinejad, H. (1991). Subject-dependent co-occurrence and word sense disambiguation. In *Proceedings of the 29th Annual Meeting on Association for Computational Linguistics*, pages 146–152. Association for Computational Linguistics.

Hachey, B., Radford, W., Nothman, J., Honnibal, M., and Curran, J. R. (2013). Evaluating entity linking with wikipedia. *Artificial intelligence*, 194:130–150.

Han, X. and Sun, L. (2011). A generative entity-mention model for linking entities with knowledge base. In *ACL*, pages 945–954. Association for Computational Linguistics.

Han, X. and Sun, L. (2012). An entity-topic model for entity linking. In *EMNLP*, pages 105–115.

Han, X., Sun, L., and Zhao, J. (2011). Collective entity linking in web text: a graph-based method. In *SIGIR*, pages 765–774.

Han, X. and Zhao, J. (2009). Named entity disambiguation by leveraging Wikipedia semantic knowledge. In *Proceedings of CIKM*, pages 215–224.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

He, Z., Liu, S., Li, M., Zhou, M., Zhang, L., and Wang, H. (2013). Learning entity representation for entity disambiguation. In *ACL*, pages 30–34.

Herbrich, R., Graepel, T., and Obermayer, K. (2000). Large margin rank boundaries for ordinal regression.

Hermann, K. M. and Blunsom, P. (2014). Multilingual distributed representations without word alignment. In *Proceedings of ICLR*.

Hoffart, J., Seufert, S., Nguyen, D. B., Theobald, M., and Weikum, G. (2012). KORE: Keyphrase overlap relatedness for entity disambiguation. In *CIKM*, pages 545–554. ACM.

Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., and Weikum, G. (2011). Robust disambiguation of named entities in text. In *EMNLP*, pages 782–792.

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, pages 321–377.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2006). OntoNotes: The 90% solution. In *Proceedings of HLT/NAACL*, New York.

Hwa, R., Resnik, P., Weinberg, A., Cabezas, C. I., and Kolak, O. (2005). Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11:311–325.

Ide, N. and Véronis, J. (1998). Introduction to the special issue on word sense disambiguation: the state of the art. *Computational linguistics*, 24(1):2–40.

Irvine, A., Callison-Burch, C., and Klementiev, A. (2010). Transliterating from all languages. In *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*.

Ji, H., Grishman, R., and Dang, H. T. (2011). Overview of the TAC2011 knowledge base population track. In *Text Analysis Conference (TAC)*.

Ji, H., Grishman, R., Dang, H. T., Griffitt, K., and Ellis, J. (2010). Overview of the TAC 2010 knowledge base population track. In *Text Analysis Conference (TAC)*.

Ji, H., Nothman, J., and Dang, H. T. (2016). Overview of TAC-KBP2016 tri-lingual EDL and its impact on end-to-end cold-start KBP. In *Text Analysis Conference (TAC)*.

Ji, H., Nothman, J., and Hachey, B. (2014). Overview of TAC-KBP2014 entity discovery and linking tasks. In *Text Analysis Conference (TAC)*.

Ji, H., Nothman, J., Hachey, B., and Florian, R. (2015). Overview of TAC-KBP2015 tri-lingual entity discovery and linking. In *Text Analysis Conference (TAC)*.

Jiampojamarn, S., Cherry, C., and Kondrak, G. (2008). Joint processing and discriminative training for letter-to-phoneme conversion. In *ACL*, pages 905–913.

Jiampojamarn, S., Kondrak, G., and Sherif, T. (2007). Applying many-to-many alignments and hidden markov models to letter-to-phoneme conversion. In *NAACL*.

Jimeno-Yepes, A. J. and Aronson, A. R. (2010). Knowledge-based biomedical word sense disambiguation: Comparison of approaches. *BMC Bioinformatics*, 11(1):569.

Joachims, T. (2002). Unbiased evaluation of retrieval quality using clickthrough data. In *SIGIR Workshop on Mathematical/Formal Methods in Information Retrieval*.

Joachims, T. (2006). Training linear SVMs in linear time. In *Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD)*.

Jonquet, C., Shah, N., and Musen, M. (2009). The open biomedical annotator. *Summit on Translational Bioinformatics*, 2009:56.

Joshi, M., Pedersen, T., and Maclin, R. (2005). A comparative study of support vector machines applied to the supervised word sense disambiguation problem in the medical domain. In *Proceedings of IICAI*, pages 3449–3468.

Kazama, J. and Torisawa, K. (2007). Exploiting Wikipedia as external knowledge for named entity recognition. In *Proceedings of the 2007 Joint Conference of EMNLP-CoNLL*, pages 698–707.

Kelly, E. F. and Stone, P. J. (1975). *Computer recognition of English word senses*, volume 13. North-Holland.

Kim, S., Toutanova, K., and Yu, H. (2012). Multilingual named entity recognition using parallel data and metadata from Wikipedia. In *ACL*.

Kipper, K., Dang, H. T., and Palmer, M. (2000). Class-based construction of a verb lexicon. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Austin, TX. AAAI.

Kiritchenko, S., Matwin, S., and Famili, F. (2005). Functional annotation of genes using hierarchical text categorization. In *Proceedings of BioLINK SIG: Linking Literature, Information and Knowledge for Biology*.

Klementiev, A. and Roth, D. (2006). Named entity transliteration and discovery from multilingual comparable corpora. In *Proc. of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 82–88.

Klementiev, A. and Roth, D. (2008). Named entity transliteration and discovery in multilingual corpora. In Goutte, C., Cancedda, N., Dymetman, M., and Foster, G., editors, *Learning Machine Translation*. MIT Press.

Krallinger, M., Leitner, F., Rabal, O., Vazquez, M., Oyarzabal, J., and Valencia, A. (2013). Overview of the chemical compound and drug name recognition (CHEMDNER) task. In *BioCreative Challenge Evaluation Workshop*, volume 2, page 2.

Kulkarni, S., Singh, A., Ramakrishnan, G., and Chakrabarti, S. (2009). Collective annotation of wikipedia entities in web text. In *KDD*, pages 457–466. ACM.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. In *NAACL*.

Lazic, N., Subramanya, A., Ringgaard, M., and Pereira, F. (2015). Plato: A selective context model for entity resolution. *TACL*, 3:503–515.

Lee, C.-P. and Lin, C.-J. (2014). Large-scale linear RankSVM. *Neural computation*, 26(4):781–817.

Leroy, G. and Rindflesch, T. C. (2005). Effects of information and machine learning algorithms on word sense disambiguation with small datasets. *International Journal of Medical Informatics*, 74(7):573–585.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, pages 24–26. ACM.

Li, H., Kumaran, A., Pervouchine, V., and Zhang, M. (2009). Report of NEWS 2009 machine transliteration shared task. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*, pages 1–18.

Li, H., Kumaran, A., Zhang, M., and Pervouchine, V. (2010). Report of NEWS 2010 transliteration generation shared task. In *Proceedings of the 2010 Named Entities Workshop*, pages 1–11.

Li, Y., Wang, C., Han, F., Han, J., Roth, D., and Yan, X. (2013). Mining evidences for named entity disambiguation. In *SIGKDD*, pages 1070–1078. ACM.

Ling, X., Singh, S., and Weld, D. S. (2015). Design challenges for entity linking. *TACL*, 3:315–328.

Liu, L., Finch, A., Utiyama, M., and Sumita, E. (2016). Agreement on target-bidirectional lstms for sequence-to-sequence learning. In *AAAI*.

Liu, X., Li, Y., Wu, H., Zhou, M., Wei, F., and Lu, Y. (2013). Entity linking for tweets. In *ACL*, pages 1304–1311.

Lu, Z. and Wilbur, W. J. (2010). Overview of BioCreative III gene normalization. In *Proceedings of the BioCreative III Workshop*, pages 24–45.

Luo, G., Huang, X., Lin, C.-Y., and Nie, Z. (2015). Joint named entity recognition and disambiguation. In *EMNLP*, pages 879–888.

Luong, M.-T., Pham, H., and Manning, C. D. (2015). Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*.

Mao, Y., Auken, K. V., Li, D., Arighi, C. N., and Lu, Z. (2013). The gene ontology task at BioCreative IV. In *Proceedings of the Fourth Biocreative Challenge Evaluation Workshop*, volume 1, pages 119–127.

Matuschek, M. and Gurevych, I. (2014). High performance word sense alignment by joint modeling of sense distance and gloss similarity. In *Proceedings of COLING*, pages 245–256.

McDonald, R., Petrov, S., and Hall, K. (2011). Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pages 62–72, Edinburgh, Scotland, UK. Association for Computational Linguistics.

McInnes, B. T. (2008). An unsupervised vector approach to biomedical term disambiguation: Integrating UMLS and Medline. In *Proceedings of ACL: Student Research Workshop*, pages 49–54.

McNamee, P. and Dang, H. T. (2009). Overview of the TAC 2009 knowledge base population track. In *Text Analysis Conference (TAC)*, volume 17, pages 111–113.

McNamee, P., Mayfield, J., Lawrie, D., Oard, D. W., and avid. S. Doermann (2011). Cross-language entity linking. In *Proceedings of IJCNLP*, pages 255–263.

Mihalcea, R. (2004). Co-training and self-training for word sense disambiguation. In *CoNLL*, pages 33–40.

Mihalcea, R. (2007). Using Wikipedia for automatic word sense disambiguation. In *NAACL*, pages 196–203.

Mihalcea, R. and Csomai, A. (2007). Wikify!: Linking documents to encyclopedic knowledge. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 233–242.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. In *Workshop at ICLR*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Miller, G., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. (1990). Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–312.

Milne, D. and Witten, I. H. (2008). Learning to link with Wikipedia. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 509–518.

Minsky, M. and Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry*. The MIT Press.

Mooney, R. J. (1996). Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. *arXiv preprint cmp-lg/9612001*.

Morgan, A. A., Lu, Z., Wang, X., Cohen, A. M., Fluck, J., Ruch, P., Divoli, A., Fundel, K., Leaman, R., Hakenberg, J., Sun, C., Liu, H.-H., Torres, R., Krauthammer, M., Lau, W. W., Liu, H., Hsu, C.-N., Schuemie, M., Cohen, K. B., and Hirschman, L. (2008). Overview of BioCreative II gene normalization. *Genome Biology*, 9:S3.

Moro, A., Raganato, A., and Navigli, R. (2014). Entity linking meets word sense disambiguation: A unified approach. In *Transactions of the Association for Computational Linguistics*, volume 2, pages 231–244.

Murata, M., Utiyama, M., Uchimoto, K., Ma, Q., and Isahara, H. (2001). Japanese word sense disambiguation using the simple bayes and support vector machine methods. In *The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 135–138. Association for Computational Linguistics.

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

Ng, A. Y. and Jordan, M. I. (2001). On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *The Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 841–848.

Nguyen, D. B., Theobald, M., and Weikum, G. (2016). J-NERD: Joint named entity recognition and disambiguation with rich linguistic features. *TACL*, 4:215–229.

Noreen, E. W. (1989). *Computer-intensive methods for testing hypotheses*. Wiley New York.

Nothman, J., Ringland, N., Radford, W., Murphy, T., and Curran, J. R. (2012). Learning multilingual named entity recognition from Wikipedia. *Artificial Intelligence*, 194:151–175.

Palmer, M., Gildea, D., and Kingsbury, P. (2005). The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Pan, X., Cassidy, T., Hermjakob, U., Ji, H., and Knight, K. (2015). Unsupervised entity linking with abstract meaning representation. In *NAACL*, pages 1130–1139.

Pasternack, J. and Roth, D. (2009). Learning better transliterations. In *Proc. of the ACM Conference on Information and Knowledge Management (CIKM)*.

Pershina, M., He, Y., and Grishman, R. (2015). Personalized page rank for named entity disambiguation. In *NAACL*, pages 238–243.

Pilz, A. and Paaß, G. (2011). From names to entities using thematic context distance. In *CIKM*, pages 857–866.

Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods: support vector learning*.

Pradhan, S. S., Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2007). Ontonotes: A unified relational semantic representation. *International Journal of Semantic Computing*, 01(04):405–419.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285.

Ratinov, L. and Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *Proc. of the Conference on Computational Natural Language Learning (CoNLL)*.

Ratinov, L., Roth, D., Downey, D., and Anderson, M. (2011). Local and global algorithms for disambiguation to wikipedia. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Roth, D. (2017). Incidental supervision: Moving beyond supervised learning. In *Proc. of the Conference on Artificial Intelligence (AAAI)*.

Roth, D. and tau Yih, W. (2004). A linear programming formulation for global inference in natural language tasks. In Ng, H. T. and Riloff, E., editors, *Proc. of the Conference on Computational Natural Language Learning (CoNLL)*, pages 1–8. Association for Computational Linguistics.

Sarawagi, S. and Cohen, W. W. (2004). Semi-markov conditional random fields for information extraction. In *The Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 1185–1192.

Sculley, D. (2009). Large scale learning to rank. In *NIPS Workshop on Advances in Ranking*, pages 58–63.

Sennrich, R., Haddow, B., and Birch, A. (2015). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Shen, W., Wang, J., and Han, J. (2015). Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460.

Shen, W., Wang, J., Luo, P., and Wang, M. (2012). LINDEN: Linking named entities with knowledge base via semantic knowledge. In *Proceedings of WWW*, pages 449–458.

Shi, T., Liu, Z., Liu, Y., and Sun, M. (2015). Learning cross-lingual word embeddings via matrix co-factorization. In *Proceedings of ACL*.

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

Simpson, H., Cieri, C., Maeda, K., Baker, K., and Onyshkevych, B. (2008). Human language technology resources for less commonly taught languages: Lessons learned toward creation of basic language resources. *Collaboration: interoperability between people in the creation of language resources for less-resourced languages*, page 7.

Smith, J. R., Quirk, C., and Toutanova, K. (2010). Extracting parallel sentences from comparable corpora using document level alignment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 403–411. Association for Computational Linguistics.

Song, Y. and Roth, D. (2014). On dataless hierarchical text classification. In *Proc. of the Conference on Artificial Intelligence (AAAI)*.

Soyer, H., Stenetorp, P., and Aizawa, A. (2015). Leveraging monolingual data for crosslingual compositional word representations. In *Proceedings of ICLR*.

Spitkovsky, V. I. and Chang, A. X. (2012). A cross-lingual dictionary for english wikipedia concepts. In *LREC*, pages 3168–3175.

Täckström, O. (2012). Nudging the envelope of direct transfer methods for multilingual named entity recognition. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, pages 55–63. Association for Computational Linguistics.

Täckström, O., McDonald, R. T., and Uszkoreit, J. (2012). Cross-lingual word clusters for direct transfer of linguistic structure. In *NAACL*.

Tanenblatt, M. A., Coden, A., and Sominsky, I. L. (2010). The ConceptMapper approach to named entity recognition. In *Proceedings of LREC*.

Tao, T., Yoon, S.-Y., Fister, A., Sproat, R., and Zhai, C. (2006). Unsupervised named entitly transliteration using temporal and phonetic correlation. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pages 250–257.

Tjong Kim Sang, E. F. (2002). Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2002*, pages 155–158. Taipei, Taiwan.

Tjong Kim Sang, E. F. and De Meulder, F. (2003a). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In Daelemans, W. and Osborne, M., editors, *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada.

Tjong Kim Sang, E. F. and De Meulder, F. (2003b). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In Daelemans, W. and Osborne, M., editors, *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada.

Tsai, C.-T., Mayhew, S., Peng, H., Sammons, M., Mangipundi, B., Reddy, P., and Roth, D. (2016). Illinois CCG entity discovery and linking, event nugget detection and co-reference, and slot filler validation systems for tac 2016. In *Text Analysis Conference (TAC)*.

Tsai, C.-T. and Roth, D. (2016a). Concept grounding to multiple knowledge bases via indirect supervision.

Tsai, C.-T. and Roth, D. (2016b). Cross-lingual wikification using multilingual embeddings. In *Proc. of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Veronis, J. and Ide, N. M. (1990). Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. In *Proceedings of the 13th conference on Computational linguistics-Volume 2*, pages 389–394. Association for Computational Linguistics.

Vulić, I. and Moens, M.-F. (2015). Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. In *Proceedings of ACL*.

Wang, H., Zheng, J. G., Ma, X., Fox, P., and Ji, H. (2015). Language and domain independent entity linking with quantified collective validation. In *EMNLP*.

Wang, M. and Manning, C. D. (2014). Cross-lingual projected expectation regularization for weakly supervised learning. In *TACL*.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014). Knowledge graph and text jointly embedding. In *EMNLP*.

Weaver, W. (1955). Translation. *Machine translation of languages*, 14:15–23.

Weeber, M., Mork, J. G., and Aronson, A. R. (2001). Developing a test collection for biomedical word sense disambiguation. In *Proceedings of the AMIA Symposium*, page 746.

Wu, F. and Weld, D. S. (2010). Open information extraction using Wikipedia. In *ACL*.

Yamada, I., Shindo, H., Takeda, H., and Takefuji, Y. (2016). Joint learning of the embedding of words and entities for named entity disambiguation. *arXiv preprint arXiv:1601.01343*.

Yarowsky, D. (1992). Word sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings the International Conference on Computational Linguistics (COLING)*, pages 454–460.

Yarowsky, D. (1994). Decision lists for lexical ambiguity resolution: application to accent restoration in Spanish and French. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 88–95.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervied methods. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yarowsky, D., Ngai, G., and Wicentowski, R. (2001). Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the first international conference on Human language technology research*, pages 1–8. Association for Computational Linguistics.

Yoon, S.-Y., Kim, K.-Y., and Sproat, R. (2007). Multilingual transliteration using feature based phonetic method. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 112–119, Prague, Czech Republic. Association for Computational Linguistics.

Zhang, B., Pan, X., Wang, T., Vaswani, A., Ji, H., Knight, K., and Marcu, D. (2016). Name tagging for low-resource incident languages based on expectation-driven learning. In *NAACL*.

Zhang, W., Sim, Y.-C., Su, J., and Tan, C.-L. (2011). Entity linking with effective acronym expansion, instance selection and topic modeling. In *IJCAI*.

Zhang, W., Su, J., Tan, C. L., and Wang, W. T. (2010). Entity linking leveraging automatically generated annotation. In *COLING*, pages 1290–1298. Association for Computational Linguistics.

Zheng, J. G., Howsmon, D., Zhang, B., Hahn, J., McGuinness, D., Hendler, J., and Ji, H. (2015). Entity linking for biomedical literature. *BMC Medical Informatics and Decision Making*, 15(Suppl 1):S4.

Zheng, Z., Li, F., Huang, M., and Zhu, X. (2010). Learning to link entities with knowledge base. In *NAACL*, pages 483–491. Association for Computational Linguistics.