2019

# Exploiting Cross-Lingual Representations For Natural Language Processing

Shyam Upadhyay
*University of Pennsylvania*, shyamupa@gmail.com

# Exploiting Cross-Lingual Representations For Natural Language Processing

**Abstract**

Traditional approaches to supervised learning require a generous amount of labeled data for good generalization. While such annotation-heavy approaches have proven useful for some Natural Language Processing (NLP) tasks in high-resource languages (like English), they are unlikely to scale to languages where collecting labeled data is difficult and time-consuming. Translating supervision available in English is also not a viable solution, because developing a good machine translation system requires expensive to annotate resources which are not available for most languages.

In this thesis, I argue that cross-lingual representations are an effective means of extending NLP tools to languages beyond English without resorting to generous amounts of annotated data or expensive machine translation. These representations can be learned in an inexpensive manner, often from signals completely unrelated to the task of interest. I begin with a review of different ways of inducing such representations using a variety of cross-lingual signals and study algorithmic approaches of using them in a diverse set of downstream tasks. Examples of such tasks covered in this thesis include learning representations to transfer a trained model across languages for document classification, assist in monolingual lexical semantics like word sense induction, identify asymmetric lexical relationships like hypernymy between words in different languages, or combining supervision across languages through a shared feature space for cross-lingual entity linking. In all these applications, the representations make information expressed in other languages available in English, while requiring minimal additional supervision in the language of interest.

**Degree Type**
Dissertation

**Degree Name**
Doctor of Philosophy (PhD)

**Graduate Group**
Computer and Information Science

**First Advisor**
Dan Roth

**Keywords**
cross-lingual, low supervision, multilingual, natural language processing, representation learning

**Subject Categories**
Computer Sciences

# EXPLOITING CROSS-LINGUAL REPRESENTATIONS FOR NATURAL LANGUAGE PROCESSING

Shyam Upadhyay

A DISSERTATION

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2019

Supervisor of Dissertation

Dan Roth, Professor of Computer and Information Science

Graduate Group Chairperson

Rajeev Alur, Professor of Computer and Information Science

Dissertation Committee

Adam Kalai, Principal Researcher, Microsoft Research

Chris Callison-Burch, Professor of Computer and Information Science

Lyle Ungar, Professor of Computer and Information Science

Mitchell P. Marcus, Professor of Computer and Information Science

EXPLOITING CROSS-LINGUAL REPRESENTATIONS
FOR NATURAL LANGUAGE PROCESSING

*Dedicated to Maa and Papa.*

# ACKNOWLEDGEMENT

*Nothing of me is original. I am the combined effort of everybody I've ever known.*

Chuck Palahniuk, Invisible Monsters

Very few Ph.D. students get the opportunity of working with a great advisor like Dan. Without his continuous support, patience and his open style of advising that grants research freedom, most of the work in this thesis would not have seen the light of day. Besides being a great advisor, Dan is one of the kindest and most considerate person I know. I am immensely grateful for all that he has done for me. I am also indebted to my thesis committee — Adam, Mitch, Lyle, and Chris. Not only have they provided valuable feedback on the thesis at various stages, but also have been a source of sage advice.

During my Ph.D., I also got the valuable opportunity to work with some of the best researchers in the field through internships at Microsoft Research and Google. Working with Ming-Wei and Scott was instrumental in shaping my thinking and temperament for research. The wisdom I got from Ming-Wei will be invaluable for the rest of my research life. The next summer at MSR Cambridge was equally enjoyable, where Adam, Matt, Kai-Wei, and James gave me the freedom to define and work on a research problem, and helping me find a way towards a solution. Lastly, the summer spent working with Gokhan, Dilek and Larry taught me how to approach problems outside my comfort zone, and keeping in mind practical considerations when thinking about new ideas. I am thankful for all these experiences, which have shaped my outlook and approach towards research.

I would like to thank former and current members of CogComp — especially Mark Sammons, Kai-Wei Chang, Rajhans Samdani, and Gourab Kundu, who helped me fit in the group in my early days. Working in CogComp also brought me in contact with Christos, Parisa, Michael, Snigdha, and Wenpeng, who took the trouble of mentoring me when they were themselves wrestling with hard research problems. Thanks to Christos and Michael for giving me research advice at different stages of my Ph.D., and constantly reminding me "not to work too hard over the weekends". I am also indebted to Yangqiu Song, who really did all the hard work that later became Chapter 4. Special thanks to Snigdha, who was influential in helping me frame the early draft of this thesis.

During my stay at UIUC and UPenn, I developed close friends in brilliant fellow students like Subhro, Haoruo, Nitish, Daniel, John, Chen-Tse, Stephen, Dan (Deutsch), Qiang, Anne, Jordan, Reno, João, Daphne, and others. When I reflect how much I have learned from all of you over the years, it amazes me. I am sure you will have prolific and satisfying careers ahead of you. I will dearly miss the discussions over lunches and happy hours we had. I will especially miss the evening meals I had with Subhro, Nitish, Snigdha, and Shashank. The not-so-good Indian food in the US will taste even worse without your company.

I would be remiss in not thanking my remote collaborators — Manaal, Chris (Dyer), Yogarshi, and Marine. Working with all of you was a unique experience and I have acquired countless skills and research wisdom from our interactions. Special thanks to Manaal, who I often trouble for research and career advice from time to time.

It would be unfair not to thank the folks behind the scenes who ensure things run smoothly in CogComp at UIUC and UPenn. Thanks to Dawn Cheek, Eric Horn and Jennifer Sheffield, who often went out of their way to help students like me.

Lastly, I would like to thank Maa, Papa and everyone in my family whose efforts and sacrifices got me here. Thanks for your unconditional love and support when things were not going well, and uplifting my spirits when I was facing the despair that naturally accompanies the research process. This is for you.

# ABSTRACT

## EXPLOITING CROSS-LINGUAL REPRESENTATIONS FOR NATURAL LANGUAGE PROCESSING

Shyam Upadhyay

Dan Roth

Traditional approaches to supervised learning require a generous amount of labeled data for good generalization. While such annotation-heavy approaches have proven useful for some Natural Language Processing (NLP) tasks in high-resource languages (like English), they are unlikely to scale to languages where collecting labeled data is difficult and time-consuming. Translating supervision available in English is also not a viable solution, because developing a good machine translation system requires expensive to annotate resources which are not available for most languages.

In this thesis, I argue that cross-lingual representations are an effective means of extending NLP tools to languages beyond English without resorting to generous amounts of annotated data or expensive machine translation. These representations can be learned in an inexpensive manner, often from signals completely unrelated to the task of interest. I begin with a review of different ways of inducing such representations using a variety of cross-lingual signals and study algorithmic approaches of using them in a diverse set of downstream tasks. Examples of such tasks covered in this thesis include learning representations to transfer a trained model across languages for document classification, assist in monolingual lexical semantics like word sense induction, identify asymmetric lexical relationships like hypernymy between words in *different* languages, or combining supervision across languages through a shared feature space for cross-lingual entity linking. In all these applications, the representations make information expressed in other languages available in English, while requiring minimal additional supervision in the language of interest.

# Contents

# List of Tables

# List of Figures

# CHAPTER 1 : Introduction

## 1.1. Overview

Most work in Natural Language Processing (NLP) is driven by the availability of supervision in the form of labeled data. Such labeled data is relatively easily available for simple NLP tasks in high-resource languages (like document classification for English), but this is not true for other languages. In fact, for more sophisticated NLP tasks like semantic parsing, fully labeled data is hard to come by even in English, let alone other languages. Consequently, even though there are over 7000 languages from over 140 different language families in use across the world (Lewis, 2009), the state of NLP is skewed towards high-resource languages like English.

*Why should one care about NLP in other languages?* Developing NLP technology that can operate multilingually has several benefits. A large population of the world is multilingual, and produces information at a unprecedented scale across several languages thanks to the web. Indeed, over 40% of the content on the web is not in English, and the rate at which such content is generated has been growing dramatically.[1] Although most of this content is publicly available (online news, social media platforms etc.), yet consuming it is complicated by the multitude of languages used. Multilingual NLP makes this information available to a broader audience than what the language in which it was expressed can reach, helping overcome the metaphorical language-imposed barrier to knowledge. For instance, knowing what entities appear in a certain tweet written in Hindi can improve the understanding of an English reader. This can help faster dissemination of critical information, that can prove crucial in urgent situations (e.g., early warning systems, financial trading). Multilingual NLP can also help from a scientific standpoint, in understanding how languages evolve and interact with each other, revealing previously unknown similarities or differences between languages (Bender, 2009, 2011).

[1] Based on `https://en.wikipedia.org/wiki/Languages_used_on_the_Internet`.

Unfortunately, traditional supervised learning approaches, which form the backbone of current NLP technology, are inherently ill-equipped to deal with the lack of labeled data, which poses a significant challenge in scaling to other languages. As a result, advances made in NLP for high-resource languages take longer to percolate to the rest of the languages. To bridge this language barrier, efforts have been made in two directions.

**Annotation-heavy Approaches.** The first direction is motivated by the overwhelming success of supervised machine learning approaches for English NLP tasks. Given enough training data, traditional machine learning approaches have shown remarkable generalization abilities. One natural step is to apply these approaches to NLP tasks in non-English languages, by providing supervision in the target language. Often, such supervision is derived from multilingual resources that are used in machine translation, such as parliamentary proceedings (Koehn, 2005) or translations of the Bible (Christodouloupoulos and Steedman, 2015; Agić et al., 2015). A popular approach for achieving this is *annotation projection*, where annotation for some linguistic structure is projected from a high resource language to a low resource language, usually by means of some cross-lingual alignment (Yarowsky and Ngai, 2001; Hwa et al., 2005). Nevertheless, the community has also invested in curating annotated resources in non-English languages for various NLP tasks (Xue, 2008; de Melo and Weikum, 2009; McDonald et al., 2013), so that traditional supervised learning approaches can be developed for these languages. Indeed, statistical machine learning approaches have become integral to advancing NLP in English in past three decades, and it is reasonable to assume that they will succeed for other languages once the appropriate datasets become available.

**Translation-based Approaches.** The second direction has advocated extending NLP to languages other than English by automatically translating text in the target language to English using machine translation, and then using state-of-the-art NLP models already developed for English. Like annotation-heavy approaches, this direction also relies on the success of the statistical learning approaches for English NLP tasks, with the hope that good translation from a language to English will be sufficient to bridge the language barrier. Indeed, translation quality from a non-English language to English has improved considerably over the past few years with the advent of neural machine translation (Sutskever et al., 2014; Bahdanau et al., 2015; Luong et al., 2015a; Sennrich et al., 2016, inter alia).

While promising, both of these directions suffer from limitations. Traditional supervised learning approaches require generous amount of labeled data to achieve good generalization. While such annotation-heavy approaches have proved successful for high resource languages such as English, their applicability to a new language is limited due to lack of labeled data. Collecting high-quality labelled data for any NLP task in an arbitrary language is challenging due to the human effort involved and the paucity of expert annotators. There are practical limitations as well — generating high-quality annotations such as treebanks (Marcus et al., 1993; Prasad et al., 2008; McDonald et al., 2013; Socher et al., 2013) takes time and expertise. This may not be ideal if one has limited time to develop a system in a language to achieve certain goals (e.g., for disaster management).

Similar problems affect machine translation. Building a translation system requires resources in the target language (namely large parallel corpora with English), which are not always available and are expensive to annotate at scale (Lopez and Post, 2013). Unless several million lines of in-domain parallel text are available, statistical machine translation approaches perform poorly (Kolachina et al., 2012; Irvine, 2014), an issue that is exacerbated for the now popular neural machine translation approaches (Koehn and Knowles, 2017). From a practical standpoint too, including a translation system in the pipeline may prove to be an overkill for simple understanding tasks such document classification. For instance, knowing that गुरुत्वाकर्षण means *gravity* and क्वांटम यांत्रिकी means *quantum mechanics* is sufficient to realize a Hindi document containing the above words is about physics, without translating the entire document. Another practical issue is that even though we can apply NLP tools on the translated text, often the output is desired in the target language itself (e.g., dependency parsing, sequence tagging). For these problems, a translation-based approach incurs the additional cost of projecting the output back to the original language, which besides introducing latency, might be a hard task in itself. For instance, how to project back the Part-Of-Speech (POS) tags for tokens in a English phrase that aligns to a single word in Hindi?

All the above challenges become more acute when working with low-resource languages. To extend NLP tools to languages beyond English, we need to find approaches that eliminate or reduce the dependence on high quality annotated data, and that enable one to scale to multiple languages quickly without investing too much effort.

## 1.2. Thesis Statement

In this thesis, I argue that *cross-lingual representations* are an effective means of extending NLP tools to languages beyond English, without resorting to generous amounts of annotated data or expensive machine translation. These cross-lingual representations encode semantics and maintain similarity of lexical items across multiple languages. Such multilingually-enriched semantic representations can be learnt in an inexpensive manner, often from cross-lingual signals completely unrelated to the task of interest. By using these representations as features (in lieu of monolingually derived representations) for training a model for a certain NLP task, we can learn models that operate on input from multiple languages. Using such language-agnostic feature representations in the model enable us to leverage any existing annotations in a high-resource language for joint multilingual training, thereby reducing the amount of supervision required for the task in a new language.

## 1.3. Outline of This Thesis

The rest of the document is organized as follows.

- Chapter 2 gives a brief history of different approaches of representing words in NLP applications, and highlights their limitations in capturing cross-lingual semantics.

- Chapter 3 introduces cross-lingual representations as a means to overcome this limi-

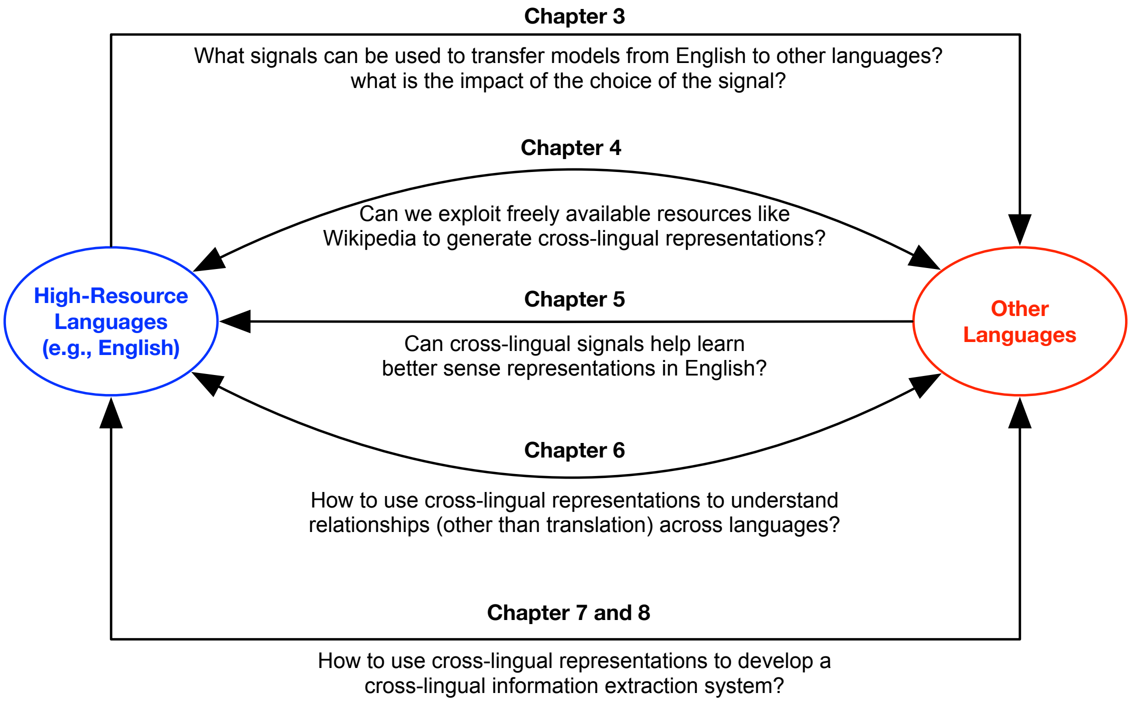

Chapter 3
What signals can be used to transfer models from English to other languages?
what is the impact of the choice of the signal?
Chapter 4
Can we exploit freely available resources like
Wikipedia to generate cross-lingual representations?
High-Resource
Languages
(e.g., English)
Chapter 5
Other
Languages
Can cross-lingual signals help learn
better sense representations in English?
Chapter 6
How to use cross-lingual representations to understand
relationships (other than translation) across languages?
Chapter 7 and 8
How to use cross-lingual representations to develop a
cross-lingual information extraction system?

Figure 1: Overview of the chapters in this thesis. The double-headed arrows indicate tasks involving sharing of information between languages, and single-headed arrows indicate tasks involving transfer from one language to another.

tation, and motivates them as an effective way of translating feature spaces. I then describe different cross-lingual signals that we can exploit to learn cross-lingual representations and evaluate their suitability for achieving semantic and syntactic transfer. Based on (Upadhyay et al., 2016).

- In Chapter 4, I describe another approach to learn cross-lingual representations using multilingual encyclopedic resources such as Wikipedia to encode words in different languages in a shared semantic space, and use it to classify documents without any supervision. Based on (Song et al., 2016).

- In Chapter 5, I describe our work on learning multi-sense representations in English using cross-lingual signals, where we show that using a small amount of multilingual data we can train models for sense induction that compare favorably with a model trained on large amount of monolingual data. Based on (Upadhyay et al., 2017).

- Chapter 6 shows how, by using appropriate cross-lingual word representations, one can detect *asymmetric* relations across languages. The chapter demonstrates that cross-lingual representations can aid in semantic tasks beyond those requiring coarse semantics (such as document classification). Based on (Upadhyay* et al., 2018).

  The next two chapters examine a downstream information extraction task, namely *cross-lingual entity linking*, along two dimensions.

- Chapter 7 describes how we can exploit cross-lingual representations to develop a cross-lingual entity linking system. In particular, I describe how we can augment the limited supervision in a low-resource language with supervision from a high-resource language by using a shared multilingual representation space. Based on (Upadhyay et al., 2018a).

- Chapter 8 examines the problem of name transliteration from a language to English, a crucial component for performing entity linking in the cross-lingual setting. I show that we can improve transliteration from low-resource languages into English using constraints to drive learning with limited supervision. Based on (Upadhyay et al., 2018b).

Finally, Chapter 9 summarizes the contributions of this thesis and provides an overview of future directions.

# CHAPTER 2 : History of Word Representations

## 2.1. Introduction

The generalizability of any machine learning model is a function of the data representation (or features) that it operates on to make predictions. For instance, the generalizability of a document classifier depends on what features are used to represent a document. For most NLP applications, features of smaller units of language such as words are used to derive features for larger linguistic structures such as sentences and documents. For instance, a common choice of word-level features in the document classification task is “how often did word $w$ appear in the document”, and the document-level feature is simply the sum of all the active word-level features in that document. The choice of feature representation for units such as words is thus a crucial decision when developing a NLP model.

The idea of learning mathematical representations useful for building classifiers is a general one (Bengio et al., 2013), and the aim of this chapter is to review different approaches of representing words as mathematical objects in NLP. The techniques and principles introduced in this chapter for training monolingual representations will serve as the starting point for cross-lingual representation learning approaches described in the rest of the thesis.

I start with a discussion of one-hot word representations and their limitations. I will then discuss how the principles of distributional semantics can be put into practice to derive cluster-based or vector-based word representations. Owing to the popularity of vector-based representations, I will briefly discuss the different frameworks for learning vector representations and their benefits. I end the chapter by discussing some known limitations of these word representations, that I will address in the rest of the thesis.

## 2.2. One-Hot Word Representations

Conventional approaches represent each word $w$ in the vocabulary $V$ by denoting $w$ by its index in $V$. For instance, for a word vocabulary $V = \{$*king*, *queen*, *computer*, *virus*, *disease*}, the one-hot representations will be {1, 2, 3, 4, 5}. This representation scheme is better visualized by treating each word as a vector of 0s with a 1 at the index associated with the word (or a *one-hot* vector),

$$v_{king} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, v_{queen} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, v_{computer} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, v_{virus} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, v_{disease} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{2.1}$$

It is easy to see that this is an extremely crude representation that fails to capture relationships (or lack thereof) between words. For instance, related words like *king* and *queen* are treated the same (two *distinct* atomic symbols), and so are unrelated words like *computer* and *king.* Using features built on top of one-hot word representations will also suffer from the same problem. For instance, the parameters learnt for the feature “the word *king* is present in the document” during training cannot be shared for the feature “the word *queen* is present in the document” that is only seen at test time. As a result, these one-hot word representations have limited utility when used as the building blocks for a NLP model. Another drawback is that the representation size increases with the size of the vocabulary, which is cumbersome when working with large corpora.

## 2.3. Distributional Semantics

| | | |
|---|---|---|
| Marco saw a hairy little | *wampimuk* | crouching behind the tree. |
| During the winter the | *wampimuk* | hibernates in his burrow. |
| The girls fed their pet | *wampimuk* | berries and acorns. |

Table 1: The distributional hypothesis in action: We (humans) can guess the meaning of words like *wampimuk* by the context in which it appears.

How can we learn more meaningful representations for words that capture some notion of relatedness? Intuitively, humans infer the meaning of a word from the context in which the word appears. For instance, in Table 1, the meaning of a (hypothetical) word like *wampimuk* in the sentence "Marco saw a hairy little wampimuk crouching behind the tree." can be guessed based on the neighboring words.[1] Such patterns of word usage also allows humans to guess that words like *squirrel* or *rabbit* are similar to *wampimuk*, because they can substitute *wampimuk* in the above context. These observations form the basis of the field of distributional semantics, where word meanings are modeled as a function of statistics computed over a corpus. This intuition can be formalized in two popular distributional semantics hypotheses.

The Distributional Hypothesis

The distributional hypothesis states that *you shall know a word by the company it keeps* (Weaver, 1955; Firth, 1957; Furnas et al., 1983, inter alia). The hypothesis has been restated differently (and more formally) over the years — *words that are similar in meaning occur in similar contexts* (Rubenstein and Goodenough, 1965); *words with similar meanings will occur with similar neighbors if enough text material is available* (Schütze and Pedersen, 1995).

The definition of context and similarity in the distributional hypothesis is task-dependent. The notion of similarity is often dictated by the representation paradigm — vector-based word representations use a geometric similarity metric like cosine or the dot product, while cluster-based word representations simply check for cluster overlap. Most popular representation learning approaches define context to be the *lexical* neighborhood of the word under consideration, but this choice can also be dictated by the task. For instance, in Chapter 6, I will describe word representations that use *syntactic* context of words to learn representations for detecting asymmetric relations like hypernymy.

**Distributed v/s Distributional.** An important distinction is to be made between distributional word representations and *distributed* representations, A related term used in representation learning literature (Bengio et al., 2013). Word representations that appeal to the distributional hypothesis are called *distributional* representations. On the other

[1] Example inspired from Goldberg (2017).

hand, *distributed* representations are those where each word's representation is described by multiple contexts, and each context participates in describing multiple words. Distributed representations are in contrast to *local* representations, such as the one-hot representation discussed earlier. Note that a representation can be both distributional and distributed (e.g., representations learnt using Skip-gram with Negative Sampling (SGNS), described in Section 2.9).

While the distributional hypothesis is widely popular, it is worth noting that not all word representation approaches appeal to it. Another related hypothesis that describes the semantics of a word through statistics computed on a corpus is the *bag-of-words hypothesis.*

**The Bag-of-Words Hypothesis**

The bag-of-words hypothesis states that *similar documents contain similar words, and similar words appear in similar documents.* More formally, *documents that have similar column vectors in a term-document matrix are similar* (Salton et al., 1975; Salton and Buckley, 1988).

The term-document matrix is computed by treating each document as a bag-of-words, hence the hypothesis is referred to as the bag-of-words hypothesis. The bag-of-words hypothesis examines the document level co-occurrence of words, and is close relative of the distributional hypothesis.[2] In Chapter 4, I will describe Explicit Semantic Analysis (ESA) representations, that appeal to the bag-of-words hypothesis to represent words as the 'bag-of-concepts' that they describe in an encyclopedic resource like Wikipedia.

By invoking the distributional hypothesis and the bag-of-words hypothesis, representation learning approaches can extract contextual knowledge directly from raw corpora. This way of extracting semantics automatically is particularly appealing compared to the time-consuming approach of hand-coding semantics (e.g., building lexical ontologies). Another benefit of these representations is their re-usability — once extracted from a large and

[2] In fact, by defining the context of a word to be the documents it appears in, one can arrive at the bag-of-words hypothesis from the distributional hypothesis.

general enough corpora, the same word representation can be used for multiple tasks.

## 2.4. Cluster-based Word Representations

The problem with the one-hot representations were that they were extremely high dimensional (of the size of the vocabulary), and thus did not permit any sharing. An alternative to the crude one-hot representation is to first cluster the words in the vocabulary, such that words that share meaningful linguistic properties gets assigned to the same cluster. Intuitively, this cluster-based word representation reduces the effective vocabulary size to the number of word clusters, such that words that are similar (clustered together) have same one-hot representations. Here, I discuss one popular clustering algorithm for learning such representations, Brown clustering (Brown et al., 1992).

The input to the Brown clustering algorithm is a corpus of words $w_1, w_2, \cdots, w_T$, where $T$ is the size of the corpus. The aim of the algorithm is to assign each word to a cluster, where $\mathcal{C} : V \rightarrow 1, 2, \cdots, K$ is a cluster assignment function that maps each word $w_i$ in the vocabulary $V$ to its cluster $\mathcal{C}(w_i)$. The sequence of cluster assignments $\mathcal{C}(w_1), \mathcal{C}(w_2), \cdots, \mathcal{C}(w_T)$ for words $w_1, w_2, \cdots, w_T$ is modeled using a *class-based bi-gram language model.* Under this model the generative story of the corpus is the following — transition to the current cluster $\mathcal{C}(w_i)$ conditioned on the previous cluster $\mathcal{C}(w_{i-1})$, sample a word $w_i$ conditioned on the current cluster $\mathcal{C}(w_i)$, and repeat the process. The log-likelihood of the corpus $w_1, w_2, \cdots, w_T$ (or the *clustering quality*) under this model is computed as,

$$\log \Pr(w_1, w_2, \cdots, w_T) = \sum_{i=1}^{n} \log \Pr(w_i \mid \mathcal{C}(w_i)) \Pr(\mathcal{C}(w_i) | \mathcal{C}(w_{i-1})) \tag{2.2}$$

The output of the clustering algorithm is a binary tree, whose leaves are words, and internal nodes are clusters of the words in the sub-tree rooted at that node. Each cluster can be expressed using a binary code, which is the representation shared by all the words that belongs to that cluster. It can be shown that the clustering quality is a function of the mutual information of adjacent clusters,

$$Quality(\mathcal{C}) = \sum_{c,c'} \Pr(c, c') \log \frac{\Pr(c, c')}{\Pr(c)\Pr(c')} - \sum_{w} \Pr(w) \log \Pr(w) \tag{2.3}$$

$$= I(\mathcal{C}) - H \tag{2.4}$$

Thus, Brown clustering uses mutual information of adjacent clusters (i.e., at bi-gram level) as a measure of distributional similarity between the words in those clusters. Starting with each word in its own distinct cluster, the brown clustering algorithm greedily merges pair of clusters that cause the smallest decrease in the likelihood of the corpus with respect to the class-based bi-gram language model. The greedy clustering strategy naturally leads to a hierarchical clustering upon termination. However, naively implementing this clustering strategy has a run-time of $O(|V|^5)$, which was reduced to $O(|V|^3)$ in Brown et al. (1992). Later, Liang (2005) proposed a more efficient approach that reduced the run-time to $O(|V|m^2 + n)$ by starting with $m$ initial clusters for each of the $m$ most popular words, where $m$ is number of initial clusters and $n$ is the corpus length.

Brown clustering has proved successful in several semi-supervised scenarios, improving performance on tasks like dependency parsing (Koo et al., 2008; Tratz and Hovy, 2011), syntactic chunking (Turian et al., 2010), with sequence labeling tasks like named-entity recognition enjoying large gains (Freitag, 2004; Miller et al., 2004; Ratinov and Roth, 2009).

**Other Clustering Representations.** Many variants of the Brown clustering algorithm exists. For instance, Ushioda (1996) extended the Brown clustering algorithm to compute clusters for words and phrases, while Martin et al. (1998) extended the bi-gram model to tri-grams. Similarly, Uszkoreit and Brants (2008) used a more expressive class-based language model that transitions to the current cluster $\mathcal{C}(w_i)$ conditioned on the previous word $w_{i-1}$. The word-to-class transitions allow the model more freedom to improve the clustering quality,

$$\log \Pr(w_1, w_2, \cdots, w_T) = \sum_{i=1}^{n} \log \Pr(w_i \mid \mathcal{C}(w_i)) \Pr(\mathcal{C}(w_i)|w_{i-1}) \tag{2.5}$$

More recently, Stratos et al. (2014) noticed that the greedy strategy employed in Brown clustering is simply a heuristic and is not guaranteed to recover the (provably) correct clustering when given sufficient number of training examples. Furthermore, even with Liang (2005)'s improved algorithm, the run-time is prohibitively long. Stratos et al. (2014) proposed an improved cluster based representation learning approach that is guaranteed to recover the correct clustering and runs around 10 times faster than the Liang (2005) implementation.

**Limitations of Cluster-based Representations.** A key limitation of using clustering-based representations is that they are not amenable to end-to-end learning aimed at a downstream task. This is due to the fact that the cluster assignment function $\mathcal{C}$ is, in general, discontinuous, so end-to-end learning methods like back-propagation cannot be used with them. Furthermore, approaches like Brown clustering scale *quadratically* in the number of clusters, and thus are extremely slow to train, preventing more expressive partitioning of the space. The discreteness of the space induced by the clustering also means that the granularity of similarity score between pair of words is discretely quantized. For instance, at a certain level of the Brown clusters, two words either belong to the same cluster (similarity of 1), or don't (similarity of 0). One can circumvent this by describing a word by its path from the root in the binary tree (e.g., *apple* becomes 101101). However, this approach partially alleviates the discreteness of the space. We will see that vector space models allow for a more expressive representation equipped with a continuous similarity measures between words, and are amenable to fast and efficient end-to-end learning.

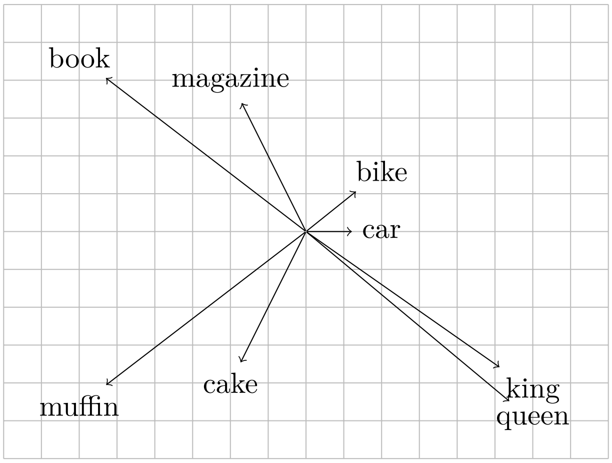


Figure 2: Vector-based word representations in a 2-dimensional space.

## 2.5. Vector-based Word Representations

Vector-based word representations use a point in a $n$-dimensional vector space to describe each word in a language (see Figure 2). Under this representation paradigm, geometric proximity in the vector space is used as surrogate for semantic similarity of two words. For instance, in Figure 2, related words like *king* and *queen* are closer than *king* and *car*. The similarity of two words $v$ and $w$ represented in a vector space as $\mathbf{v}$ and $\mathbf{w}$ respectively, can then be computed using some natural geometric measure of vector proximity, like the dot product $dot(v, w) = \mathbf{v}^\intercal \mathbf{w}$, or the cosine $cosine(v, w) = \frac{dot(v,w)}{|\mathbf{v}||\mathbf{w}|}$. This notion of "proximity as a surrogate for similarity" traces its roots in cognition (Lakoff and Johnson, 1980, 1999) as discussed in (Sahlgren, 2006).

As these representations *embed* a word in a geometric space, they are also called word *embeddings* or word *vectors*. Over the years, several different ways of learning vector-based word representations have been developed, that differ in the form of the vector representation (sparse or dense), and the learning framework used — *count-based* model and *matrix factorization* (either exact or approximate). I briefly discuss these below.

| dimension → | 1st | 2nd | 3rd | 4th | 5th | 6th | 7th | … |
|---|---|---|---|---|---|---|---|---|
| context<br>word | computer | is | of | fruit | breed | physics | bank | … |
| apple | 10 | 70 | 90 | 20 | 0 | 0 | 0 | … |
| cat | 0 | 60 | 75 | 5 | 20 | 0 | 0 | … |
| interest | 25 | 45 | 55 | 5 | 5 | 30 | 30 | … |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋱ |

Table 2: Example of a word-context co-occurrence matrix extracted from a corpus. Each dimension of a row corresponds to co-occurrence with a certain context word. For instance, the word apple co-occurs with fruit (4th dimension) 20 times in the corpus.

## 2.6. Sparse Word Representations using Count Models

Sparse word representations (Lin, 1998a,b; Turney and Pantel, 2010; Baroni and Lenci, 2010; Levy and Goldberg, 2014a, inter alia) describe each word in the vocabulary using a high dimensional but extremely *sparse* (>90% entries of the vector are 0s) vector.

The key ingredient in all sparse word representations is a *co-occurrence matrix* $\mathbf{M} \in R^{|V|\times|C|}$, such as the one shown in Table 2, where $|V|$ is the size of the word vocabulary $V$, $|C|$ is the size of the context vocabulary $C$. The entry $\mathbf{M}[w, c]$ counts how often the word $w \in V$ has appeared in a context $c \in C$.[3] As a result, these approaches are also called *count-based models* for learning word representations (Baroni et al., 2014). The matrix $\mathbf{M}$ is extremely sparse, e.g., for the text8 corpus containing about 17 million tokens and 71 thousand distinct words, the co-occurrence matrix has ≈1% non-zero entries only (Jiang et al., 2018). The sparsity of $\mathbf{M}$ is an artifact of the *Zipfian distribution* of words in natural language (Zipf, 1949). The representation of a word $w$ can be directly read off the from the relevant row $\mathbf{w} \in \mathbb{R}^{|C|}$ in the co-occurrence matrix, where $|C|$ is the size of the context vocabulary.

Variants of count-based models differ along two dimensions — the definition of context, and how the entries of $\mathbf{M}$ are computed. As discussed in Section 2.3, the definition of the

[3] While I yield to the popular usage, the entries of the matrix can denote event frequencies other than simple co-occurrence, such as, "how often did $w$ take $c$ as its parent in a syntactic tree?".

context is task-dependent, with lexical neighborhood being a popular choice. Similarly, there are several ways to compute the entries of the co-occurrence matrix $\mathbf{M}$. A naive way of computing the entries of the co-occurrence matrix is to simply set each to the raw co-occurrence frequency $n(w, c)$. However, raw frequency is not a good measure of relatedness between words, because it is skewed towards co-occurrences with function words like *the*, *of*, *is* etc., that do not truly reflect the semantics of the context. Different approaches of computing the entries of $\mathbf{M}$ have been proposed to remedy this issue.

**Point-wise Mutual Information (PMI).** A popular approach is to weigh each entry of the co-occurrence matrix using point-wise mutual information or PMI (Church and Hanks, 1990). PMI measures the association between word w and context c as,

$$PMI(w, c) = \log \frac{\Pr(w, c)}{\Pr(w) \Pr(c)} \tag{2.6}$$

The probability of the co-occurrence is simply estimated using counts $\Pr(w, c) = \frac{n(w,c)}{N}$, $\Pr(w) = \frac{n(w)}{N}$, $\Pr(c) = \frac{n(c)}{N}$, where $n(w, c)$ is the frequency of $(w, c)$ co-occurrence, $n(w)$ and $n(c)$ are the frequencies of $w$ and $c$ respectively, and $N$ is the total corpus size. Intuitively, PMI measures how much the probability $\Pr(w, c)$ differs from what we would expect it to be assuming independence of $\Pr(w)$ and $\Pr(c)$ (Bouma, 2009).

**Positive PMI (PPMI).** An issue with PMI is that negative PMI values do not convey meaningful information. When will PMI assume a negative value? when we have $\Pr(w, c) < \Pr(w) \Pr(c)$, that is when one of $w$ or $c$ tend to occur individually rather than together. For instance, this can happen when computing co-occurrences with words like *the*. PPMI helps to ignore such uninformative co-occurrences. Furthermore, for unobserved $(w, c)$ pairs $PMI(w, c) = log\, 0 = -\infty$, which makes the co-occurrence matrix dense (and ill-defined). To remedy these issues, Levy and Bullinaria (2001) suggested using positive PMI, that simply ignores negative values $PPMI(w, c) = \max[PMI(w, c), 0]$.

**Positive Local Mutual Information (PLMI).** PMI and PPMI have a bias towards rare events (i.e., small $n(w, c)$). To see this, notice that for perfectly correlated word $w$ and context $c$ (i.e., $\Pr(w) = \Pr(c) = \Pr(w, c)$), the PMI or PPMI value is $-\log \Pr(w, c)$, which is higher for low $\Pr(w, c)$ (i.e., rare co-occurrences). Positive Local Mutual Information (PLMI) (Evert, 2005, 2008) balances PPMI by including a factor for the co-occurrence count $n(w, c)$ of word $w$ and context $c$, $PLMI(w, c) = n(w, c) \times PPMI(w, c)$. This way rare co-occurrences get appropriately low entries in $\mathbf{M}$.

**Limitations of Sparse Representations.** While sparse word representations do overcome most of the limitations of one-hot word presentations, they have their own limitations. The entries of $\mathbf{M}$ are frequency estimates from a large corpus, that may be unreliable due to chance co-occurrences of unrelated word pairs and missing evidence for unobserved but related word pairs. Furthermore, many columns of the matrix $\mathbf{M}$ will be highly correlated. For instance, words that appear frequently with the context word *big* are also likely to appear with the context word *large*. These correlated context dimensions introduce redundancy in the representation. The size of the representations also increases with the size of the context vocabulary, just like one-hot representations.

Dense low dimensional word representations alleviate the above limitations of the sparse word representations. These representations are *dense*, in that most entries in the representation are non-zero values. Unlike sparse representation, the representation size is significantly smaller than the size of the vocabulary (e.g., dense representation of size 100 can be used for a vocabulary of size 200k words). The reduced representation size also reduces the redundancy due to correlated dimensions in the representation. Over the years, several different approaches have been used to learn dense word representations. While some approaches derive the dense representations by factorizing the sparse co-occurrence matrix (either *exactly* or *approximately*), others directly learn dense representation by using them for a prediction task like *neural language modeling*.

## 2.7. Dense Word Representations using Exact Matrix Factorization

Exact factorization approaches first factorize the sparse high dimensional word co-occurrence matrix *exactly* using a *dimensionality reduction* technique like Singular Value Decomposition (SVD) (Golub and Kahan, 1965) or Principal Component Analysis (PCA) (Pearson, 1901). A low rank approximation of the matrix is then constructed from the resulting factors, from which the dense representations are derived.

A popular example of this approach is Latent Semantic Analysis (LSA) (Deerwester et al., 1990; Landauer and Dumais, 1997).[4] LSA factorizes the co-occurrence matrix $\mathbf{M} \in \mathbb{R}^{|V|\times|C|}$ into factors $\mathbf{U}, \Sigma, \mathbf{V}$ using SVD,

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\intercal} \tag{2.7}$$

where $\mathbf{\Sigma} \in \mathcal{R}^{n\times n}$ is a diagonal matrix containing the singular values $(\sigma_1, \cdots, \sigma_n)$, and $\mathbf{U} \in \mathbb{R}^{|V|\times|n|}$ and $\mathbf{V} \in \mathbb{R}^{|C|\times|n|}$ are orthogonal matrices. Note that the factorization is *exact*, as $\mathbf{U}$, $\mathbf{\Sigma}$, $\mathbf{V}$ can exactly recover $\mathbf{M}$. It can be shown that the best rank $d$ approximation $\mathbf{M}_d$ to $\mathbf{M}$ under the squared difference norm (spectral norm), can be computed using the output of SVD, $\mathbf{M}_d = \sum_{i=1}^{d} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\intercal}$ where $\mathbf{u}_i$ and $\mathbf{v}_i$ are $i^{th}$ column of $\mathbf{U}$ and $\mathbf{V}$, respectively (Eckart and Young, 1936). The dense word representation are obtained by projecting the rank-$d$ approximation of the co-occurrence matrix using $\mathbf{V}_d$,

$$\mathbf{M}_d\mathbf{V}_d = \mathbf{U}_d\mathbf{\Sigma}_d \tag{2.8}$$

where $\mathbf{M}_d\mathbf{V}_d \in \mathbb{R}^{|V|\times d}$ is the matrix of the word representations. Over the years, other matrix factorization approaches have improved over LSA in different ways, like making LSA translation-invariant (Gardner et al., 2016) or scale-invariant (Dhillon et al., 2012, 2015), or using information theoretic measures for discrete distributions like Hellinger PCA (Lebret and Collobert, 2014).

[4] Also referred to Latent Semantic Indexing (LSI).

The motivation behind using exact matrix factorization approach is to reduce both the dimensionality and the sparsity of the observations simultaneously (Sahlgren, 2006). The low rank approximation merges the dimensions associated with contexts that are "similar", thereby compressing information spread across multiple related dimensions in the sparse representation. An added benefit of this low rank approximation is that it allows the user to control the size of the representation unlike sparse representations where the size scales with the size of vocabulary. Another attractive property of exact matrix factorization approaches like SVD is that they do not require hyper-parameter tuning (e.g., learning rates) and can be solved efficiently.

## 2.8. Dense Word Representations using Neural Language Modeling

The Brown clustering approach trained word representations by using a class-based bi-gram language model to assign discrete clusters to the words in the vocabulary. However, Bengio et al. (2003) showed that useful word representations can also be learnt as a by-product of using a neural network for performing the language modeling task, i.e., predict the next word $w_t$ given a sequence of preceding words $w_1, \cdots, w_{t-1}$.

In a neural language modeling approach the sequence of left context words $w_1, \cdots, w_{t-1}$ is first transformed into a sequence of vectors $\mathbf{w}_1, \cdots, \mathbf{w}_{t-1}$ using an *embedding matrix* $\mathbf{W} \in \mathbb{R}^{|V| \times d}$. Using this sequence of vectors, a neural network $g$, computes the score for the word $w$ being the next word. This score is then normalized by using the *softmax* operation to obtain the next word probability distribution $\Pr(w_t = w \mid w_1, \cdots, w_{t-1})$,

$$\Pr(w_t = w \mid w_1, \cdots, w_{t-1}) = \frac{\exp g(w, \mathbf{w}_1, \cdots, \mathbf{w}_{t-1})}{\sum_{w'} \exp g(w', \mathbf{w}_1, \cdots, \mathbf{w}_{t-1})} \tag{2.9}$$

the parameters of the neural network $g$ and the embedding matrix $\mathbf{W}$ are parameters that are learnt by maximizing the log-likelihood of the corpus $\frac{1}{T} \sum_{i=1} \Pr(w_i \mid w_1..w_{i-1})$ using gradient descent (Cauchy, 1847) and back-propagation (Rumelhart et al., 1986). After training, the rows of the $\mathbf{W}$ are the dense word representations.

Early approaches like that of Bengio et al. (2003) used a fixed number of words from the left context to compute the of the next word probability using a feed-forward layer. This was improved upon by Collobert and Weston (2008); Collobert et al. (2011), who used a convolutional operation instead to combine the context word representations, minimized using a max margin loss instead of cross-entropy loss. Later, Mikolov et al. (2010) showed further improvements using a *recurrent neural network* (Elman, 1990) that in principle can take into account an arbitrary long past context of the word, unlike the bi-gram assumption made by Brown clustering.

## 2.9. Dense Word Representations using Approximate Matrix Factorization

So far, we have seen two approaches to learn dense word representations, either by computing co-occurrence matrix and factorizing it *exactly* using low-rank matrices, or training a model to predict the next word. These approaches naturally lead to the following question — *what if we train a model to predict entries of the word-context co-occurrence matrix, instead of just the next word?* Such an approach will approximately factorize the word-context co-occurrence matrix, but can use more expressive ways to model the entries of the co-occurrence matrix.

### Skip-gram with Negative Sampling (SGNS)

The language modeling objective only utilizes the past context (i.e., context to the left) to predict the next word. Mikolov et al. (2013a) proposed the *skip-gram model*, an alternative training strategy to learn word representations that utilizes both the right and left context around a word.

The skip-gram model's learning objective aims to train a classifier for a prediction task, and word representations are learnt as a by-product of this training objective. The prediction task aims to predict the surrounding words in a window around a central (or *pivot*) word, using the representations of the pivot and surrounding words as parameters of the classifier.

Formally, the learning objective of the skip-gram approach is,

$$\mathcal{L} = -\sum_{w} \sum_{w_c \in \textsc{Neighbors}(w)} \log \Pr(w_c \mid w) \tag{2.10}$$

The skip-gram formulation attempts to learn representations that are good at predicting neighboring words (in a predefined context window size), where the prediction are modeled using a log-linear conditional probability,

$$\Pr(w_c \mid w) = \frac{\exp(\mathbf{w}_c{}^{\intercal}\mathbf{w})}{\sum_{w' \in V} \exp(\mathbf{w}'^{\intercal}\mathbf{w})} \tag{2.11}$$

The denominator of the conditional probability requires sum over $|V|$ terms, computing which can become prohibitive for vocabularies derived from large corpora. To avoid this computational bottleneck, efficient implementations of the skip-gram model employ *negative sampling* to optimize an approximation of the skip-gram objective. The *skip-gram with negative sampling (SGNS)* formulation poses the following binary classification task — is a given $(w, c)$ pair a co-occurrence observed in the corpus $D$, or obtained by randomly pairing a word $w$ with a context $c$? The randomly paired are referred as *negative samples* from a noise distribution $D'$, leading the following learning objective,

$$\max \sum_{(w,c) \in D} \log \Pr(D = 1 \mid (w, c)) + \sum_{(w,c) \in D'} \log \Pr(D = 0 \mid (w, c)) \tag{2.12}$$

$$\text{where } \Pr(D = 1 \mid (w, c)) = \frac{1}{1 + \exp(-\mathbf{w}^{\intercal}\mathbf{c})} = 1 - \Pr(D = 0 \mid (w, c)) \tag{2.13}$$

is the probability that the model assigns to the $(w, c)$ pair originating from the corpus. Note that the SGNS objective does not involve computing a normalization term over the entire vocabulary, and thus can be efficiently optimized. The negative sampling approach can be shown to be a special case of the more general technique of noise contrastive estimation (Gutmann and Hyvärinen, 2012; Dyer, 2014), used for training un-normalized probabilistic models. It was shown by Levy and Goldberg (2014c) that the SGNS learning objective implicitly factorizes a shifted version of the PPMI matrix.

**Related Word Prediction Tasks.** Instead of predicting the next word using the left context words (as in language modeling), or predicting the context words using the pivot word (as in SGNS), other word prediction tasks can also be used to learn word representations. For instance, Even-Zohar et al. (1999); Even-Zohar and Roth (2000) train word representations by predicting a missing word using the representations of the other words in a sentence. The representations are learnt as a set of discriminators in the Sparse Network of Winnows (SNoW) architecture (Roth, 1998), with the aim of improving context-sensitive spelling correction. A similar learning objective to the above was proposed by Mikolov et al. (2013a), which they termed contextual bag-of-words (CBOW). In contrast to SGNS, the aim of CBOW is to train a classifier for the opposite prediction task — predict the pivot word using all the surrounding words.

$$\mathcal{L} = -\sum_{w} \log \Pr(w \mid \textsc{Neighbors}(w)) \tag{2.14}$$

$$\text{where } \Pr(w \mid \textsc{Neighbors}(w)) = \frac{\exp(\mathbf{w}^{\intercal}\mathbf{v})}{\sum_{w' \in V} \exp(\mathbf{w'}^{\intercal}\mathbf{v})} \text{ and } v_{\textsc{Neighbors}(w)} = \sum_{w_c \in \textsc{Neighbors}(w)} \mathbf{w_c} \tag{2.15}$$

The CBOW model is several times faster than the SGNS model. On the other hand, SGNS works well with even a small amount of training data, because each context window generates multiple training instances, unlike CBOW, where each context window generates a single training instance.

### Global Vector Representations (GloVe)

Unlike SGNS, global vector representations (GloVe) (Pennington et al., 2014) *explicitly* factorizes the log of co-occurrence matrix. GloVe approximates the log-co-occurrence count $\log n(w, c)$ using the dot product of vector representations $\mathbf{w}$ and $\mathbf{c}$ of the word and context, weighted using a function $f$ of the co-occurrence frequency $f(n(w, c))$. Formally, the loss for the entry corresponding to word $w_i$ and context $c_j$ in the co-occurrence matrix is,

$$\mathcal{L}(w_i, c_j) = f(n(w_i, c_j)) \times \left(\mathbf{w}_i{}^\intercal \mathbf{c}_j + b_{w_i} + b_{c_j} - \log n(w_i, c_j)\right)^2 \tag{2.16}$$

$$\text{where} \qquad f(x) = \begin{cases} \left(\frac{x}{x_{max}}\right)^\alpha & \text{if } x < x_{max} \\ 0 & \text{otherwise} \end{cases} \tag{2.17}$$

The choice of $f$ ensures that unobserved entries get a weight of 0, and neither rare co-occurrences nor frequent co-occurrences are over-weighted.[5]

The GloVe objective is closely related to PMI matrix factorization. Indeed, fixing $b_{w_i} = \log n(w_i)$ and $b_{c_j} = \log n(c_j)$ results in an objective that is approximately factorizing the PMI-weighted co-occurrence matrix shifted by $\log N$, where $N$ is the corpus size. However, unlike exact factorization approaches like LSA, GloVe weighs each entry in the matrix differently (using $f(n(w_i, c_j))$), and learns $b_{w_i}$ and $b_{c_j}$, make it more expressive.

## 2.10. Limitations of Word Representations

Despite their overwhelming success as features for NLP tasks, word representations have their shortcomings. I discuss a few aspects in which these representations fall short, and how the community has attempted to overcome them.

**Issues of Distributional Hypothesis.** Distributional approaches do not account for the compositional nature of language, and cannot represent larger units of language like phrases or sentences. Attempts have been made to remedy this by appealing to formal semantics, and combining it with distributional representations (Grefenstette and Sadrzadeh, 2011; Garrette et al., 2011; Grefenstette, 2013; Lewis and Steedman, 2013; Beltagy et al., 2016).

Problems with distributional similarity are also exposed in lexical semantics tasks. For instance, distributional methods assign high similarity to word pairs like *new* and *old*, *happy* and *sad*, and are thus unable to distinguish antonyms from synonyms (Grefenstette,

[5] Pennington et al. (2014) report $\alpha = 3/4$ gave best results.

1992; Padó and Lapata, 2003). This is an artifact of the distributional hypothesis itself, as antonymous word pairs tend to occur in similar contexts even though they convey opposite meanings in those contexts.

A related issue is that measures like cosine similarity are symmetric, and are thus unable to reveal any asymmetric relationships that might hold between words. For instance, the cosine between *rodent* and *squirrel* is the same regardless of the order in which its is computed, even though there is an asymmetric relation between the two — *squirrel* is a kind of *rodent.* In Chapter 6, I will discuss an asymmetric similarity measure that can remedy this issue.

**Single Representation per Word.** The approaches discussed so far associate each word with a single representation. As a result, representations for polysemous words like *screen* are forced to capture multiple meanings by superimposing all relevant meaning representations (Arora et al., 2018). Accurately capturing the semantics of such ambiguous words is important, as many frequent words have multiple senses.[6] Several approaches to remedy this limitation by learning a fixed number ($> 1$) of representations per word have been proposed that use monolingual distributional information (Reisinger and Mooney, 2010; Huang et al., 2012). In Chapter 5, I will describe one such approach that uses multilingual translational information in addition to monolingual distributional information to *dynamically* learn different number of representations per word.

**Interpretability.** A limitation of using dense vector representations is that the low-rank approximation process obscures the meaning of each dimension of the vector. For instance, in Table 2 the 5th dimension of the word cat denotes how often it co-occurs with breed, but after (say) factorizing the co-occurrence matrix, we cannot assign any meaning to the $5^{th}$ dimension of a GloVe vector. This hurts interpretability — while each dimension in sparse representations correspond to presence (or absence) of a particular context word, this is not so for the dense representations. To retain the interpretability of the sparse vector

[6] Principle of Economical Versatility of Words (Zipf, 1949).

spaces, without incurring the limitations of naive sparse word representations, different sparse coding approaches have been proposed (Murphy et al., 2012; Faruqui et al., 2015b; Subramanian et al., 2018), that transform dense $d$-dimensional vector representations to sparse $d$-dimensional ones. In Chapter 6, we will examine one such sparse coding objective in the context of learning dependency based cross-lingual representations.

**Cross-lingual Semantics.** While word representations can be trained monolingually for each language using a large corpus, these representations are unable to capture meaningful relationships that exist between words *across* languages. For instance, monolingually trained word representations in Hindi and English will not reflect the fact that दूध in Hindi and *milk* in English are translationally equivalent. The reason behind this is that both representations have been trained *solely* using monolingual distributional information, and have no cross-lingual signal to align the vector spaces that are learnt *independently.* As a result, any NLP model that uses word representations as features for training is also limited to operate on a *single* language. This is undesirable, because this forces one to develop separate models for different languages, and prevents utilizing task-specific supervision available in one language to aid in another language. Ideally, one would expect that a model trained using supervision available in one language should work reasonably well on related language, something that a human learner can effectively adapt to do. In Chapter 3, I will describe how we can overcome this limitation by using different forms of cross-lingual alignments to align the semantic spaces across languages. Using such cross-lingual word representations for performing lexical semantic tasks, facilitating model transfer, and sharing of supervision across languages is the main theme of this thesis.

# CHAPTER 3 : Cross-lingual Word Representations

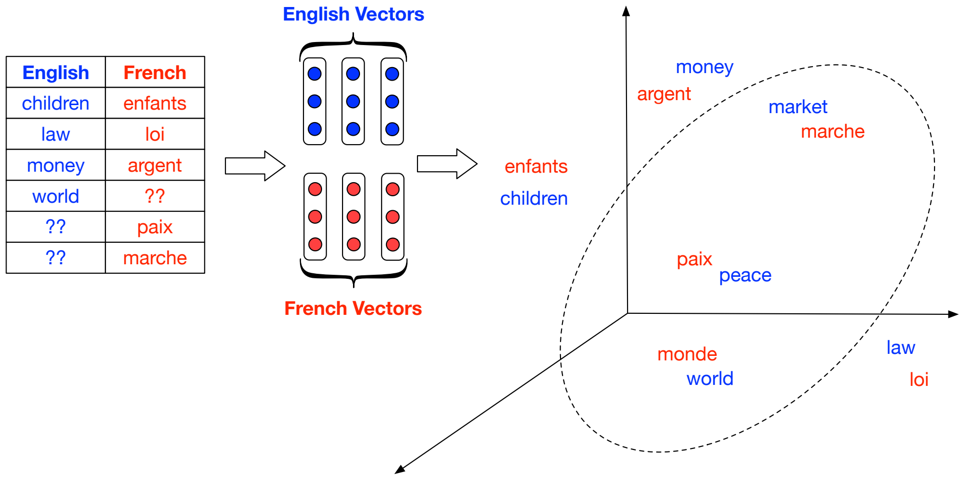
## 3.1. Introduction

One of the limitations of traditional word representations discussed in the last chapter is that they are confined to a single language. As a result, NLP models that employ these representations as features for training cannot be transferred to a related language, despite the underlying task remaining the same. In this chapter, I discuss how word representations can be learnt across languages to capture cross-lingual relationships between words in addition to the notion of monolingual distributional similarity, and facilitate model transfer.

I will discuss the various cross-lingual alignments that are available for learning cross-lingual representations, and evaluate their ability to extrinsically facilitate model transfer across languages for semantic and syntactic applications respectively. The discussion will also focus on the cost of obtaining the cross-lingual alignment and the suitability of different signals for different tasks.

## 3.2. Motivation

*Why do we expect cross-lingual word representations to aid in model transfer?* Suppose we have a supervised document classification model in English, and we wish to use it for document classification in French. One of the features used in the classification model in English is "the word *law* is present in the document". To use the English model, we need to translate this feature to its French equivalent "the word *loi* is present in the document". One way to translate the feature representation is to use a discrete bilingual dictionary (such as one shown in Figure 3), mapping a word in English to its translational equivalent word in French. In practice, any such dictionary will have missing entries (missing dictionary entries are indicated by *??*). Consequently, using such a discrete bilingual dictionary to translate features will have coverage problems, which can lead to loss of useful features. For instance, the dictionary in Figure 3 cannot translate the feature "the word *world* is present



| English | French |
|---|---|
| children | enfants |
| law | loi |
| money | argent |
| world | ?? |
| ?? | paix |
| ?? | marche |

Figure 3: Cross-lingual word representations as a vector space approximation of discrete translation dictionaries. By encoding items present in the dictionary as points in a continuous vector space, one can recover some of the missing entries. Note that while the figure uses vector-based representations (which partition the space implicitly) a similar reasoning can be applied to cluster-based representations (which partition the space explicitly).

in the document".

On the other hand, if the words in both languages are embedded in a shared representation space using the bilingual dictionary (we will see how in Section 3.4.3), such that word pairs that are translational equivalent are close, it is possible to recover from this sparsity of the discrete dictionary. Intuitively, if two words in English are geometrically close in the vector space, and one of them is missing its translation, the other word's nearest neighbor in the other language can serve as the missing translation. Indeed, the ability to recover such missing entries is the basis of applications like bilingual dictionary induction (Rapp, 1995, 1999; Vulić and Moens, 2015; Irvine and Callison-Burch, 2017, inter alia), a task we will examine in Section 3.5.2.

In addition to capturing cross-lingual relationships, these embeddings can also aid in learning better representations for monolingual lexical semantics. This is possible because two or more words that align to similar words in another language should be semantically similar. Indeed, this intuition is the basis of paraphrase detection (Bannard and Callison-Burch, 2005; Callison-Burch, 2007). We will see cross-lingual signals aiding one such task in Chapter 5.

**Notation.** Let $W = \{w_1, w_2, \ldots, w_{|W|}\}$ be the vocabulary of a language $l_1$ with $|W|$ words, and $\mathbf{W} \in \mathbb{R}^{|W| \times l}$ be the corresponding embedding matrix denoting word embeddings of length $l$, such that row $i$ of the matrix $\mathbf{W}$ corresponds to the vector of the word $w_i$. Similarly, let $V = \{v_1, v_2, \ldots, v_{|V|}\}$ be the vocabulary of another language $l_2$ with $|V|$ words, and $\mathbf{V} \in \mathbb{R}^{|V| \times m}$ be the corresponding embedding matrix denoting word embeddings of length $m$. We denote the word vector for a word $w$ by $\mathbf{w}$.

### 3.3. Cluster-based Cross-lingual Representations

In cross-lingual cluster based representations, each cluster contains words in two (or more) languages that share some meaningful linguistic properties. A general approach for learning such representations was proposed by Täckström et al. (2012).

Täckström et al. (2012)'s approach involves two-stages, where words in the source language $S$ are first clustered monolingually, and these clusters are then projected to the target language $T$ using word-alignments from parallel corpora. The monolingual clustering can be performed by using any of the clustering algorithms discussed in Section 2.4. Each target word is assigned to the cluster with which it is most often aligned according to the word alignments. Formally, a target word $v_i$ is assigned to the cluster $\mathcal{C}(v_i)$ which solves,

$$\mathcal{C}(v_i) = \underset{k \in [1, \cdots, K]}{\arg\max} \sum_{(v_i, w_j) \in \mathbf{Q}} n_{i,j} \times \mathbb{1}[\mathcal{C}(w_j) = k] \tag{3.1}$$

where $\mathbf{Q}$ is the set of word alignments, $n_{i,j}$ is the number of times word $v_i$ is aligned to word $w_j$, and $\mathbb{1}[p]$ is an indicator variable that is 1 when condition $p$ is true. Täckström et al. (2012) showed that such cross-lingual word clusters can be included in lieu of brittle, language-specific lexical features for delexicalized transfer models.

A obvious limitation of the two-stage approach is that words outside the alignment dictionary will not get mapped to any cluster. To fix this, Täckström et al. (2012) also suggested a joint clustering approach that alternatingly optimizing the monolingual clustering objective in each language, followed by performing the projection step in Equation 3.1.

## 3.4. Vector-based Cross-lingual Representations

The cross-lingual clustering approach described above suffer from the same limitations of cluster-based representations discussed in Section 2.4. Furthermore, optimization-wise the approach of fixing the clusters alternatingly is not satisfactory; ideally one would like a joint learning objective where the updates respect both monolingual and cross-lingual signals *simultaneously.* The vector-based cross-lingual word representations described in this section do not suffer from these issues.

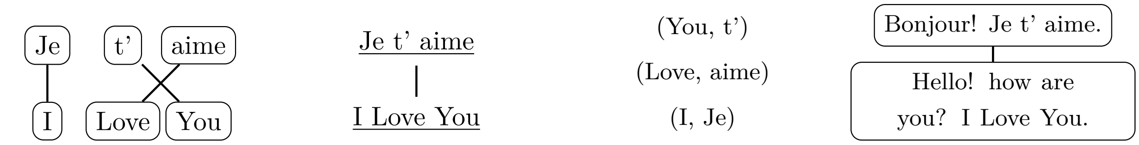
I will describe four different cross-lingual representation learning approaches, and show how each is an instance of a general learning objective (Algorithm 1) to facilitate better understanding. The representation learning approaches use different forms of cross-lingual alignments (illustrated in Figure 4) as supervision for aligning the vector spaces. The alignments differ in the nature of cross-lingual information they encode, and in terms of the cost of obtaining them. I describe them in the following sections.

**Algorithm 1** A general algorithm for inducing cross-lingual word embeddings.

```
Input:
    Initial embedding matrices $\mathbf{W}^0, \mathbf{V}^0$,
    Suitably defined monolingual losses $A$ and $B$, and cross-lingual loss $C$,
    Scalar weights $\alpha$ and $\beta$.
Output: $\mathbf{W}^*$,$\mathbf{V}^*$
Initialize $\mathbf{W} \leftarrow \mathbf{W}^0$,$\mathbf{V} \leftarrow \mathbf{V}^0$
$(\mathbf{W}^*, \mathbf{V}^*) \leftarrow \arg\min \alpha A(\mathbf{W}) + \beta B(\mathbf{V}) + C(\mathbf{W}, \mathbf{V})$
return $(\mathbf{W}^*, \mathbf{V}^*)$
```

### *3.4.1. Sentence and Word-level Alignment*

Parallel corpora are a valuable source of translational information that are available for many high-resource languages. A parallel corpus contains aligned sentences in two (or more) languages, where aligned sentences are translations of each other. Word alignments



(a) Sentence and Word-level Alignment

(b) Sentence-level Alignment

(c) Word-level Alignment

(d) Document-level Alignment

Figure 4: Different forms of cross-lingual alignments one can use as supervision for learning cross-lingual representations. From left to right, the cost of the supervision required varies from expensive (sentence and word-level alignments) to cheap (document-level alignment).

can be automatically induced from such aligned sentences in a parallel corpora using a statistical aligner (e.g., IBM Model 1 aligner (Brown et al., 1993), the `cdec` aligner (Dyer et al., 2013)). These sentence and word-level alignments (Figure 4a) derived from a parallel corpus are used to learn cross-lingual representations. There are many approaches that use these alignments to learn cross-lingual representations (Klementiev et al., 2012; Zou et al., 2013; Kočiský et al., 2014). This section describes one such approach, that of Luong et al. (2015b), henceforth referred as BiSkip.

The learning objective of BiSkip is an extension of the skip-gram model (Mikolov et al., 2013a), where the context of a word is expanded to include bilingual links obtained from word alignments, so that the model is trained to predict words cross-lingually. In particular, given a word alignment $(v, w) \in \mathbf{Q}$ from word $v \in V$ in language $l_2$ to $w \in W$ in language $l_1$, BiSkip predicts the context words of $w$ using $v$ and vice-versa. Formally, we can define a cross-lingual loss term $D_{12}$ for predicting the neighbors of word $w$ in language $l_1$ using the embedding for $v$ in language $l_2$ as,

$$D_{12}(\mathbf{W}, \mathbf{V}) = - \sum_{(v,w) \in \mathbf{Q}} \sum_{w_c \in \textsc{Neighbors}_1(w)} \log \Pr(w_c \mid v) \tag{3.2}$$

where $\textsc{Neighbors}_1(w)$ is the context of $w$ in language $l_1$, $\mathbf{Q}$ is the set of word alignments, and $\Pr(w_c \mid v) \propto \exp(\mathbf{w}_c{}^\intercal \mathbf{v})$. A similar cross-lingual loss term $D_{21}$ can be defined for predicting neighbors of $v$ in language $l_2$ using embedding for $w$ in $l_1$. In addition to the cross-lingual terms $D_{12}$ and $D_{21}$, the BiSkip learning objective contains monolingual terms for predicting the monolingual context of $v$ and $w$ in the respective languages,

$$D_{11}(\mathbf{W}) = -\sum_{w \in W} \sum_{w_c \in \textsc{Neighbors}_1(w)} \log \Pr(w_c \mid w) \tag{3.3}$$

$$D_{22}(\mathbf{V}) = -\sum_{v \in V} \sum_{v_c \in \textsc{Neighbors}_2(v)} \log \Pr(v_c \mid v) \tag{3.4}$$

The BiSkip objective can be cast into Algorithm 1 as,

$$A(\mathbf{W}) = D_{11}(\mathbf{W}) \qquad B(\mathbf{V}) = D_{22}(\mathbf{V}) \tag{3.5}$$

$$C(\mathbf{W}, \mathbf{V}) = D_{12}(\mathbf{W}, \mathbf{V}) + D_{21}(\mathbf{W}, \mathbf{V}) \tag{3.6}$$

where $A(\mathbf{W})$ and $B(\mathbf{V})$ are the familiar SGNS formulation of the monolingual part of the objective from Section 2.9.

### *3.4.2. Sentence-level Alignment*

Sometimes a phrase in a language aligns with a word in another language, a fact that word-level alignments cannot capture. To overcome this limitation, Hermann and Blunsom (2014a) developed the Bilingual Compositional Vector Model (BiCVM) algorithm, that learns cross-lingual word representations using only sentence-level alignment (Figure 4b). Hermann and Blunsom (2014a) argued that allowing the model to learn similarity at a coarser level (namely, sentence-level) is better in such cases, than forcing it to respect word alignments. BiCVM assumes that aligned sentences have equivalent meaning, thus their sentence representations should be similar.

Let $\vec{v} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i, \ldots]$ and $\vec{w} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_j, \ldots]$ denote two aligned sentences, where $\boldsymbol{x}_i \in \mathbf{V}, \boldsymbol{y}_j \in \mathbf{W}$, are vectors corresponding to the words in the sentences. Let functions $f : \vec{v} \rightarrow \mathbb{R}^n$ and $g : \vec{w} \rightarrow \mathbb{R}^n$, map sentences to their semantic representations in $\mathbb{R}^n$. BiCVM generates word vectors by minimizing the squared $\ell_2$ norm between the sentence representations of aligned sentences. To prevent the degeneracy arising from directly minimizing the $\ell_2$ norm, BiCVM uses a noise-contrastive large-margin update, with randomly drawn sentence pairs $(\vec{v}, \vec{w}^n)$ as negative samples. The margin-based loss (with margin $\delta$) for the sentence pairs $(\vec{v}, \vec{w})$ and $(\vec{v}, \vec{w}^n)$ can be written as,

$$E(\vec{v}, \vec{w}, \vec{w}^n) = \max\left(\delta + \Delta E(\vec{v}, \vec{w}, \vec{w}^n), 0\right) \tag{3.7}$$

$$\text{where } \Delta E(\vec{v}, \vec{w}, \vec{w}^n) = E(\vec{v}, \vec{w}) - E(\vec{v}, \vec{w}^n) \tag{3.8}$$

$$\text{and } E(\vec{v}, \vec{w}) = \|f(\vec{v}) - g(\vec{w})\|^2 \tag{3.9}$$

This can be cast into Algorithm 1 by,

$$C(\mathbf{W}, \mathbf{V}) = \sum_{\substack{\text{aligned } (\vec{v},\vec{w}) \\ \text{random } \vec{w}^n}} E(\vec{v}, \vec{w}, \vec{w}^n) \tag{3.10}$$

$$A(\mathbf{W}) = \|\mathbf{W}\|^2 \qquad B(\mathbf{V}) = \|\mathbf{V}\|^2 \tag{3.11}$$

with $A(\mathbf{W})$ and $B(\mathbf{V})$ being regularizers, with $\alpha = \beta$.

### *3.4.3. Word-level Alignment*

The previous two approaches relied on parallel corpora in the languages of interest for deriving the cross-lingual alignment, which is an expensive resource not available for many languages (Lopez and Post, 2013). In contrast, bilingual translation dictionaries (Figure 4c) containing a few thousand words are much easier to obtain (or annotate) than a parallel corpus. These dictionaries can serve as another form of cross-lingual alignment, to ensure that the representations for translationally equivalent word pairs (i.e., entries in the dictionary) are similar. This principle is the basis of many representation learning algorithms that use bilingual dictionaries (Mikolov et al., 2013b; Lu et al., 2015; Smith et al., 2017; Artetxe et al., 2017); this section describes the approach proposed in Faruqui and Dyer (2014), henceforth referred as BiCCA (Bilingual Canonical Correlation Analysis based embeddings).

The BiCCA model projects independently trained monolingual embedding matrices $\mathbf{W}, \mathbf{V}$ using Canonical Correlation Analysis (CCA) (Hotelling, 1936) to respect a bilingual dictionary. CCA is an algorithm that takes as input pairs of observations $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$ drawn from two different feature spaces ($\mathbf{x}_i \in \mathbf{X}, \mathbf{y}_i \in \mathbf{Y}$), and finds directions $\mathbf{d}_1$ and $\mathbf{d}_2$ such that the linear projections $\{\mathbf{d}_1{}^{\intercal}\mathbf{x}_i\}_{i=1}^{n}$ and $\{\mathbf{d}_2{}^{\intercal}\mathbf{y}_i\}_{i=1}^{n}$ are maximally correlated.

For applying CCA, first matrices $\mathbf{W}' \subseteq \mathbf{W}, \mathbf{V}' \subseteq \mathbf{V}$ are constructed such that $|\mathbf{W}'| = |\mathbf{V}'|$ and the corresponding rows $(\mathbf{w}_i, \mathbf{v}_i)$ in the matrices are representations of words $(w_i, v_i)$ that are translations of each other. The projection is then computed as:

$$\mathbf{P}_W, \mathbf{P}_V = \mathrm{CCA}(\mathbf{W}', \mathbf{V}') \tag{3.12}$$

$$\mathbf{W}^* = \mathbf{W}\mathbf{P}_W \qquad \mathbf{V}^* = \mathbf{V}\mathbf{P}_V \tag{3.13}$$

where, $\mathbf{P}_V \in \mathbb{R}^{l \times d}, \mathbf{P}_W \in \mathbb{R}^{m \times d}$ are the projection matrices with $d \leq \min(l, m)$ and the $\mathbf{V}^* \in \mathbb{R}^{|V| \times d}, \mathbf{W}^* \in \mathbb{R}^{|W| \times d}$ are the word vectors that have been aligned using the entries in the bilingual dictionary.

The BiCCA objective can be viewed as the following instantiation of Algorithm 1,[1]

$$\mathbf{W}^0 = \mathbf{W}' \qquad \mathbf{V}^0 = \mathbf{V}' \tag{3.14}$$

$$C(\mathbf{W}, \mathbf{V}) = \|\mathbf{W} - \mathbf{V}\|^2 + \gamma\left(\mathbf{V}^{\intercal}\mathbf{W}\right) \tag{3.15}$$

$$A(\mathbf{W}) = \|\mathbf{W}\|^2 - 1 \qquad B(\mathbf{V}) = \|\mathbf{V}\|^2 - 1 \tag{3.16}$$

where $\mathbf{W} = \mathbf{W}^0\mathbf{P}_W$ and $\mathbf{V} = \mathbf{V}^0\mathbf{P}_V$, where $\alpha = \beta = \gamma = \infty$ to set hard constraints.

### *3.4.4. Document-level Alignment*

Another inexpensive cross-lingual alignment can be derived by identifying *comparable documents* between two languages. Comparable documents are *thematically* aligned, in that they contain overlapping information, but neither document is an exact translation of the other (Figure 4d). Examples of comparable documents in different languages are news

[1] This formulation is described in Section 6.5 of (Hardoon et al., 2004)

articles that report on the same event, captions of the same image, or Wikipedia pages describing the same entity. Such document-level alignments offer an attractive alternative to using word or sentence level alignments to train cross-lingual representations, as they are relatively easier to obtain. This section describes the approach proposed by Vulić and Moens (2015), henceforth referred as BiVCD (Bilingual Vectors from Comparable Data), that exploits this cross-lingual information.

Let $\mathcal{D}_e$ and $\mathcal{D}_f$ denote a pair of comparable documents with length (in words) $p$ and $q$ respectively (assume $p > q$). BiVCD first merges these two comparable documents into a single *pseudo-bilingual* document. The pseudo-bilingual document is constructed by assuming a *monotonic* alignment between the words of $\mathcal{D}_e$ and $\mathcal{D}_f$. That is, every $R^{th}$ word of the merged pseudo-bilingual document is picked sequentially from $\mathcal{D}_f$, where $R = \lfloor \frac{p}{q} \rfloor$ is the *length ratio* of two documents. Finally, a skip-gram model is trained on the corpus of pseudo-bilingual documents, to generate vectors for all words in $\mathbf{W}^* \cup \mathbf{V}^*$. The vectors constituting $\mathbf{W}^*$ and $\mathbf{V}^*$ can then be easily identified.

Instantiating BiVCD in the general algorithm is obvious — $C(\mathbf{W}, \mathbf{V})$ assumes the familiar SGNS objective over the pseudo-bilingual document,

$$C(\mathbf{W}, \mathbf{V}) = -\sum_{s \in W \cup V} \sum_{t \in \textsc{Neighbors}(s)} \log \Pr(t \mid s) \tag{3.17}$$

where $t, s \in W \cup V$ are words in the combined vocabulary, $\textsc{Neighbors}(s)$ is neighborhood of $s$ in the pseudo-bilingual document, and $\Pr(t \mid s) \propto \exp(\mathbf{t}^\intercal \mathbf{s})$, similar to SGNS.

Although BiVCD is designed to use comparable corpus, in the experiments it is provided with parallel data (treating two aligned sentences as comparable) to ensure comparability with other methods.

## 3.5. Experiments

The experiments in this section evaluate the different cross-lingual representation learning approaches on four different tasks — monolingual lexical similarity (Section 3.5.1), cross-lingual dictionary induction (Section 3.5.2), cross-lingual document classification (Section 3.5.3) and cross-lingual dependency parsing (Section 3.5.4). The aim of the monolingual lexical similarity task is to assess how much the English representation benefits from the cross-lingual training. On the other hand, the aim of the remaining experiments is to measure the ability of the cross-lingually trained representations to facilitate transfer of semantic or syntactic knowledge across languages.

### *3.5.1. Monolingual Lexical Similarity*

The first experiment evaluates if the inclusion of cross-lingual knowledge improves the quality of English embeddings, by using them for two monolingual lexical semantic tasks.

**Word Similarity.** Word similarity datasets contain word pairs that are assigned similarity ratings by humans. The task evaluates how well the notion of word similarity according to humans is emulated in the vector space.

This experiment uses the SimLex word similarity dataset for English (Hill et al., 2014), which contains 999 word pairs, with a balanced set of noun, adjective and verb pairs. Evaluation is based on the Spearman's rank correlation coefficient (Spearman, 1904) between the human rankings and rankings produced by computing cosine similarity between the vectors of two words. Improvement is considered statistically significant if $p < 0.1$ according to Steiger's method (Steiger, 1980) for calculating the significance of differences between two *dependent* correlation coefficients. The Spearman rank correlation coefficient and Steiger's test are described in detail in Appendix A.1.

Table 3 shows the performance of the English embeddings induced using different cross-lingual alignments on the SimLex word similarity task, for different language pairs. As a

| Lang. Pair | Mono | BiSkip | BiCVM | BiCCA | BiVCD |
|---|---|---|---|---|---|
| English–German | 0.29 | 0.34 | **0.37** | 0.30 | 0.32 |
| English–French | 0.30 | 0.35 | **0.39** | 0.31 | 0.36 |
| English–Swedish | 0.28 | 0.32 | **0.34** | 0.27 | 0.32 |
| English–Chinese | 0.28 | 0.34 | **0.39** | 0.30 | 0.31 |
| avg. | 0.29 | 0.34 | **0.37** | 0.30 | 0.33 |

Table 3: Word similarity score measured in Spearman's correlation ratio for English on SimLex-999. The best score for each language pair is shown in **bold**. Scores that are significantly better (per Steiger's Method with $p < 0.1$) than the next lower score are underlined. For example, for English-Chinese, BiCVM is significantly better than BiSkip, which in turn is significantly better than BiVCD.

baseline, the score obtained by monolingual English embeddings trained on the respective English side of each language is shown in column marked Mono. For all language pairs (except for English-Swedish for BiCCA), the bilingually trained vectors achieve better scores than the mono-lingually trained vectors.

Overall, across all language pairs, BiCVM is the best performing model in terms of Spearman's correlation, but its improvement over BiSkip and BiVCD is often insignificant. It is notable that 2 of the 3 top performing models, BiCVM and BiVCD, need sentence aligned and document-aligned corpus only, which are easier to obtain than parallel data with word alignments required by BiSkip.

**Qvec.** Qvec is another intrinsic evaluation metric for estimating the quality of English word vectors, proposed by Tsvetkov et al. (2015). The score produced by Qvec measures how well a given set of word vectors is able to quantify linguistic properties of words, with higher being better. Tsvetkov et al. (2015) showed high Qvec scores has strong correlation with good performance on downstream semantic applications.

Table 4 shows Qvec scores for the English embeddings learnt using different cross-lingual alignments. On an average, BiSkip achieves the best score across language pairs, followed by Mono (mono-lingually trained English vectors), BiVCD and BiCCA. A possible explanation for why Mono scores are better than those obtained by some of the cross-lingual models is

| Lang. Pair | Mono | BiSkip | BiCVM | BiCCA | BiVCD |
|---|---|---|---|---|---|
| English–German | 0.39 | **0.40** | 0.31 | 0.33 | 0.37 |
| English–French | 0.39 | **0.40** | 0.31 | 0.33 | 0.38 |
| English–Swedish | 0.39 | **0.39** | 0.31 | 0.32 | 0.37 |
| English–Chinese | 0.40 | **0.40** | 0.32 | 0.33 | 0.38 |
| avg. | 0.39 | **0.40** | 0.31 | 0.33 | 0.38 |

Table 4: Intrinsic evaluation of English word vectors measured in terms of Qvec score across models. Best scores for each language pair is shown in **bold**.

that Qvec measures monolingual semantic content based on a linguistic oracle made for English. Cross-lingual training might affect these semantic properties arbitrarily.

Interestingly, BiCVM, which was the best model according to SimLex, ranks last according to Qvec. The fact that the best models according to Qvec and SimLex are different reinforces observations made in previous work that performance on word similarity tasks alone does not reflect quantification of linguistic properties of words (Tsvetkov et al., 2015; Schnabel et al., 2015).

### *3.5.2. Cross-lingual Dictionary Induction*

The cross-lingual dictionary induction task (Vulić and Moens, 2013; Gouws et al., 2015; Mikolov et al., 2013b) judges how good cross-lingual embeddings are at detecting word pairs that are translationally equivalent (e.g., *cow* in English and *vache* in French).

The experiment follows the setup of Vulić and Moens (2013), and evaluates if the correct translation of an English word is present in its top-10 nearest neighbors in the target language, using the word embeddings and cosine similarity as a measure of distance. Instead of manually creating a gold cross-lingual dictionary for evaluation, *gold dictionaries* are derived using the Open Multilingual WordNet data released by Bond and Foster (2013). The data includes synset alignments across 26 languages with over 90% accuracy. First, words from each synset whose corpus frequency is less than 1000 are pruned. Then, for each pair of aligned synsets $s_1 = \{k_1, k_2, \cdots\}$ $s_2 = \{g_1, g_2, \cdots\}$, all elements from the set

| $l_1$ | $l_2$ | BiSkip | BiCVM | BiCCA | BiVCD |
|---|---|---|---|---|---|
| English | German | **79.7** | 74.5 | 72.4 | 62.5 |
| | French | **78.9** | 72.9 | 70.1 | 68.8 |
| | Swedish | **77.1** | 76.7 | 74.2 | 56.9 |
| | Chinese | **69.4** | 66.0 | 59.6 | 53.2 |
| avg. | | **76.3** | 72.5 | 69.1 | 60.4 |

Table 5: Cross-lingual dictionary induction results (top-10 accuracy). The same trend was also observed across models when computing MRR (mean reciprocal rank).

$\{(k, g) \mid k \in s_1, g \in s_2\}$ are included in the gold dictionary, where $k$ and $g$ are the lemmas. Using this approach, dictionaries of sizes 1.5k, 1.4k, 1.0k and 1.6k pairs are constructed for English–French, English–German, English–Swedish and English–Chinese respectively. These dictionaries serve as the gold dictionaries for evaluating the cross-lingual embeddings.

Top-10 accuracy is reported in the experiments, which is the fraction of the entries $(e, f)$ in the gold dictionary for which $f$ belongs to the list of top-10 neighbors of the word vector of $e$, according to the induced cross-lingual embeddings. The results are shown in Table 5. From the results, it can be seen that for dictionary induction, the performance improves with the quality of supervision. As we move from cheaply supervised methods (e.g., BiVCD) to more expensive supervision (e.g., BiSkip), the accuracy improves. This suggests that for cross-lingual similarity tasks, the more expensive the cross-lingual knowledge available, the better. Models using weak supervision like BiVCD perform poorly in comparison to models like BiSkip and BiCVM, with performance gaps greater than 10 points on an average.

### *3.5.3. Cross-lingual Document Classification*

To judge how good are these cross-lingual representations at facilitating model transfer for a downstream *semantic* task, we use them as features for cross-lingual document classification (CLDC).

In the CLDC task, a document classifier is trained using the document representations in language $l_1$, and then the trained model is tested on documents from language $l_2$. The

| $l_1$ | $l_2$ | BiSkip | BiCVM | BiCCA | BiVCD |
|---|---|---|---|---|---|
| English | German | **85.2** | $\underline{85.0}$ | 79.1 | 79.9 |
|  | French | $\underline{\mathbf{77.7}}$ | 71.7 | 70.7 | 72.0 |
|  | Swedish | $\underline{\mathbf{72.3}}$ | $\underline{69.1}$ | $\underline{65.3}$ | 59.9 |
|  | Chinese | **75.5** | 73.6 | 69.4 | $\underline{73.0}$ |
| German | English | **74.9** | $\underline{71.1}$ | 64.9 | $\underline{74.1}$ |
| French |  | $\underline{\mathbf{80.4}}$ | 73.7 | $\underline{75.5}$ | $\underline{77.6}$ |
| Swedish |  | $\underline{73.4}$ | 67.7 | 67.0 | $\underline{\mathbf{78.2}}$ |
| Chinese |  | **81.1** | 76.4 | 77.3 | $\underline{80.9}$ |
| avg. |  | **77.6** | 73.5 | 71.2 | 74.5 |

Table 6: Cross-lingual document classification accuracy when trained on language $l_1$, and test on language $l_2$. Best scores for each language is shown in **bold**. Majority baselines for English → $l_2$ and $l_1$ → English are 49.7% and 46.7% respectively, for all languages. Scores significantly better (per McNemar's Test, $p < 0.05$) than the next best score are underlined. For example, for Swedish → English, BiVCD is significantly better than BiSkip, which is significantly better than BiCVM.

document representation is computed by taking the TF-IDF (Luhn, 1957; Sparck Jones, 1972; Salton and Buckley, 1988) weighted average of vectors (from the appropriate language) of the words present in the document. By using supervised training data in one language and evaluating without further supervision in another, CLDC assesses whether the learned cross-lingual representations are semantically coherent across multiple languages.

The cross-lingual document classification (CLDC) setup of Klementiev et al. (2012) and the RCV2 Reuters multilingual corpus[2] are used in this experiment. A multi-class classifier is trained using an averaged perceptron (Freund and Schapire, 1999) for 10 iterations, using the document vectors of language $l_1$ in the RCV2 corpus as features, and then tested on the documents in target language $l_2$. When $l_1 =$ English (i.e., for the English → $l_2$ direction), the model is trained on 1000 documents in English and tested on 1800 documents of target language $l_2$. When $l_2 =$ English (i.e., for the $l_1$ → English direction), the model is trained on 1000 documents of language $l_1$ and tested on 5000 documents of English. Table 6 shows the classification accuracies of different models across different language pairs.

[2] `http://trec.nist.gov/data/reuters/reuters.html`

| $l$ | Mono | BiSkip | BiCVM | BiCCA | BiVCD |
|---|---|---|---|---|---|
| German | 71.1 | **72.0** | 60.4 | **71.4** | 58.9 |
| French | 78.9 | **80.4** | 73.7 | **80.2** | 69.5 |
| Swedish | 75.5 | **78.2** | 70.5 | **79.0** | 64.5 |
| Chinese | 73.8 | 73.1 | 65.8 | 71.7 | 67.0 |
| avg. | 74.8 | **75.9** | 67.6 | **75.6** | 66.8 |

Table 7: Labeled attachment score (LAS) for dependency parsing when trained and tested on language $l$ (direct-parsing experiment). Mono refers to parser trained with mono-lingually induced embeddings. Scores in **bold** are better than the Mono scores for each language, showing improvement from cross-lingual training.

Table 6 shows that in almost all cases, BiSkip performs significantly better than the remaining models. This suggests that for transferring semantic knowledge across languages via embeddings, using sentence and word level alignment when training is superior to using sentence or word level alignment alone. This observation parallels the trend in the cross-lingual dictionary induction experiments.

It is interesting to note that BiVCD does well on languages like French and Chinese, but poorly on German and Swedish. A possible reason is that BiVCD assumes a monotonic alignment between words in documents to create the pseudo-cross-lingual document for training, which appears to work well with languages with largely the same subject-verb-object order as English.

### *3.5.4. Cross-lingual Dependency Parsing*

How useful are these cross-lingual representations as features for a downstream *syntactic* task? This is evaluated using two experiments — (a) *direct-parsing*: using representations for language $l_1$ trained using some cross-lingual signals as features for training a dependency parser for $l_1$, and (b) *model-transfer*: training a dependency parser on the English Treebank and then transferring the model to language $l_2$ via cross-lingual representations.

The direct-parsing experiment evaluates whether using cross-lingually trained vectors is better than using mono-lingually trained vectors for training dependency parsers. The

| $l_1$ | $l_2$ | BiSkip | BiCVM | BiCCA | BiVCD |
|---|---|---|---|---|---|
| English | German | 49.8 | 47.5 | **51.3** | 49.0 |
|  | French | 65.8 | 63.2 | **65.9** | 60.7 |
|  | Swedish | 56.9 | 56.7 | **59.4** | 54.6 |
| avg. |  | 57.5 | 55.8 | **58.9** | 54.8 |
| German | English | 49.7 | 45.0 | **50.3** | 43.6 |
| French |  | 53.3 | 50.6 | **54.2** | 49.5 |
| Swedish |  | 48.2 | 49.0 | **49.9** | 44.6 |
| avg. |  | 50.4 | 48.2 | **51.5** | 45.9 |

Table 8: Labeled attachment score (LAS) for cross-lingual dependency parsing when trained on language $l_1$, and evaluated on language $l_2$ (model-transfer experiment). The best score for each language is shown in **bold**.

comparison is done against parsers trained using mono-lingually trained word vectors. These vectors are the same used as input to BiCCA. All other settings remain the same.

In the model-transfer experiments, a model is trained using embeddings for language $l_1$ and then tested on language $l_2$, replacing embeddings for language $l_1$ with those of $l_2$. For these experiments, the dependency parsing model from Guo et al. (2015) is used. The model trains a transition-based dependency parser using nonlinear activation function, with the source-side word embeddings as lexical features. These embeddings can be replaced by target-side embeddings at test time. The universal dependency treebank (McDonald et al., 2013) version-2.0 is used for training and evaluation.

The results for the direct-parsing experiments are in Table 7 and the model-transfer experiments are in Table 8. On an average, for both experiments, BiCCA and BiSkip perform better than other models. BiSkip is a close second to BiCCA, with an average performance gap of less than 1 point. For the direct-parsing experiment, improvement over the monolingual embeddings (column Mono in Table 7) was obtained with BiSkip and BiCCA, while BiCVM and BiVCD consistently performed worse. Unlike the semantic transfer tasks considered earlier, the models with expensive supervision requirements like BiSkip and BiCVM could not outperform BiCCA, a model with relatively inexpensive requirements, in both the experiments. A possible reason for this is that approaches like BiCVM and BiVCD operate on sentence level contexts to learn the embeddings, which only captures the semantic meaning of the sentences and ignores the internal syntactic structure. As a result, embedding trained using BiCVM and BiVCD are not informative for syntactic tasks. On the other hand, approaches like BiSkip and BiCCA both utilize the word alignment information to train their embeddings and thus do better in capturing some notion of syntax.

## 3.6. Summary

This chapter provided a review and a unifying perspective of different approaches for learning distributional word representations cross-lingually. A myriad of approaches of learning cross-lingual word representations have been proposed, and reviewing all of them is beyond the scope of the thesis (for a more thorough survey, see Ruder et al. (2018)). Nevertheless, the general principles and the algorithms in use are more-or-less the same. We saw how each of these approaches can be viewed as instances of the same algorithm, revealing the same underlying principle — combine monolingual distributional information with some form of cross-lingual alignment. Experimental evaluation on different semantic and syntactic tasks revealed some cross-lingual alignments are a natural fit for certain tasks (e.g., word level alignments are essential for syntactic transfer), while for others (e.g., dictionary induction) the performance was correlated with the cost and quality of the cross-lingual alignment. Nevertheless, all forms of alignments successfully facilitated model transfer (to varying degrees) across languages, a desiderata of cross-lingual representations.

# CHAPTER 4 : Deriving Cross-lingual Representations from Wikipedia

## 4.1. Introduction

The previous chapter described different approaches for learning cross-lingual representations by incorporating various forms of cross-lingual alignments into distributional information acquired from unstructured text. In addition to unstructured resources, there are also *structured* sources of semantic information, such as encyclopedic resources like Wikipedia. In this chapter, I discuss an approach for learning cross-lingual representation that uses such multilingual encyclopedic resources. Unlike the representations discussed in the previous chapter, all of which appealed to the distributional hypothesis, the representation used in this chapter appeals to the bag of words hypothesis (Salton et al., 1975; Salton and Buckley, 1988) described in Chapter 2. Specifically, I show how we can exploit the inter-lingual structure of Wikipedia to apply the bag-of-words hypothesis *across* languages, and learn shared sparse semantic representations, that can be used to classify documents in multiple languages with no supervision.

## 4.2. Background

This chapter builds on two related research directions — an approach to derive semantic representations of documents using encyclopedic resources, and an approach to perform unsupervised document classification. I briefly describe them below.

### *4.2.1. Explicit Semantic Representations (ESA)*

Gabrilovich and Markovitch (2009) noticed that distributional approaches for learning word representations are completely divorced from efforts to organize world knowledge (like WordNet (Miller, 1995), SUMO (Niles and Pease, 2001), CyC (Lenat, 1995)). As a result, these statistical representations need lots of data to learn relations (e.g. 'Clinton' is related to 'Arkansas') that are explicitly stated in compiled knowledge resources, because such co-occurrences might not be frequent enough even in large corpora.

| word<br>entity | president | executive | … | administration |
|---|---|---|---|---|
| Barack_Obama | 95 | 11 | ⋯ | 22 |
| United_States | 28 | 3 | ⋯ | 3 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| London | 0 | 2 | ⋯ | 3 |
| | English | | | |

Table 9: Concept-term matrix constructed using the English Wikipedia. Explicit Semantic Analysis (ESA) representations are derived from the matrix above by reweighing each entry using TF-IDF and normalizing the columns. Each column describes a word in English using a "bag-of-concepts" in Wikipedia.

However, learning good representations from such compiled knowledge resources is not straight-forward. Most of these resources are relatively small in size due to the expensive manual effort involved in creating them, and are unlikely to scale to large vocabularies. Nevertheless, this manual effort is necessary to ensure high quality. What is needed is a resource that compiles human knowledge at scale while maintaining its high quality. Gabrilovich and Markovitch (2009) argued that Encyclopedias like Wikipedia are one such resource, and showed how the meaning for words and longer pieces of text can be derived *directly* from such encyclopedic resources.

Wikipedia is an collaboratively maintained online encyclopedic resource consisting of description of entities, events, and technical terms in hypertext document, referred to as Wikipedia *articles* (or pages). Wikipedia is constructed by thousands of volunteer editors around the world, who not only generate content but also ensure its quality. The English Wikipedia contains over 5 million articles, dwarfing other encyclopedic resources like Britannica in both size and quality (Giles, 2005), and growing in size every year.[1]

Gabrilovich and Markovitch (2009) introduced Explicit Semantic Analysis (ESA), a method to exploit encyclopedic resources like Wikipedia to generate semantic representations of

[1] An average of 605 articles were added *everyday* to the English Wikipedia during 2017-2018, according to https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia.

words and documents. ESA assumes each Wikipedia article $\mathcal{A}$ corresponds to a concept, and each word can be represented by the concepts that contain it. ESA computes a concept-term matrix $\mathbf{T}$ (such as one shown in Table 9) of size $|C| \times |V|$ from Wikipedia, where $V$ is the vocabulary of the English Wikipedia, and $C = \{\mathcal{A}_1, \cdots, \mathcal{A}_{|C|}\}$ is the set of Wikipedia articles, such that each row corresponds to an English Wikipedia article, while each column corresponds to a word in the vocabulary. The entry $\mathbf{T}[i,j]$ denotes the TF-IDF (Luhn, 1957; Sparck Jones, 1972; Salton and Buckley, 1988) value of the $j^{th}$ word of the vocabulary, in the Wikipedia page $\mathcal{A}_i$. The matrix $\mathbf{T}$ will be sparse, for reasons similar to those discussed in Section 2.6. The (sparse) explicit semantic representation of word $w$ (whose index in the vocabulary is $i$) is the $i^{th}$ column of $\mathbf{T}$,

$$\mathbf{\Phi}(w) = \begin{bmatrix}\Phi(\mathcal{A}_1, w) & \dots & \Phi(\mathcal{A}_{|C|}, w)\end{bmatrix} \in \mathbb{R}^{|C|} \tag{4.1}$$

where $\Phi(\mathcal{A}_i, w)$ is the weight indicating how important word $w$ is in the Wikipedia article $\mathcal{A}_i$. Note that these representations are *explicit* (as opposed to latent), as the dimensions of the ESA representation convey meaningful information about concepts relevant to a word, unlike dense vector representations like LSA.

To generate the representation of a document $\mathcal{D}$ using ESA, a TF-IDF weighted average of the ESA representation of words in the document is computed, as in Section 3.5.3. A document $\mathcal{D}$ can be viewed as an indicator vector over words in the vocabulary,

$$\mathcal{D} = \begin{bmatrix}w_1 & \dots & w_V\end{bmatrix} \in \mathbb{R}^{|V|} \tag{4.2}$$

where $V$ is the vocabulary and $w_i \in \{0, 1\}$. If $p_i$ is the inverse document frequency (IDF) of $w_i$ in English Wikipedia, then the vector representation for $\mathcal{D}$ is,

$$\mathbf{\Phi}(\mathcal{D}) = \frac{1}{V} \sum_i p_i \mathbf{\Phi}(w_i) \tag{4.3}$$

Gottron et al. (2011) showed that ESA are a kind of generalized vector space model (Wong et al., 1985), where document similarity is affected by correlations between the different context dimensions (encoded as $\mathbf{G}$), i.e., similarity($x$,$y$) = $\boldsymbol{x}^{\intercal}\mathbf{G}\boldsymbol{y}$.

I will use $\mathbf{\Phi}(\mathcal{D})$ and $\mathbf{\Phi}(w)$ to denote the ESA representation of a document $\mathcal{D}$ and a word $w$ respectively in the rest of the chapter.

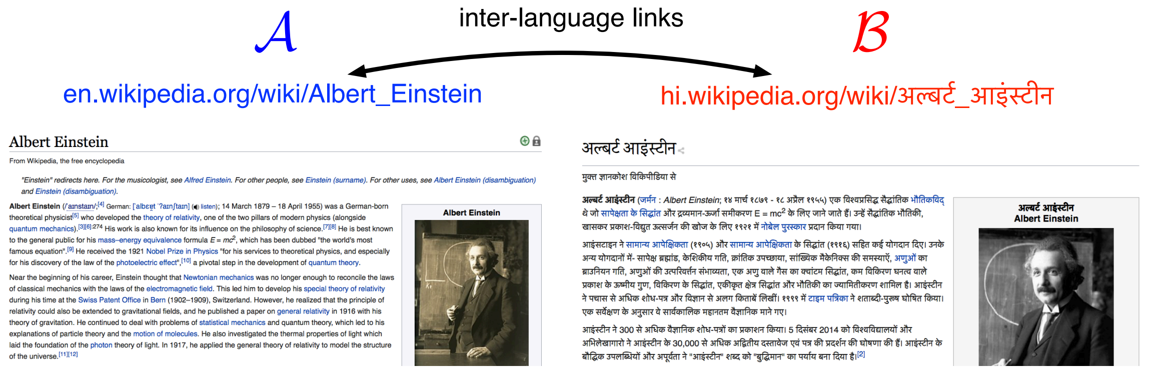
### *4.2.2. Dataless Document Classification using ESA*

Humans can assign a label to a document without requiring any training examples, simply by understanding what the label means. However, traditional document classification approaches treat the labels as atomic symbols (similar to one-hot representations in Chapter 2), without ascribing any meaning to them. As a result, these methods require supervision to train a model to map documents to these atomic symbols. *How can one assign a more meaningful representation to both documents and labels so that supervision is not needed?* Chang et al. (2008) showed how one achieve this by using ESA representations discussed earlier, by representing documents and labels in the same semantic space. As a result, classification can be done simply by a nearest neighbor search, without requiring any supervision. They named their approach *dataless classification* which we describe below.[2]

Let $\mathcal{D}$ denote a document, and $\{l_1, \cdots, l_m\}$ denote the set of $m$ possible labels that can be assigned. Chang et al. (2008) assumed that each label $l_i$ is accompanied by a short description $desc(l_i)$ that contains prototypical terms that illustrate the semantics of the label. For instance, the label *sport* can be described by the terms "baseball basketball hockey" and the label *politics* can be described as "democrats gun-rights congress constitution".

Dataless classification generates representation $\mathbf{\Phi}(\mathcal{D})$ for a document $\mathcal{D}$ and representations $\{\mathbf{\Phi}(l_1), \ldots, \mathbf{\Phi}(l_m)\}$ for each of the $m$ labels, in the *same* semantic space. The document representation $\mathbf{\Phi}(\mathcal{D})$ is constructed as above (Equation 4.3), while the label representation

[2] The name is a bit of a misnomer, as data (from Wikipedia) is being used. It is dataless in the sense that it is unsupervised.


$\mathcal{A}$
inter-language links
$\mathcal{B}$
en.wikipedia.org/wiki/Albert_Einstein
hi.wikipedia.org/wiki/अल्बर्ट_आइंस्टीन

Albert Einstein

From Wikipedia, the free encyclopedia

*"Einstein" redirects here. For the musicologist, see Alfred Einstein. For other people, see Einstein (surname). For other uses, see Albert Einstein (disambiguation) and Einstein (disambiguation).*

**Albert Einstein** (/ˈaɪnstaɪn/;[4] German: [ˈalbɛɐ̯t ˈʔaɪnʃtaɪn] (listen); 14 March 1879 – 18 April 1955) was a German-born theoretical physicist[5] who developed the theory of relativity, one of the two pillars of modern physics (alongside quantum mechanics).[3][6]:274 His work is also known for its influence on the philosophy of science.[7][8] He is best known to the general public for his mass–energy equivalence formula $E = mc^2$, which has been dubbed "the world's most famous equation".[9] He received the 1921 Nobel Prize in Physics "for his services to theoretical physics, and especially for his discovery of the law of the photoelectric effect",[10] a pivotal step in the development of quantum theory.

Near the beginning of his career, Einstein thought that Newtonian mechanics was no longer enough to reconcile the laws of classical mechanics with the laws of the electromagnetic field. This led him to develop his special theory of relativity during his time at the Swiss Patent Office in Bern (1902–1909), Switzerland. However, he realized that the principle of relativity could also be extended to gravitational fields, and he published a paper on general relativity in 1916 with his theory of gravitation. He continued to deal with problems of statistical mechanics and quantum theory, which led to his explanations of particle theory and the motion of molecules. He also investigated the thermal properties of light which laid the foundation of the photon theory of light. In 1917, he applied the general theory of relativity to model the structure of the universe.[11][12]

Albert Einstein

अल्बर्ट आइंस्टीन

मुक्त ज्ञानकोश विकिपीडिया से

**अल्बर्ट आइंस्टीन** (जर्मन : *Albert Einstein*; १४ मार्च १८७९ - १८ अप्रैल १९५५) एक विश्वप्रसिद्ध सैद्धांतिक भौतिकविद् थे जो सापेक्षता के सिद्धांत और द्रव्यमान-ऊर्जा समीकरण $E = mc^2$ के लिए जाने जाते हैं। उन्हें सैद्धांतिक भौतिकी, खासकर प्रकाश-विद्युत ऊत्सर्जन की खोज के लिए १९२१ में नोबेल पुरस्कार प्रदान किया गया।

आइंसटाइन ने सामान्य आपेक्षिकता (१९०५) और सामान्य आपेक्षिकता के सिद्धांत (१९१६) सहित कई योगदान दिए। उनके अन्य योगदानों में- सापेक्ष ब्रह्मांड, केशिकीय गति, क्रांतिक उपच्छाया, सांख्यिक मैकेनिक्स की समस्याऍ, अणुओं का ब्राउनियन गति, अणुओं की उत्परिवर्त्तन संभाव्यता, एक अणु वाले गैस का क्वांटम सिद्धांत, कम विकिरण घनत्व वाले प्रकाश के ऊष्मीय गुण, विकिरण के सिद्धांत, एकीकृत क्षेत्र सिद्धांत और भौतिकी का ज्यामितीकरण शामिल है। आइंस्टीन ने पचास से अधिक शोध-पत्र और विज्ञान से अलग किताबें लिखीं। १९९९ में टाइम पत्रिका ने शताब्दी-पुरुष घोषित किया। एक सर्वेक्षण के अनुसार वे सार्वकालिक महानतम वैज्ञानिक माने गए।

आइंस्टीन ने 300 से अधिक वैज्ञानिक शोध-पत्रों का प्रकाशन किया। 5 दिसंबर 2014 को विश्वविद्यालयों और अभिलेखागारो ने आइंस्टीन के 30,000 से अधिक अद्वितीय दस्तावेज एवं पत्र की प्रदर्शन की घोषणा की हैं। आइंस्टीन के बौद्धिक उपलब्धियों और अपूर्वता ने "आइंस्टीन" शब्द को "बुद्धिमान" का पर्याय बना दिया है।[2]

अल्बर्ट आइंस्टीन
Albert Einstein

Figure 5: Inter-language links in English Wikipedia and Hindi Wikipedia link the corresponding articles that describe the entity `Albert_Einstein` in both Wikipedias.

is computed by treating its description as a 'document'. For example, the labels *sports* and *politics* will be assigned representations $\mathbf{\Phi}(desc(sports))$ and $\mathbf{\Phi}(desc(politics))$ respectively. As both the documents and the labels now reside in the same space, we can predict the label(s) that maximizes the similarity of the label and the document representation,

$$l^* = \arg\max_i \frac{\mathbf{\Phi}(\mathcal{D})^\intercal \mathbf{\Phi}(l_i)}{\|\mathbf{\Phi}(\mathcal{D})\| \|\mathbf{\Phi}(l_i)\|} \tag{4.4}$$

Besides being "dataless", an attractive property of this approach is that classification can be done *on-the-fly*, that is, the label space is not necessarily known in advance. One can easy operate with a new label at no cost, by simply using it's description to generate the label representation at test time.

## 4.3. Inter-lingual Structure of Wikipedia

While the ESA representations described earlier used the English Wikipedia, Wikipedias exist in over 250 languages, where each article describes a concept in that language. Furthermore, concepts described in different Wikipedias often overlap, indirectly introducing information redundancy across languages, as some concepts have articles in more than one languages. Such articles in Wikipedia are linked through *inter-language links*. Formally, an inter-language link $\mathcal{A}_i \leftrightarrow \mathcal{B}_j$ indicates that article $\mathcal{A}_i$ in the language $l_1$'s Wikipedia and article $\mathcal{B}_j$ in language $l_2$'s Wikipedia describe the same concept. For instance, Fig-

| Language | # Articles | Intersection | % |
|---|---|---|---|
| German | 2.06M | 1.06M | 52 |
| French | 1.87M | 1.21M | 65 |
| Italian | 1.36M | 927k | 68 |
| Spanish | 1.29M | 850k | 66 |
| Chinese | 942k | 519k | 55 |
| Arabic | 521k | 330k | 63 |
| Turkish | 292k | 203k | 69 |
| Tamil | 104k | 61k | 59 |
| Tagalog | 69k | 53k | 77 |

Table 10: Statistics showing number of articles in Wikipedias for 9 languages, and the size of the intersection (as identified by inter-language links) with the English Wikipedia. The last column shows the relative size of the intersection (in % age). For instance, 52% of German Wikipedia articles have an English Wikipedia counterpart.

ure 5 shows that the article अल्बर्ट_आइंस्टीन in the Hindi Wikipedia corresponds to the article `Albert_Einstein` in the English Wikipedia. These inter-language links identify the same concept in different Wikipedias, thereby unifying the concepts across languages. Table 10 shows the statistics of the number of articles in Wikipedias for 9 languages, along with the size of the intersection of the respective Wikipedia with the articles in English Wikipedia.

### 4.4. Cross-lingual ESA (CLESA)

*How can one learn ESA representations for words outside the English vocabulary?* The fact that one can resolve Wikipedia articles written in different languages to the same concept can help us define cross-lingual extension of ESA representations, henceforth referred as CLESA. CLESA appeals to the bag-of-words hypothesis across words in different languages — if a word in language $l_1$ (e.g., क्रिकेटर in Hindi) and a word in language $l_2$ (e.g., *cricketer* in English) describe similar concepts in their respective Wikipedias, then they are similar.[3]

For generating CLESA vectors for languages $l_1$ and $l_2$ with $C_1$ and $C_2$ articles each, the concept-term matrix has to be constructed appropriately. First, the concept space $C'$ is com-

[3] The Hindi word translates to *cricketer*.

| word<br>entity | president | executive | … | administration | torre | grande | … | politico | ओलम्पिक | डेमोक्रेटिक | … | हवाई |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| `Barack_Obama` | 95 | 11 | … | 22 | 1 | 1 | … | 17 | 0 | 2 | … | 2 |
| `United_States` | 28 | 3 | … | 3 | 0 | 10 | … | 0 | 0 | 0 | … | 2 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| `London` | 0 | 2 | … | 3 | 10 | 12 | … | 2 | 0 | 0 | … | 2 |
|  | English |  |  |  | Spanish |  |  |  | Hindi |  |  |  |

Table 11: Concept-term matrix constructed using the English, Spanish and Hindi Wikipedias (compare to Table 9). From left to right, the words in Spanish and Hindi translate to *tower*, *big*, *politician*, *olympic*, *democratic*, *hawaii* respectively. The term counts are computed in the respective article in the Spanish or Hindi Wikipedia. CLESA representations are derived from this matrix by weighing each entry using TF-IDF and normalizing the columns. Each column describes a word using a "bag-of-concepts" in Wikipedia.

puted by identifying concepts that have articles in both $l_1$'s Wikipedia and $l_2$'s Wikipedia,

$$C' = \{C_1, \ldots, C_{|C'|}\} = \{\ldots, \mathcal{A}_i, \ldots\} \cap \{\ldots, \mathcal{B}_j, \ldots\} \tag{4.5}$$

Here the intersection is computed by identifying corresponding articles $\mathcal{A}_i \leftrightarrow \mathcal{B}_j$ using the inter-language links, ignoring articles with no counterparts. Next, the vocabulary $V'$ is extended to languages other than English by computing it over the Wikipedia of the relevant languages. This way, a concept-term matrix of size $C' \times V'$ is constructed, where $V' = V_1 \cup V_2$ is the union of vocabularies of Wikipedia in different languages and $C'$ is the intersection of concepts appearing in $l_1$'s and $l_2$'s Wikipedia. An example concept-term matrix for English, Spanish and Hindi is shown in Table 11. The CLESA representation of a word $w \in V'$ can be defined similar to Equation (4.1),

$$\mathbf{\Phi}(w) \triangleq \begin{bmatrix} \Phi(C_1, w) & \ldots & \Phi(C_{|C'|}, w) \end{bmatrix} \in \mathbb{R}^{|C'|} \tag{4.6}$$

The document and the label representation can be defined similarly.

**Other Cross-lingual ESA Variants.** Other cross-lingual extensions of ESA that are similar to ours have also been proposed. Potthast et al. (2008) and Sorg and Cimiano (2012) generated ESA like representations for performing cross-lingual information retrieval. Independent to us, Søgaard et al. (2015) used an inverted index constructed from Wikipedia to represent words using Wikipedia concepts, and used these as features for performing downstream tasks like POS tagging, word alignment, document classification and dependency parsing. Later, Camacho-Collados et al. (2016) developed NASARI representation using inter-language links in Wikipedia, with the goal of performing lexical semantic tasks like word sense clustering across languages. Though superficially different, the underlying idea among all these works are the same as ours.

## 4.5. Experimental Setup and Experiments

The aim of the experiments is to: (a) quantify the effectiveness of CLESA based document classification in terms of number of labeled examples needed to achieve similar performance. (b) determine if CLESA based document classification superior to monolingual ESA based document classification in the target language.

**Building CLESA.** First the relevant Wikipedias were tokenized using Lucene's tokenizer[4], and articles with <100 words or <5 inter-language links were removed. This step removes short articles and disambiguation pages from the respective Wikipedias. The remaining articles were used to construct the concept-term matrix for CLESA by creating an inverted index using Lucene. For classification, the label space is assumed to be provided in English. For language $L$ the CLESA concept-term matrix is computed by considering the intersection of the Wikipedia articles with English, as described above.

[4] http://lucene.apache.org

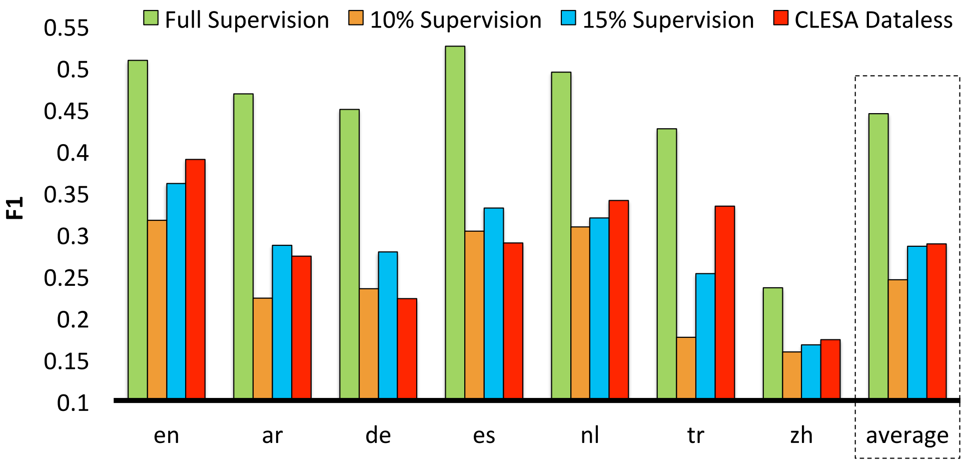


Figure 6: Comparing CLESA based document classification with supervised classification on the TED dataset (averaged macro-F1 scores over 15 labels) for 7 languages — English (en), Arabic (ar), German (de), Spanish (es), Dutch (nl), Turkish (tr) and Chinese (zh).

### *4.5.1. CLESA v/s Supervised Classification*

This experiment compares CLESA based document classification with a fully supervised model on the TED dataset[5], and a model trained using 10% and 15% of the supervision, sampled randomly over 10 different runs (the scores are averaged). The TED dataset contains ≈1200 labeled documents in total (1000 train + 200 test) per language, with a label space of 15 topic labels. A $\ell_2$-regularized linear support vector machine trained using bag-of-words features is used as the supervised method. Results are in Figure 6.

Figure 6 shows that CLESA based dataless classification is comparable to supervised learning with 10% labeled data (100 examples), and a little worse than supervised learning with 15% labeled data (150 examples). This suggests that one can get accuracy competitive to that of a supervised classifier with 100 examples using the unsupervised CLESA based document classification, which is an encouraging result.

[5] `http://www.clg.ox.ac.uk/tedcldc.html`

| metric(p) | accuracy@1 | | accuracy@3 | |
|---|---|---|---|---|
| range | Mono | CLESA | Mono | CLESA |
| $1 \geq p \geq 0.9$ | 2 | 20 | 5 | 28 |
| $0.9 > p \geq 0.8$ | 7 | 8 | 9 | 8 |
| $0.8 > p \geq 0.7$ | 10 | 7 | 16 | 8 |
| $0.7 > p \geq 0$ | 69 | 53 | 58 | 44 |

Table 12: Comparing monolingual ESA (Mono) and cross-lingual ESA (CLESA) for classification on translated documents from 20-Newsgroup. The table shows the number of languages which achieve accuracies in a certain range with a given ESA representation (Mono or CLESA). For instance, CLESA achieves accuracy@1 in the range of 70% to 80% (i.e., $0.8 > p \geq 0.7$) for 7 languages. It can be seen that CLESA achieves high accuracies (e.g., accuracy@1 lies in $1 \geq p \geq 0.9$) for 20 languages, whereas the same is true for monolingual ESA for 2 languages only.

### *4.5.2. CLESA v/s Monolingual ESA*

An alternative approach to perform cross-lingual document classification is to first translate the label space to the target language and then perform monolingual ESA based classification. This experiment compares this approach with CLESA based document classification. The test set is constructed by translating a set of 100 English documents from the 20-Newsgroup dataset (Lang, 1995) and their label descriptions to language $L$ (all 88 languages supported by Google Translate). Then, monolingual ESA representations are constructed using language $L$'s Wikipedia, and used for dataless document classification. The results are compared to CLESA based document classification results in Table 12.

From Table 12, it can be seen that CLESA achieves high accuracies ($1 \geq p \geq 0.9$) for a larger number of languages (for instance, 20 compared to 2 for accuracy@1) compared to monolingual ESA based classification. This suggests that even though the number of concepts used for CLESA is less than the monolingual ESA, CLESA is successful for more languages. This result shows that benefit of sharing information across languages through a shared semantic space, rather than using a monolingual semantic space.

## 4.6. Summary

This chapter described an approach to learn cross-lingual word representations using multilingual encyclopedic resources like Wikipedia by appealing to the bag-of-words hypothesis. Key to this was the inter-lingual structure of Wikipedia, arising from information redundancy across Wikipedias in different languages, that allowed us to identify equivalent concepts in two or more languages. These representations can then be used in a 'dataless' classification approach for labeling documents in an unsupervised manner, by projecting the document and labels in the same semantic space. Unlike the representation in the previous chapters, these cross-lingual representations were explicit and hence interpretable, where the dimensions correspond to concepts in Wikipedia to which the word is relevant.

Interesting extensions to ESA and CLESA remain open. For instance, hybrid approaches that combine the distributional information extracted from corpus co-occurrences with the explicit representation of ESA/CLESA could combine the benefits of both these paradigms. A limitation of ESA and CLESA like representations that all word co-occurrences in the same document are considered equal regardless of the distance between the words, something accounted for in distributional representations like SGNS and GloVe. Hybrid approaches can potentially alleviate this limitation. Extensions of CLESA can also be developed by, say, comparing concepts that behave similarly over time as in Radinsky et al. (2011) across languages. Such meta-data is available in the form article revision histories in Wikipedia. Other structural properties of Wikipedia, like article link structure or category information, can also be incorporated to learn better ESA/CLESA representations (Scholl et al., 2010).

# CHAPTER 5 : Multi-sense Representation Learning with Cross-lingual Signals

## 5.1. Introduction

Traditional distributional word representations encode the meaning of a word in a *single* vector. This does not allow a word to have different meanings (or *senses*) in different contexts, and thus these representations are unable to reflect polysemy. For instance, the meaning of the word *screen* is different in a medical context and an electronics context, something a single vector cannot adequately capture. Several algorithms that rely on the "one-sense per collocation" hypothesis of Yarowsky (1993, 1995) have been developed to learn *multi-sense* representations, which allow a single word to have multiple representations, one for each sense. However, this hypothesis provides only a weak signal for sense identification, and such algorithms require large amount of training data to learn good sense representations.

In this chapter, I describe an approach to learning multi-sense word representation using cross-lingual translational information and non-parametric sense modeling. Experiments show that after training on a small multilingual corpus, this approach achieves performance competitive to a model trained monolingually on a much larger corpus.

Why do we expect incorporating cross-lingual translational information to aid in learning sense representations? The intuition behind this is a well established fact in the literature that *different senses of the same word may be translated into different words in another language* (Dagan and Itai, 1994; Resnik and Yarowsky, 1999; Diab and Resnik, 2002; Ng et al., 2003). For example, *bank* in English may be translated to *banc* or *banque* in French, depending on whether the sense is financial or geographical. Such bilingual translational information indirectly allows the model to identify which sense of a word is being used. Indeed, many recent work on learning multi-sense embeddings using cross-lingual signals (Bansal et al., 2012; Guo et al., 2014; Kawakami and Dyer, 2015; Šuster et al., 2016). I build upon this line of work, and discuss why using more than two languages to derive the translational information is an improvement on this idea.

## 5.2. Related Work — Representing Word Senses

I first review different approaches to learn sense-specific word representations, and place our work in their context. These approaches can be broadly categorized into two categories — approaches that utilize lexical resources, and approaches that are purely data-driven.[1]

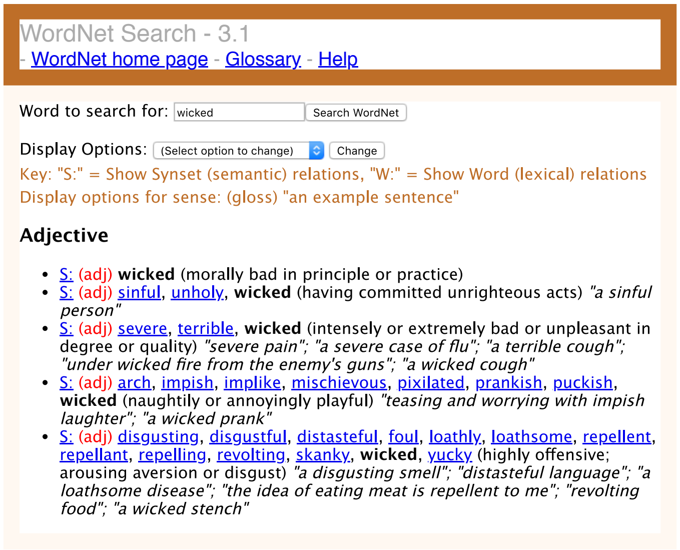
### *5.2.1. Sense Representations using Lexical Resources*

Over the past few decades, many efforts have been made to organize lexical semantic knowledge with some resources like WordNet (Miller, 1995), FrameNet (Baker et al., 1998), BabelNet (Navigli and Ponzetto, 2012) etc., also listing word senses. Naturally, many representation learning approaches have attempted at combining distributional information extracted from raw corpora with information available in these manually created lexical resources. I describe some of the recent work in this direction below, and discuss the limitation of relying on such lexical resources.

Incorporating knowledge expressed in lexical resources into word representations is not a new idea. Early work in this direction was done by Mohammad et al. (2008) and Yih et al. (2012), who combined relations expressed in a thesaurus (Roget, 1852) with distributional representations for differentiating synonyms from antonyms. A more recent line of work that uses lexical resource is that of *retrofitting* (Faruqui et al., 2015a; Jauhar et al., 2015), which infuses lexical semantic resources like WordNet, FrameNet, PPDB (Ganitkevitch et al., 2013), ConceptNet (Speer et al., 2017) into embeddings during a post-processing step. Retrofitting takes as input vector-based word representations and a graph expressing semantic relations between words, and adjusts the word representations such that remain close to their original representations and close to the "neighbors" in the graph.

The most popularly used lexical semantic resource among the ones listed above is WordNet. WordNet (Miller, 1995) is a relation graph describing relationships between nodes denoting *synsets*, where each synset is set of sense-specific word lemmas that are synonymous. A

[1] Also referred as *knowledge-based* and *unsupervised* approaches (Camacho-Collados and Pilehvar, 2018).



WordNet Search - 3.1
- WordNet home page - Glossary - Help

Word to search for: wicked  Search WordNet

Display Options: (Select option to change)  Change
Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations
Display options for sense: (gloss) "an example sentence"

**Adjective**

- S: (adj) **wicked** (morally bad in principle or practice)
- S: (adj) sinful, unholy, **wicked** (having committed unrighteous acts) *"a sinful person"*
- S: (adj) severe, terrible, **wicked** (intensely or extremely bad or unpleasant in degree or quality) *"severe pain"; "a severe case of flu"; "a terrible cough"; "under wicked fire from the enemy's guns"; "a wicked cough"*
- S: (adj) arch, impish, implike, mischievous, pixilated, prankish, puckish, **wicked** (naughtily or annoyingly playful) *"teasing and worrying with impish laughter"; "a wicked prank"*
- S: (adj) disgusting, disgustful, distasteful, foul, loathly, loathsome, repellent, repellant, repelling, revolting, skanky, **wicked**, yucky (highly offensive; arousing aversion or disgust) *"a disgusting smell"; "distasteful language"; "a loathsome disease"; "the idea of eating meat is repellent to me"; "revolting food"; "a wicked stench"*

Figure 7: Senses listed in WordNet for the word *wicked* when used as an adjective. None of the senses listed reflect the sense of *wicked* in the sentence, “Tony Hawk did a *wicked* ollie at the convention”, where *wicked* is used as an adjective denoting awesome-ness.

polysemous word lemma belongs to multiple synsets, one for each sense. For instance, the lemma *good* belongs to the synsets *good.a.01* and *commodity.n.01*, one for its sense used as an adjective (beneficial), and one for its sense used as a noun (commercial object). Relations such as hypernymy, holonymy etc., are encoded using edges between the synsets nodes. Every lemma belonging to a synset is also associated with a sense frequency, denoting how many times that sense of the lemma was used in a sense-tagged corpus. The sense frequencies in WordNet were estimated using the 220k word SemCor corpus (Miller et al., 1993) that was manually tagged with the WordNet senses.

**Limitation of Lexical Resources.** While resources like WordNet are available for many languages (Bond and Foster, 2013), they are not exhaustive in listing all possible senses for the words listed therein. For instance, Figure 7 shows the senses for wicked listed in the English WordNet, and how none describe the sense of *wicked* in “Tony Hawk did a *wicked* ollie at the convention”.[2] Indeed, the number senses of a word is highly dependent on the task and cannot be pre-determined using a lexicon, as argued by Kilgarriff (1997). For

[2] Tony Hawk is a professional skateboarder. An ollie is skateboarding trick.

instance, the sense frequency estimates are hardly representative of all polysemous words in WordNet — Bennett et al. (2016) found out that approximately 61% of the polysemous lemmas have no sense annotations in WordNet 3.0, with only 20% having at least 5 sense annotations. Similarly, some of these resources are automatically created (like BabelNet), and thus may contain incorrect or noisy sense annotations. It is clear that all such lexical resources are incomplete in some form (either the sense is not listed, or the word itself is not present), and are likely to remain so, as language is ever-changing.

Ideally, the word senses should be inferred in a data-driven manner, so that new senses not listed in such lexicons can be discovered. This necessitates the need for developing data-driven approaches to identify the different senses of a word.

### *5.2.2. Data-driven Sense Representations*

Data driven approaches induce word senses in an unsupervised manner from un-annotated corpora, and hold appeal that they can discover new senses of a word that are not listed in any lexical resource. However, learning sense representations in a data-driven manner is complicated by the fact that no large sense-annotated corpora exists. As a result, data-driven approaches by appealing to the one-sense per collocation hypothesis of Yarowsky (1993, 1995). I categorize data-driven approaches for inducing multi-sense embeddings by the learning paradigms they adopt — *two-staged approaches* and *joint learning* approaches.

**Two-staged approaches.** Two-staged approaches first compute some context representations from a raw corpus, that are then used derive sense representations. Popular examples are (Reisinger and Mooney, 2010; Huang et al., 2012), who induce multi-sense embeddings by first clustering the contexts in which a given word appears, and then using the clustering to obtain the sense vectors for that word. The contexts can be topics induced using latent topic models (Liu et al., 2015a,b), or Wikipedia (Wu and Giles, 2015) or coarse part-of-speech tags (Qiu et al., 2014).

A few two-staged approaches also utilize the cross-lingual translational information discussed earlier (Bansal et al., 2012; Guo et al., 2014). For instance, Guo et al. (2014) first sense-tag and project sense annotations using bilingual parallel data, and then learn multi sense representations using a standard neural language modeling objective. This approach ignores any monolingual distributional signal to differentiate senses, and may suffer when the bilingual signal is not sufficient (as I shortly discuss in Section 5.2.3).

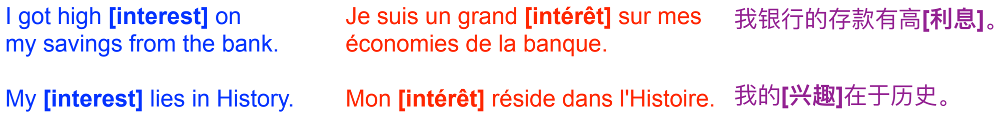
**Joint Learning Approaches.** In contrast to two staged approaches, some approaches (Neelakantan et al., 2014; Li and Jurafsky, 2015) jointly learn the sense clusters and sense-specific embeddings by maximizing a SGNS-like learning objective (Section 2.9), using Bayesian non-parametrics. Neelakantan et al. (2014) proposed two models to learn fixed or dynamic number of sense vectors for each word using a modified version of the skip-gram objective. First, the sense representation of the current word is predicted as the vector that is closest to its averaged context representation. This vector is then used for the gradient update. In their dynamic version, Neelakantan et al. (2014) increase the number of sense vectors when the similarity of the current context representation with each of the existing sense vectors is less than a hyper-parameter $\lambda$. Li and Jurafsky (2015) build on (Neelakantan et al., 2014) in a more principled manner, using a Chinese Restaurant Process (CRP) to decide if the current appearance of a word is an old sense or a new one (sit on an existing "table" or a new "table" in CRP).

**Our Approach.** Our approach belongs to the joint learning category. Specifically, we learn multi-sense representations using a multilingual variant of the skip-gram model that dynamically adds new sense vectors for a word using Bayesian non-parametrics. The closest non-parametric approach to ours is that of Bartunov et al. (2016), who proposed a multi-sense variant of the skip-gram model that learns the different number of sense vectors for all words from a large monolingual corpus (in their case, the English Wikipedia). Our work can be viewed as the multi-view extension of their model, leveraging both monolingual and cross-lingual distributional signals as the views of the data. In our experiments, we compare our model to monolingually trained version of their model.

### *5.2.3. Motivation for Our Work*

As stated in the introduction, the motivation behind incorporating cross-lingual translational information to aid in learning sense representations was that *different senses of the same word may be translated into different words in another language*. Many works used this fact by exploiting bilingual parallel corpora to derive the signal (Bansal et al., 2012; Guo et al., 2014; Kawakami and Dyer, 2015; Šuster et al., 2016).

However, bilingual translational signals often do not suffice. It is possible that polysemy for a word survives translation. Figure 8 shows an illustrative example — both senses of *interest* get translated to *intérêt* in French. However, this becomes much less likely as the number of languages under consideration grows. For instance, by considering the Chinese translation in Figure 8, it can be seen that these senses translate to different surface forms in Chinese. Note that the opposite can also happen (i.e., same surface forms in Chinese, but different in French). Inspired by this, this chapter proposes a model that can use multiple languages to indirectly identify the sense of a word and learn multi-sense representations.



| | | |
|---|---|---|
| I got high **[interest]** on my savings from the bank. | Je suis un grand **[intérêt]** sur mes économies de la banque. | 我银行的存款有高**[利息]**。 |
| My **[interest]** lies in History. | Mon **[intérêt]** réside dans l'Histoire. | 我的**[兴趣]**在于历史。 |

Figure 8: **Benefit of Multilingual Information (beyond bilingual)**: Two different senses of the word *interest* and their translations to French and Chinese (word translation shown in [**bold**]). While the surface form of both senses of *interest* are same in French, they are different in Chinese, illustrating the benefit of having more than two languages.

## 5.3. Model Description

Let us define some notation before describing the model. Let $E = \{x^e_1, .., x^e_i, .., x^e_{N_e}\}$ denote the words of the English side and $F = \{x^f_1, .., x^f_i, .., x^f_{N_f}\}$ denote the words of the non-English side of the parallel corpus. We assume that we have word alignments $A_{e \to f}$ and $A_{f \to e}$ mapping words in English sentence to their translation in non-English sentence (and vice-versa), so that $x^e$ and $x^f$ are aligned if $A_{e \to f}(x^e) = x^f$.

We define NEIGHBORS$(x, L, d)$ as the neighborhood in language $L$ of size $d$ (on either side) around word $x$ in its sentence. The English and non-English neighboring words are denoted by $y^e$ and $y^f$, respectively. Note that $y^e$ and $y^f$ need not be translations of each other. Each word $x^f$ in the non-English vocabulary is associated with a dense vector $\boldsymbol{x}^f$ in $\mathbb{R}^m$, and each word $x^e$ in English vocabulary admits at most $T$ sense vectors, with the $k^{th}$ sense vector denoted as $\boldsymbol{x}^e_k$. A context vector is maintained for each word in the English and non-English vocabularies. The context vector is used as the representation of the word when it appears as the context for another word. As our main goal is to model multiple senses for words in English, we do not model polysemy in the non-English language and use a single vector to represent each word in the non-English vocabulary.

The joint conditional distribution of the context words $y^e, y^f$ given an English word $x^e$ and its corresponding translation $x^f$ on the parallel corpus is defined as,

$$\Pr(y^e, y^f \mid x^e, x^f; \alpha, \theta), \tag{5.1}$$

where $\theta$ are model parameters, i.e., all sense and context representations, and hyper-parameter $\alpha$ governs the prior on latent senses described below.

By indexing the senses of $x^e$ by the random variable $z$, Equation (5.1) can be rewritten as,

$$\int_\beta \sum\nolimits_z \Pr(y^e, y^f z, \beta \mid x^e, x^f, \alpha; \theta) d\beta$$

where $\beta$ are the parameters determining the model probability on each sense for $x^e$ (i.e., the weight on each possible value for $z$). We place a Dirichlet process (Ferguson, 1973) prior on sense assignment for each word. Thus, adding the word-$x$ subscript to emphasize that these are word-specific senses,

$$\Pr(z_x = k \mid \beta_x) = \beta_{xk} \prod\nolimits_{r=1}^{k-1} (1 - \beta_{xr}) \tag{5.2}$$

$$\beta_{xk} \mid \alpha \overset{ind}{\sim} Beta(\beta_{xk} \mid 1, \alpha), \quad k = 1, \ldots. \tag{5.3}$$

That is, each of the potentially infinite senses for word $x$ have their probability determined by a sequence of independent *stick-breaking weights*, $\beta_{xk}$, from the constructive definition of the Dirichlet Process (Sethuraman, 1994). The hyper-parameter $\alpha$ (or prior concentration) provides information on the number of senses we expect to observe in our corpus.

After conditioning upon word sense, the context probability decomposes as,

$$\Pr(y^e, y^f \mid z, x^e, x^f; \theta) = \Pr(y^e \mid x^e, x^f, z; \theta) \cdot \Pr(y^f \mid x^e, x^f, z; \theta). \tag{5.4}$$

Both the first and the second terms are sense-dependent, and each factors as,

$$\Pr(y \,|\, x^e, x^f, z\!=\!k; \theta) \propto \Psi(x^e, z\!=\!k, y) \cdot \Psi(x^f, y) \tag{5.5}$$

$$\text{where } \Psi(x^e, z\!=\!k, y) = \exp(\boldsymbol{y}^\intercal \boldsymbol{x}_k^e) \quad \text{and} \quad \Psi(x^f, y) = \exp(\boldsymbol{y}^\intercal \boldsymbol{x}^f) \tag{5.6}$$

where $\boldsymbol{x}_k^e$ is the representation corresponding to the $k^{th}$ sense of the word $x^e$, and $\boldsymbol{y}$ is the representation of either $y^e$ or $y^f$. The factor $\Psi(x^e, z = k, y)$ use the corresponding sense vector in a skip-gram-like formulation. This results in total of 4 factors,

$$\Pr(y^e, y^f \mid z, x^e, x^f; \theta) \propto \Psi(x^e, z, y^e) \cdot \Psi(x^f, y^f) \cdot \Psi(x^e, z, y^f) \cdot \Psi(x^f, y^e) \tag{5.7}$$

Figure 9 illustrates each factor. This approach is reminiscent of the BiSkip model of Luong et al. (2015b) from Section 3.4.1. BiSkip jointly learnt representations for two languages $l_1$ and $l_2$ by optimizing an objective containing 4 skip-gram terms for the aligned pair ($x^e$,$x^f$) — two predicting monolingual contexts $l_1 \rightarrow l_1$, $l_2 \rightarrow l_2$ , and two predicting cross-lingual contexts $l_1 \rightarrow l_2$, $l_2 \rightarrow l_1$.

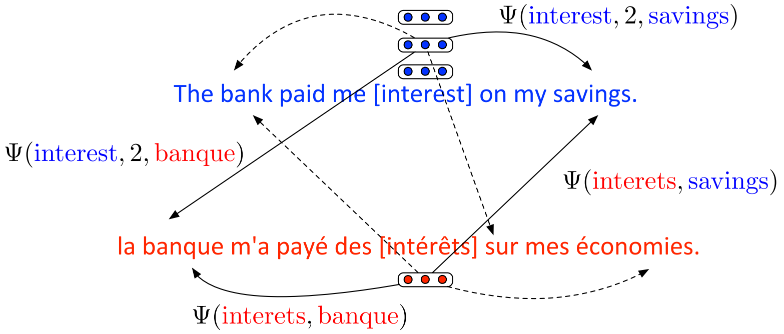

Ψ(interest, 2, savings)
The bank paid me [interest] on my savings.
Ψ(interest, 2, banque)
Ψ(interets, savings)
la banque m'a payé des [intérêts] sur mes économies.
Ψ(interets, banque)

Figure 9: The aligned pair (*interest*,*intérêt*) is used to predict monolingual and cross-lingual context in both languages (see factors in Equation (5.7)). Each sense vector (here 2nd is shown) for *interest*, participates in the update. Only the polysemy in English is modelled.

### 5.4. Learning and Disambiguation

**Learning.** Learning involves maximizing the log-likelihood,

$$\Pr(y^e, y^f \mid x^e, x^f; \alpha, \theta) = \int_\beta \sum_z \Pr(y^e, y^f, z, \beta \mid x^e, x^f, \alpha; \theta) d\beta \tag{5.8}$$

for which a variational approximation is used. Let $q(z, \beta) = q(z) \cdot q(\beta)$ where

$$q(z) = \prod_i q(z_i) \qquad q(\beta) = \prod_{w=1}^{V} \prod_{k=1}^{T} \beta_{wk} \tag{5.9}$$

are the fully factorized variational approximation of the true posterior in Equation 5.8, where $V$ is the size of English vocabulary, and $T$ is the maximum number of senses for any word. The optimization problem solves for $\theta$, $q(z)$ and $q(\beta)$ using the stochastic variational inference technique (Hoffman et al., 2013) similar to Bartunov et al. (2016).

The resulting learning algorithm is shown as Algorithm 2. The first for-loop (line 1) updates the English sense vectors using the cross-lingual and monolingual contexts. First, the expected sense distribution for the current English word $w$ is computed using the current estimate of $q(\beta)$ (line 4). The sense distribution is updated (line 8) using the combined monolingual and cross-lingual contexts (line 6) and re-normalized (line 10). Using the updated sense distribution $q(\beta)$'s sufficient statistics is re-computed (line 11) and the global

**Algorithm 2** Pseudocode of Learning Algorithm

```
Input:
    Parallel corpus $E = \{x^e_1, .., x^e_i, .., x^e_{N_e}\}$ and $F = \{x^f_1, .., x^f_i, .., x^f_{N_f}\}$
    Word alignments $A_{e \to f}$ and $A_{f \to e}$
Hyper-parameters:
    Prior concentration $\alpha$ and maximum number of allowed senses $T$, window sizes $d, d'$
Output: $\theta$, $q(\beta)$, $q(z)$
 1: for $i = 1$ to $N_e$ do                                  ▷ Update English vectors.
 2:     $w \leftarrow x^e_i$
 3:     for $k = 1$ to $T$ do
 4:         $z_{ik} \leftarrow \mathbb{E}_{q(\beta_w)}[\log \Pr(z_i = k|, x^e_i)]$
 5:     end for
 6:     $y_c \leftarrow$ NEIGHBORS$(x^e_i, E, d) \cup$ NEIGHBORS$(x^f_i, F, d') \cup \{x^f_i\}$ where $x^f_i = A_{e \to f}(x^e_i)$
 7:     for $y$ in $y_c$ do
 8:         SENSE-UPDATE$(x^e_i, y, z_i)$
 9:     end for
10:     Re-normalize $z_i$ using softmax
11:     Update sufficient statistics for $q(\beta)$ like Bartunov et al. (2016)
12:     Update $\theta$ using Equation (5.10)
13: end for
14: for $i = 1$ to $N_f$ do                         ▷ Jointly update non-English vectors.
15:     $y_c \leftarrow$ NEIGHBORS$(x^f_i, F, d) \cup$ NEIGHBORS$(x^e_i, E, d') \cup \{x^e_i\}$ where $x^e_i = A_{f \to e}(x^f_i)$
16:     for y in $y_c$ do
17:         SKIP-GRAM-UPDATE$(x^f_i, y)$
18:     end for
19: end for
20: procedure SENSE-UPDATE$(x_i, y, z_i)$
21:     $z_{ik} \leftarrow z_{ik} + \log \Pr(y \mid x_i, k, \theta)$
22: end procedure
```

parameter $\theta$ is updated (line 12) as follows,

$$\theta \leftarrow \theta + \rho_t \nabla_\theta \sum_{k|z_{ik} > \epsilon} \sum_{y \in y_c} z_{ik} \log \Pr(y|x_i, k, \theta) \tag{5.10}$$

Note that in the above sum, a sense participates in an update only if its probability exceeds a threshold $\epsilon$ (= 0.001). The final model retains sense vectors whose sense probability exceeds the same threshold. The last for-loop (line 14) jointly optimizes the non-English representations using English context with the standard skip-gram updates.

**Disambiguation.** Similar to Bartunov et al. (2016), the sense of the word $x^e$ can be disambiguated given its monolingual context $y^e$ as follows,

$$\Pr(z \mid x^e, y^e) \propto \Pr(y^e \mid x^e, z; \theta) \sum_{\beta} \Pr(z \mid x^e, \beta) q(\beta) \tag{5.11}$$

Although the model trains representations using both monolingual and cross-lingual context, at test time only monolingual context is available. However it was seen that so long as the model is trained with multilingual context, it performs well at sense disambiguation on the test data. A similar observation was made by Šuster et al. (2016).

## 5.5. Multilingual Extension

As discussed in Section 5.2.3, bilingual distributional signal alone may not be sufficient as polysemy may survive translation in the second language. The model described above can be easily modified to incorporate distributional information from more than one language. For using languages $l_1$ and $l_2$ to learn multi-sense embeddings for English, we train on a concatenation of English–$l_1$ parallel corpus with an English–$l_2$ parallel corpus. This technique can easily be generalized to more than two non-English languages to learn representations using a large multilingual corpus.

**Value of $\Psi(y^e, x^f)$.** The factor modeling the dependence of the English context word $y^e$ on non-English word $x^f$ is crucial to performance when using multiple languages. Consider the case of using French and Spanish contexts to disambiguate the financial sense of the English word *bank*. In this case, the (financial) sense vector of *bank* will be used to predict vector of *banco* (Spanish context) and *banque* (French context). If vectors for *banco* and *banque* do not reside in the same space or are not "close", the model will incorrectly assume they are different contexts to introduce a new sense for *bank*. This is precisely why the bilingual models, like that of Šuster et al. (2016), cannot be extended to multilingual setting, as they pre-train the embeddings of second language before running the multi-sense embedding process. As a result of naive pre-training, the French and Spanish vectors of semantically similar pairs like (*banco*,*banque*) will lie in different spaces and need not be close. A similar reasoning applies to the model of Guo et al. (2014).

To avoid this, the vector for pairs like *banco* and *banque* should lie in the same space and close to each other and the sense vector for *bank*. The $\Psi(y^e, x^f)$ term attempts to ensure this by using the vector for *banco* and *banque* to predict the vector of *bank*. This way, the model brings the embedding space for Spanish and French closer by using English as a bridge language during joint training. A similar idea of using English as a bridging language was used in the models proposed in (Hermann and Blunsom, 2014b) and (Coulmance et al., 2015). Beside the benefit in the multilingual case, the $\Psi(y^e, x^f)$ term improves performance in the bilingual case as well, as it forces the English and second language embeddings to remain close in space.

To show the value of $\Psi(y^e, x^f)$ factor, a variant of Algorithm 2 is trained without the $\Psi(y^e, x^f)$ factor in our experiments, by only using monolingual neighborhood, which is denoted by $\textsc{Neighbors}(x_i^f, F)$ in line 15 of Algorithm 2. This variant is referred to as the One-Sided model, and the model in Algorithm 2 as the Full model.

## 5.6. Experimental Setup

We first describe the datasets and the preprocessing methods used to prepare them. We also describe the Word Sense Induction (WSI) task used in the experiments.

**Parallel Corpora.** The experiments use parallel corpora in English (en), French (fr), Spanish (es), Russian (ru) and Chinese (zh) (corpus statistics are in Table 13). The first 10M lines from the English–French Giga corpus (Callison-Burch et al., 2011) are used for en–fr. The FBIS parallel corpus (LDC2003E14) is used for en–zh. As the domain from which parallel corpus has been derived can affect the final result, it is necessary to control for domain in all parallel corpora when analyzing what choice of languages provide suitable disambiguation signal. To this end, we also used the en–fr, en–es, en–zh and en–ru sections of the MultiUN parallel corpus (Eisele and Chen, 2010). Word alignments were generated

| Corpus | Source | Lines (M) | English-Words (M) |
|---|---|---|---|
| English–French (en–fr) | EU proc. | ≈ 10 | 250 |
| English–Chinese (en–zh) | FBIS news | ≈ 9.5 | 286 |
| English–Spanish (en–es) | UN proc. | ≈ 10 | 270 |
| English–French (en–fr) | UN proc. | ≈ 10 | 260 |
| English–Chinese (en–zh) | UN proc. | ≈ 8 | 230 |
| English–Russian (en–ru) | UN proc. | ≈ 10 | 270 |

Table 13: Corpus statistics (in millions) of the parallel corpora used to train the multi-sense representations. Horizontal lines demarcate corpora from the same domain.

using `fast_align` tool (Dyer et al., 2013) in the symmetric intersection mode. Tokenization was performed using `cdec`[3] toolkit. Stanford Segmenter (Tseng et al., 2005) was used to preprocess the Chinese corpora.

**Word Sense Induction (WSI).** We evaluate our approach on the word sense induction task. In this task, the input consists of several sentences showing usages of the same word, and the required output is a clustering of all sentences that use the same sense of the given word (Nasiruddin, 2013). The predicted clustering is then compared against a reference gold clustering. Note that WSI is a harder task than Word Sense Disambiguation (WSD)(Navigli, 2009), as unlike WSD, this task does not involve any supervision or explicit human knowledge about senses of words. The disambiguation approach in Equation 5.11 is used to predict the sense given the target word and four context words.

To allow for fair comparison with earlier work, the same benchmark datasets as Bartunov et al. (2016) are used — Semeval-2007, 2010 and Wikipedia Word Sense Induction (WWSI). We report Adjusted Rand Index (ARI) (Hubert and Arabie, 1985) in the experiments, as ARI is a more strict and precise metric than F-score and V-measure. Details on ARI can be found in the Appendix A.1.

**Parameter Tuning.** Five context words on either side are used to update each English word-vectors in all the experiments. In the monolingual setting, all five words are English;

[3] `github.com/redpony/cdec`

in the multilingual settings, we used four neighboring English words plus the one foreign word aligned to the word being updated ($d = 4$, $d' = 0$ in Algorithm 2). The effect of varying $d'$, the context window size in the foreign sentence, is also analyzed.

The parameters $\alpha$ (prior concentration) and $T$ (maximum number of allowed senses per word) were tuned by maximizing the log-likelihood of a held out English text.[4] The parameters were chosen from the following values $\alpha = \{0.05, 0.1, .., 0.25\}$, $T = \{5, 10, .., 30\}$. All models were trained for 10 iteration with a decaying learning rate of 0.025, decayed to 0. Unless otherwise stated, all embeddings are 100 dimensional.

## 5.7. Experiments

The experiments in this section evaluate the benefit of leveraging bilingual and multilingual information during training. The experiments also analyze how the choice of language (i.e., using closer or farther languages) used in cross-lingual training affect the performance of the embeddings.

### *5.7.1. Word Sense Induction (WSI) Results*

The results for WSI are shown in Table 14. Recall that the One-Sided model is the variant of Algorithm 2 without the $\Psi(y^e, x^f)$ factor. Mono refers to the AdaGram model of Bartunov et al. (2016) trained on the English side of the parallel corpus. In all cases, the Mono model is outperformed by One-Sided and Full models, showing the benefit of using cross-lingual signal in training. Best performance is attained by the multilingual model English–(French,Chinese), showing value of multilingual signal. The value of $\Psi(y^e, x^f)$ term is also verified by the fact that the One-Sided model performs worse than the Full model.

The Full model can also be compared[5] to to the AdaGram model described in Bartunov et al. (2016). AdaGram achieved ARI scores of 0.069, 0.097 and 0.286 on the three datasets respectively after training 300 dimensional embeddings on English Wikipedia ($\approx$ 100M

[4] first 100k lines from the en–fr Europarl (Koehn, 2005)

[5] This comparison is unfair to the Full model, that uses a smaller corpus than AdaGram.

| Model \ Dataset | S-2007 | S-2010 | WWSI | avg. ARI |
|---|---|---|---|---|
| | English–French | | | |
| MONO | .044 | .064 | .112 | .073 |
| ONE-SIDED | .054 | .074 | **.116** | .081 |
| FULL | **.055** | **.086** | .105 | **.082** |
| | English–Chinese | | | |
| MONO | .054 | .074 | .073 | .067 |
| ONE-SIDED | **.059** | .084 | .078 | .074 |
| FULL | .055 | **.090** | **.079** | **.075** |
| | English–(French,Chinese) | | | |
| MONO | .056 | .086 | .103 | .082 |
| ONE-SIDED | **.067** | .085 | .113 | .088 |
| FULL | .065 | **.094** | **.120** | **.093** |

Table 14: Results on word sense induction (left four columns) in ARI and contextual word similarity (last column) in percent correlation. Language pairs are separated by horizontal lines. Best results in **bold**.

lines). Note that, as WWSI was derived from Wikipedia, training on Wikipedia gives AdaGram model an undue advantage, resulting in high ARI score on WWSI. In comparison, our model did not train on English Wikipedia, and uses 100 dimensional embeddings. Nevertheless, even in the unfair comparison, it noteworthy that on S-2007 and S-2010, the model achieves comparable performance (0.067 and 0.094) with multilingual training to a model trained on almost 5 times more data using higher (300) dimensional embeddings.

### *5.7.2. Effect of Language Family Distance*

Intuitively, the choice of language used in cross-lingual training can affect the results as some languages may provide better disambiguation signals than others. This experiment systematically evaluates the impact of choosing languages from a closer family (e.g., French, Spanish) or a farther family (e.g., Russian, Chinese) for cross-lingual training along with English. To control for domain, the MultiUN corpus is used for all languages.[6] To evaluate the effect of using closer languages, we pair English with French and Spanish. Similarly,

[6] Šuster et al. (2016) compared different languages but did not control for domain.

| **Dataset**<br>**Model** | S-2007 | | S-2010 | | WWSI | | Avg. ARI | |
|---|---|---|---|---|---|---|---|---|
| Lang. Setting → | en–fr,es | en–ru,zh | en–fr,es | en–fr,es | en–fr,es | en–ru,zh | en–fr,es | en–ru,zh |
| (1) Mono | .035 | .033 | .046 | .049 | .054 | .049 | .045 | .044 |
| (2) One-Sided | .044 | **.044** | .055 | .063 | .062 | .057 | .054 | .055 |
| (3) Full | **.046** | .040 | **.056** | **.070** | **.068** | **.069** | **.057** | **.059** |
| (3) - (1) | .011 | .007 | .010 | .021 | .014 | .020 | .012 | .015 |

Table 15: Effect (in ARI) of language family distance on WSI task. Best results for each column is shown in **bold**. The improvement from Mono to Full is shown as (3) - (1). Note that this is not comparable to results in Table 14, as a different training corpus is used to control for the domain.

English is paired with Russian and Chinese to evaluate the effect of using farther languages. The result for each of these combinations for all datasets is in Table 15.

From Table 15, it can be seen that using farther languages yield a slightly higher improvement on an average than using closer languages, suggesting that using languages from a farther family aids better disambiguation. These findings echo those of Resnik and Yarowsky (1999), who found that the tendency to lexicalize senses of an English word differently in a second language correlated with language distance.

To illustrate the effect of multilingual signals for representation learning, a qualitative analysis of learnt embeddings is also conducted in Appendix A.3.

### *5.7.3. Effect of Window Size*

Figure 10 shows the effect of increasing the cross-lingual window ($d'$) on the average ARI for the English–French and English–Chinese models. Increasing $d'$ improves the average score for English–Chinese model, while the opposite is true for the English-French model. This suggests that it might be beneficial to have a separate $d'$ per language. This also aligns with the earlier observation that different language families have different suitability (bigger cross-lingual context from a distant family helped) for optimal performance.

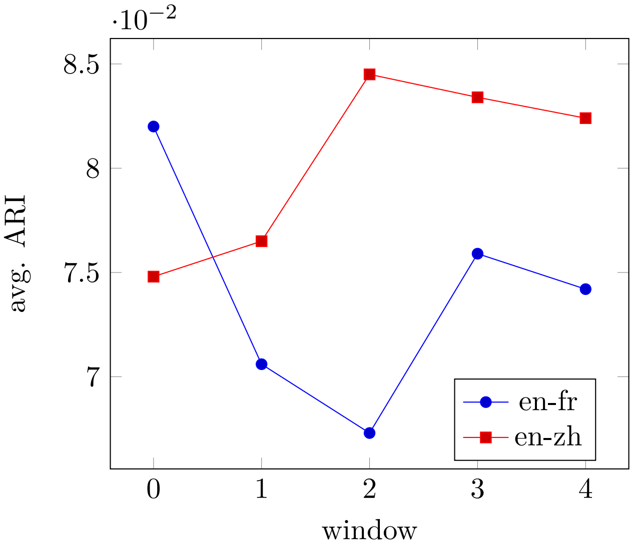


Figure 10: Tuning window size for English-Chinese and English-French models.

### *5.7.4. Qualitative Analysis of Induces Senses*

The fraction of polysemous words in the vocabulary is a function of the prior concentration parameter $\alpha$, the maximum number of allowed senses per word $T$, and the training paradigm (monolingual or multilingual). For different choices of $\alpha$ and $T$, about 10-20% polysemous words in the vocabulary were identified using monolingual training, while the same number was about 20-25% when using multilingual training. On closer examination, it was found that often the senses for a known polysemous word are over-segmented. For instance, consider the senses detected for the word *apple* in Table 16. On examining the nearest neighbors of senses $apple_1$ and $apple_3$, it is evident that these actually represent the same sense of the word (i.e., *apple* the technology company). Ideally, the model should have assigned a single sense representation for this sense instead of two. A similar observation can be made for the senses discovered for *plant* and *bank*. These extraneous senses (also called *pseudo-senses*) can also appear for a word which is used in only one sense in the corpus (i.e., should ideally be monosemous). For instance, the three senses detected for the word *pope* are all related to concepts in the Catholic Church. These pseudo-senses are generated because often the same sense of a word may appear in sufficiently different

| senses | neigbours |
|---|---|
| apple$_1$ | netscape, vmware, nintendo, tivo, mandriva |
| apple$_2$ | peach, plum, pear, strawberry, banana |
| apple$_3$ | macintosh, smartphones, microsoft, pc, ibm |
| plant$_1$ | medicinal, legume, cultivated, insects, vascular |
| plant$_2$ | fern, tuber, shrub, fertilize, conifer |
| plant$_3$ | fungus, genus, flowering, rosaceae, asteracea |
| plant$_4$ | power, nuclear, reactor, fueled, reprocessing |
| plant$_5$ | manufacturing, fabrication, bottling, steelmaking, compressor |
| monitor$_1$ | observer, un, deploy, mission, peacekeeper |
| monitor$_2$ | propublica, watchdog, politbarometer, internews |
| monitor$_3$ | lcd, display, handheld, infrared, camera |
| bank$_1$ | hsbc, citibank, lender, insurance, macquarie |
| bank$_2$ | loan, deposit, depositor, thrift, asset |
| bank$_3$ | bethlehem, jericho, ramallah, west, kalkilya |
| bank$_4$ | edge, side, opposite, along, shore |
| pope$_1$ | vatican, benedict, sainthood, papal, pontiff, homily |
| pope$_2$ | pius, excommunicate, gregory, antipope, calixtus |
| pope$_3$ | papacy, beatify, pontificate, xxiii, beatification, frail |

Table 16: Senses discovered for some words under multilingual training, and their nearest neighbors in the vector space.

contexts that the model decides to allocate a new sense representation for the word. Note that the presence of pseudo-senses is not unique to our model, but a limitation of all data-driven approaches for sense discovery. Indeed, the over-segmentation problem also affects models which learn a fixed number of senses per word, as documented in (Shi et al., 2016). Shi et al. (2016) suggested a post-processing step, where such pseudo-senses are detected and eliminated, while preserving the spatial orientation with respect to rest of the words in the vocabulary. However, this approach only addresses this problem post-hoc, instead of preventing over-segmentation of senses during training itself, which is currently an active area of research.

## 5.8. Summary

In this chapter, I described a multi-sense representation learning approach that exploits translational information from two or more languages, in addition to monolingual distributional information, to dynamically learn multiple representations per word in English. The experiments showed that using the additional multilingual signal yielded comparable performance to a monolingual model that was trained on five times more data. Unlike other chapters, the aim of this chapter was to show that cross-lingual signals can also aid in learning better representations for a monolingual lexical semantic task like sense induction.

One limitation of the sense-specific representation learning approaches is that the sense vectors learnt are *static*. That is, once trained, the sense vector(s) for a word remains the same, regardless of the context in which it appears. The act of compiling a sense vector for a word divorces it from the contexts that defined (and warranted) its representation. This goes against one of the remarks made by Firth (1935) — *the complete meaning of a word is always contextual, and no study of meaning apart from a complete context can be taken seriously.* Indeed, this criticism also applies to single-sense word representations discussed in earlier chapters. Ideally, the representation of a word (or one of its senses) cannot be static, but should dynamically change as a function of its context. To truly captures this intuition, dynamic, context-sensitive word representations that generate the representation of the word *conditioned* on the context have been developed recently (Peters et al., 2018; Devlin et al., 2018). Nevertheless, enumerating possible meanings in sense inventories do hold practical value, evident from the success of resources like WordNet, FrameNet in NLP applications. Consequently, data-driven sense representation approaches like ours are still attractive as a tool to aid lexicographers estimate how many different senses a word assumes in a large corpus, or cluster appearances of a word that resemble the same sense, easing the burden of manual annotation.

# CHAPTER 6 : Identifying Hypernymy across Languages

## 6.1. Introduction

Previous chapters showed the effectiveness of cross-lingual representations for primarily coarse semantic tasks (like dictionary induction, document classification etc.), that required capturing *symmetric* relationships (e.g., translational similarity) between words in different languages. While translational similarity helps identify correspondences, identifying other *asymmetric* semantic relationships can improve language understanding where exact equivalence does not exist. One such relationship is *hypernymy* — a word $x$ is said to be the hypernym of a word $y$ (referred to as the hyponym) if $y$ is a kind of $x$ (e.g., *dog* is a hypernym of *pitbull*).

In this chapter, I will discuss how we can detect *asymmetric* relations like hypernymy across languages using cross-lingual word representations. That is, identifying that *écureuil* ("squirrel" in French) is a kind of *rodent*, or ագռավ ("crow" in Armenian) is a kind of *bird*. Before examining the cross-lingual hypernymy detection problem, it is prudent to re-visit previous work and approaches on the monolingual version and examine the challenges of the cross-lingual version.

## 6.2. History of the Hypernymy Detection Task

The hypernymy detection problem involves predicting whether a given $(x, y)$ pair is a hyponym-hypernym pair (that is, is $y$ the hypernym of $x$?) or not. For instance, detecting that the word pair $(dolphin, mammal)$ is a hyponym-hypernym pair while the pair $(dolphin, reptile)$ is not.

The ability to detect lexical relations like hypernymy has a wide range of applications — as a component in textual entailment systems (Dagan et al., 2005; Sammons et al., 2012; Dagan et al., 2013), building semantic ontologies (Riloff and Shepherd, 1997; Chklovski and Pantel, 2004; Snow et al., 2006; Kozareva and Hovy, 2010), a feature in coreference resolution (Ponzetto and Strube, 2006), developing question answering systems (Prager et al., 2001; Yih et al., 2013), or simply doing information extraction (Etzioni et al., 2005; Demeester et al., 2016).

### *6.2.1. Hypernymy Detection in English*

Hypernymy detection in English is a well studied problem, with three prominent threads in the literature — path-based approaches, distributional approaches, and hybrid approaches that combine the two. I discuss them briefly below.

**Path-based Approaches.** Path-based approaches identify lexico-syntactic patterns (or *paths*) between $x$ and $y$ in a sentence that indicate lexical relationships like hypernymy (Hearst, 1992; Snow et al., 2004), synonymy (Lin et al., 2003), or meronymy (Girju et al., 2006). Instances of such patterns indicative of hypernymy include "$x$ such as $y$" (as in "animals such as dogs"), "$x$ and other $y$" (as in "squirrel and other rodents") among others. Statistics over these patterns are computed for each candidate pair $(x, y)$ over a large corpus, and weights learnt for each pattern to determine hypernymy. As these patterns are quite specific, their presence is a precise and accurate indicator of hypernmy.

The key drawback of path-based approaches is that they requires co-occurrence of both $x$ and $y$ in the same sentence, *exactly* as in the predefined pattern. As a result, reliable statistics are often hard to obtain for word pairs that co-occur rarely, a problem that is exacerbated in limited size corpus. Consequently, while such path-based approaches have been effective in terms of precision, they leave much to be desired in recall.

**Distributional Approaches.** Distributional approaches (Lin, 1998a; Weeds and Weir, 2003; Lenci and Benotto, 2012) probe the distributional representations of $x$ and $y$ to determine if the hypernymy relation holds between them. The distributional representations $\boldsymbol{x}$ and $\boldsymbol{y}$ for $x$ and $y$ are probed using a *similarity measure* $Sim(x, y)$, that is either learnt or unsupervised. Different distributional approaches differ either in the choice of the representations for $x$ and $y$, or the unsupervised similarity measure $Sim$.

*Dependency-based* distributional representation (Lin, 1998a; Levy and Goldberg, 2014b) have emerged as a popular representation used in distributional approaches. On the other hand, a myriad of supervised and unsupervised measures are available to determine if an asymmetric relation like hypernymy holds between a $(x, y)$ pair. Supervised measures learn classifiers using distributional features for a $(x, y)$ pair, which can be any of the following: concatenation $\boldsymbol{x} \oplus \boldsymbol{y}$ (Baroni et al., 2012), difference $\boldsymbol{y} - \boldsymbol{x}$ (Fu et al., 2014), dot product $\boldsymbol{y}^{\intercal}\boldsymbol{x}$, or a combination of $f = \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|} - \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}$ and $f^2$ as in Roller et al. (2014). Such approaches suffer from two major drawbacks — firstly, there exist limited resources that can serve as supervision for the similarity measure, and secondly, many of these approaches are shown to be prone to lexical memorization (Levy et al., 2015).[1] In contrast, unsupervised measures probe the features in word embeddings to determine hypernymy. There exist a variety of similarity measures that can be used to quantify the directional relationship between two words (Lin, 1998a; Weeds and Weir, 2003; Lenci and Benotto, 2012). One such unsupervised similarity measure, named *Bal*anced *A*verage *P*recision *Inc*lusive (or *BalAPinc*), will be discussed in detail in Section 6.3.

**Hybrid Approaches.** Hybrid approaches (Mirkin et al., 2006; Kaji and Kitsuregawa, 2008) attempt to combine the path-based and distributional signals to identify lexical relationships. By combining a high-precision but low-recall path-based approach and a low-precision but high-recall distributional approach, hybrid approaches combine the benefits of both. The most recent work in this thread is that of Shwartz et al. (2016), who encoded the paths using a RNN into a dense vector, and combined it with the distributional vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. The network was supervised using hyponym-hypernym pairs and negative examples sourced from lexical resources like WordNet (Miller, 1995) and WikiData (Vrandečić, 2012).

### *6.2.2. Cross-lingual Hypernymy Detection*

The cross-lingual hypernymy detection problem involves determining if $(x, y)$, where $x$ and $y$ are in different languages, constitutes a hyponym-hypernym pair. For instance, determining

[1] That is, memorizing that words like *mammal* are hypernyms of $x$, regardless of $x$.

if (*écureuil*, *rodent*) constitutes a hyponym-hypernym pair, where *écureuil* is the French word for *squirrel*. The rest of the chapter assumes that $y$ is always a word in English.

The ability to detect hypernyms across languages would have benefits similar to monolingual hypernymy detection. For instance, it could serve as a building block in cross-lingual tasks, like cross-lingual textual entailment tasks (Negri et al., 2012, 2013), constructing multilingual taxonomies (Fu et al., 2014), event coreference in multilingual news sources (Vossen et al., 2015), evaluating Machine Translation via entailment (Pado et al., 2009) or detecting semantic divergence in parallel text (Carpuat et al., 2017).

At first glance, translating words to English and then identifying hypernyms in a monolingual setting may appear to be a sufficient solution. However, this approach cannot capture many phenomena. For instance, the English words *cook*, *leader*, *supervisor* can all be hypernyms of the French word *chef*, as it does not have an exact translation in English covering its possible usages in French. But translating *chef* to *cook* precludes identifying *leader* or *supervisor* as a hypernym. Similarly, language-specific usage patterns can also influence hypernymy decisions. For instance, the French word *chroniqueur* translates to *chronicler* in English, but is more frequently used in French to refer to journalists (making *journalist* its hypernym). This motivates approaches that extend distributional methods for detecting monolingual hypernymy to cross-lingual settings.

Building models that can robustly identify hypernymy across languages is a challenging problem. Firstly, in the cross-lingual setting is that there is no direct approach to incorporate path-based features, because $x$ and $y$ belong to different languages, and are unlikely to appear together in the same sentence. This precludes path-based or hybrid approaches that have been used for monolingual settings. As a result, purely distributional approaches seem to be the only option for detecting cross-lingual hypernymy relationships. However, state-of-the-art distributional approaches for detecting monolingual hypernymy require syntactic analysis (Roller and Erk, 2016; Shwartz et al., 2017) (e.g., dependency parsing) to learn word representations, which may not available for many languages.

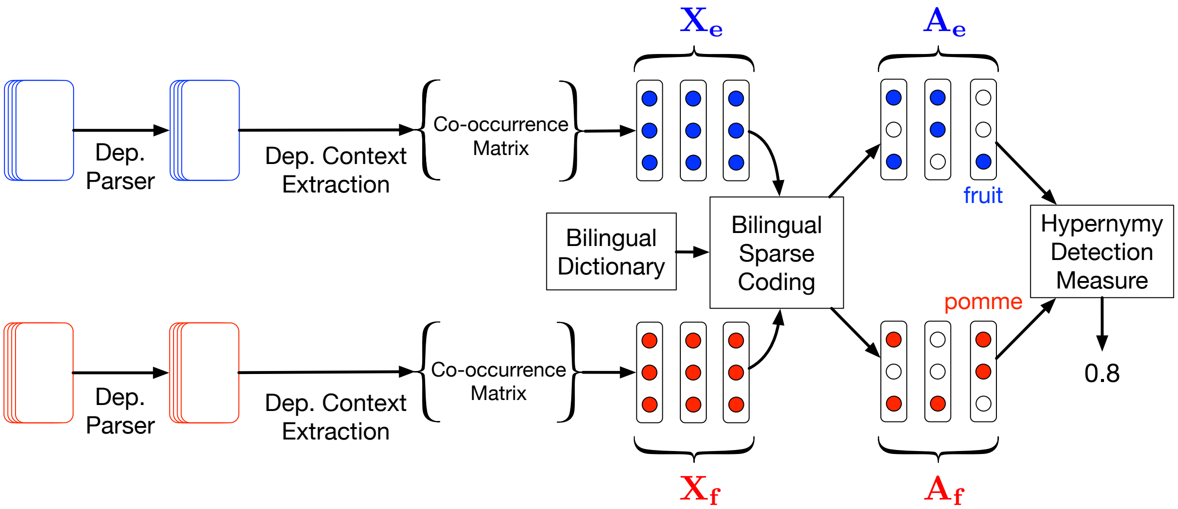


Figure 11: The BiSparse-Dep Approach, that learns sparse bilingual embeddings using dependency-based contexts. The resulting sparse embeddings, together with an unsupervised hypernymy detection measure, can detect hypernyms across languages (e.g., *pomme* is a *fruit*).

### 6.3. The BiSparse-Dep Approach

This section proposes BiSparse-Dep, a family of approaches that uses bilingual, dependency based word embeddings to detect hypernymy between words in different languages. An overview of BiSparse-Dep is in Figure 11. BiSparse-Dep has two key components: **(a)** *Dependency-based word representations*, that enable generalization across different languages with minimal customization by abstracting away language-specific word order. **(b)** *Bilingual sparse coding*, that allow us to align dependency-based word representation in a shared semantic space using a small bilingual dictionary. The resulting sparse bilingual embeddings can then be used with an *unsupervised hypernymy detection measure* (Section 6.3) to determine hypernymy between word pairs.

**Dependency-based Word Representations**

As discussed in Chapter 2, the context of a word can be described in multiple ways when learning distributional representations. One such context is defined using the syntactic neighborhood of the word in a *dependency graph*. For instance, for the sentence in Figure 12,

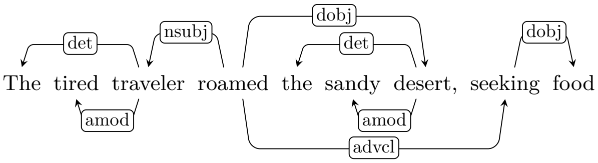


Figure 12: Dependency tree for *"The tired traveler roamed the sandy desert, seeking food"*.

the syntactic neighborhood for the target word *traveler* can be described in the following two ways:

- Full context (Padó and Lapata, 2007; Baroni and Lenci, 2010; Levy and Goldberg, 2014b): Children and parent words, concatenated with the label and direction of the relation (e.g., *roamed#nsubj*$^{-1}$ and *tired#amod* are contexts for *traveler*).

- Joint context (Chersoni et al., 2016): Parent concatenated with each of its siblings (e.g., *roamed#desert* and *roamed#seeking* are contexts for *traveler*).

Both context types encode directionality into the context, either through label direction or through sibling-parent relations. The two contexts exploit different amounts of syntactic information — Joint does not require labeled parses unlike Full. Joint context combines parent and sibling information, while Full keeps them as distinct contexts.

Word representations learnt using the syntactic neighborhood of a word (such as the ones described above) are popularly called *dependency-based word representations*. Dependency context based word representations capture functional similarity (e.g., *singing* and *rapping*), in contrast to topical similarity (e.g., *singing* and *dancing*) as captured by lexical context (Levy and Goldberg, 2014b). Such dependency based embeddings have been shown to outperform window based embeddings on many tasks (Bansal et al., 2014; Hill et al., 2014; Melamud et al., 2016). In fact, for the monolingual hypernmy detection task, it has been shown that dependency embeddings can recover Hearst patterns (Roller and Erk, 2016), and are almost always superior to window based embeddings (Shwartz et al., 2017).

### Dependency Contexts without a Treebank

Using dependency contexts in multilingual settings may not always be possible, as large dependency-parsed corpora are hard to obtain. One can parse a raw corpus in the language of interest using a dependency parser, but pre-trained dependency parsers are available for a handful of languages. Training a parser is also not feasible, as dependency treebanks used to supervise these parsers are not available for many languages. To circumvent these issues, a weak dependency parser is trained on languages related to the language of interest.

Specifically, a *delexicalized* parser is trained using treebanks of related languages, where the word form features are turned off, so that the parser is trained on purely non-lexical features (e.g., POS tags). The rationale behind this is that related languages show common syntactic structure that can be transferred to another language via delexicalized parsing (Zeman and Resnik, 2008; McDonald et al., 2011, inter alia).

### Unsupervised Hypernymy Detection Measure

When can one assert that word $y$ is a hypernym of a word $x$ using their distributional representations? To answer this question a hypothesis was formalized by Weeds et al. (2004) and Geffet and Dagan (2005),

**The Distributional Inclusion Hypothesis**

The distributional inclusion hypothesis (DIH) states that if a word $y$ is a hypernym of a word $x$, then the set of distributional features of $x$ are included in the set of distributional features of $y$ (Weeds et al., 2004; Geffet and Dagan, 2005). That is, the contexts in which $x$ occurs are a subset of those in which $y$ occurs.

DIH was introduced in (Weeds et al., 2004) as *distributional generality* for hypernymy detection, and was stated more generally for determining lexical entailment in (Geffet and Dagan, 2005). Intuitively, DIH states that a hypernym $y$ can replace appearances of its hyponym $x$. For example, *rodent* can replace *squirrel* in the sentence “the *squirrel* is hibernating for the winter". Notice that the reverse is not true — *squirrel* cannot replace *rodent* in a sentence like "The capybara is the largest living rodent". To apply DIH to detect hypernymy, one needs to quantify the amount of overlap in the co-occurrences of $x$ and $y$ with other contexts.

**Known Limitations.** The intuition behind DIH is not always correct. For instance, Kartsaklis and Sadrzadeh (2016) noted that in sentences with quantifiers (e.g., "all", "none"), replacing word with its hypernym is not always appropriate. For instance, changing "all *squirrels* hibernate" to "all *animals* hibernate". In such contexts, DIH fails. Similarly, Rimell (2014) noted that DIH is not correct in contexts that are collocational (e.g., "hot dog" to "hot animal"), highly specific ("squirrels eat nuts" to "animals eat nuts"), or when the hypernym being considered is too general (e.g., *entity* and *squirrel*).

Despite these limitations, DIH has enjoyed success in many lexical entailment applications, and several unsupervised hypernymy detection measures have been developed that appeal to it (Weeds and Weir, 2003; Weeds et al., 2004; Clarke, 2009; Lenci and Benotto, 2012).

**The *BalAPinc* Measure.** In this work, we use one such measure, named *BalAPinc*, first described by Kotlerman et al. (2009), to score word pairs for hypernymy. *BalAPinc* is defined using two other measures: *LIN* (Lin, 1998b) and *APinc* (Kotlerman et al., 2009).

The *LIN* measure defines a symmetric similarity score for a word pair ($x$,$y$),

$$LIN(x,y) = \frac{\sum_{f \in \boldsymbol{x} \cap \boldsymbol{y}} \boldsymbol{x}[f] + \boldsymbol{y}[f]}{\sum_{f \in \boldsymbol{x}} \boldsymbol{x}[f] + \sum_{f \in \boldsymbol{y}} \boldsymbol{y}[f]} \tag{6.1}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are representations of $x$ and $y$ respectively, $f \in \boldsymbol{x}$ and $f \in \boldsymbol{y}$ are feature indices active in $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively, and $f \in \boldsymbol{x} \cap \boldsymbol{y}$ are features indices that are active (non-zero values) in both $\boldsymbol{x}$ and $\boldsymbol{y}$. The value of a feature at index $f$ is $\boldsymbol{x}[f]$. The *LIN* measure computes the amount of information needed to describe the commonality of $x$ and $y$, relative to the information needed to fully describe $x$ and $y$.

On the other hand, *APinc* is an asymmetric score measuring a relevance-weighted overlap in context co-occurrences of $x$ and $y$, computed using modified version of *average precision*[2]

$$APinc(x \rightarrow y) = \frac{\sum_r P(r) \cdot rel(f)}{|\boldsymbol{x}|} \quad \text{where} \quad rel(f) = \begin{cases} 1 - \frac{rank(f,\boldsymbol{y})}{|\boldsymbol{y}|+1} & f \in \boldsymbol{y} \\ 0 & otherwise \end{cases} \tag{6.2}$$

here $rel(f)$ is a measure of feature $f$'s relevance, $rank(f, \boldsymbol{y})$ is rank of feature $f$ (among all active features) in the representation of $y$, and $P(r)$ is the precision at rank $r$ for the features $y$ with respect to features of $x$ (i.e., ratio of the number of included features of $x$ in $y$ from rank 1 to $r$, and $r$). *APinc* computes the proportion of the included (active) features of $y$, weighted by their relevance score $rel(f)$, with respect to all active features of $x$. *BalAPinc* is defined as the geometric mean of these measures,

$$BalAPinc(x \rightarrow y) = \sqrt{LIN(x, y) \cdot APinc(x \rightarrow y)} \tag{6.3}$$

### Bilingual Sparse Coding

To compare the features of words $x$ and $y$ belonging to different languages, one has to first align the independently learnt dependency representations. Moreover, the *BalAPinc* metric described above requires word representations that are sparse, so that active features for computing the score can be identified. To achieve this, we generate BiSparse-Dep embeddings using the BiSparse framework from Vyas and Carpuat (2016). BiSparse generates sparse, bilingual word embeddings using a dictionary learning objective with a sparsity inducing $l_1$ penalty. Appendix A.2 describes the learning objective in detail.

BiSparse takes as input pre-computed monolingual embeddings $\mathbf{X_e}$, $\mathbf{X_f}$ for two languages along with a translation matrix $\mathbf{S}$, and outputs sparse matrices $\mathbf{A_e}$ and $\mathbf{A_f}$ that are bilingual representations in a shared semantic space. The translation matrix $\mathbf{S}$ (of size $v_e \times v_f$)

[2] A metric used for ranking evaluation.

captures correspondences between the vocabularies (of size $v_e$ and $v_f$) of two languages. For instance, each row of $\mathbf{S}$ can be a one-hot vector that identifies the word in $v_f$ that is most frequently aligned with the word in $v_e$ for that row in a large parallel corpus, thus building a one-to-many mapping between the two languages.

## 6.4. Crowd-Sourcing Annotations

There is no publicly available dataset to evaluate models of hypernymy detection across multiple languages. While ontologies like Open Multilingual WordNet (OMW) (Bond and Foster, 2013) and BabelNet (Navigli and Ponzetto, 2012) contain cross-lingual links, these resources are semi-automatically generated and hence contain noisy edges. To make consistent claims about a hypernymy detection model, it is crucial to carefully design a dataset (Carmona and Riedel, 2017). To get reliable and high-quality test beds, evaluation datasets were collected using CrowdFlower.[3] The datasets span four languages from distinct families — French (fr), Russian (ru), Arabic (ar) and Chinese (zh) — paired with English.

**Annotation Setup for Cross-lingual Hypernymy.** To begin the annotation process, candidate pairs are first pooled using hypernymy edges across languages from OMW and BabelNet, along with translations from monolingual hypernymy datasets (Kotlerman et al., 2010; Baroni and Lenci, 2011; Baroni et al., 2012).

The annotation task requires annotators to be fluent in both English and the non-English language. To ensure only fluent speakers perform the task, task instructions are provided in the non-English language itself, and the task is restricted to annotators verified by CrowdFlower to have the respective language skills. Annotators also need to pass a quiz based on a small amount of gold standard data to gain access to the task.

Annotators choose between three options for each word pair $(p_f, q_e)$, where $p_f$ is a non-English word and $q_e$ is an English word: "$p_f$ is a kind of $q_e$", "$q_e$ is a part of $p_f$" and "none of the above". Word pairs labeled with the first option are considered as positive examples

[3] A crowd-sourcing platform, `www.figure-eight.com`.

while those labeled as "none of the above" are considered as negative.[4] The second option was included to filter out meronymy examples that were part of the noisy pool. It is left to the annotator to infer whether the relation holds between any senses of $p_f$ or $q_e$, if either of them are polysemous.

For every candidate hypernym pair $(p_f, q_e)$, annotators are also asked to judge its reversed and translated *hyponym* pair $(q_f, p_e)$. That is, if $(citron, food)$ is a hypernym candidate, annotators are also shown $(aliments, lemon)$ that is a potential hyponym candidate (*potential*, because as mentioned in Section 6.2.2, translation need not preserve semantic relationships). The purpose of presenting the hyponym pair, $(q_f, p_e)$, is two-fold. First, it emphasizes the directional nature of the task to the annotators. Second, it also identifies hyponym pairs, which we use as negative examples. The hyponym pairs are challenging since differentiating them from hypernyms truly requires detecting asymmetry.

Each pair was judged by at least 5 annotators, and judgments with 80% agreement (at least 4 annotators agree) are considered for the final dataset. This is a stricter condition than certain monolingual hypernymy datasets - for instance, EVALution (Santus et al., 2015) - where agreement by 3 annotators is deemed enough. Inter-annotator agreement measured using Fleiss' Kappa (Fleiss, 1971) was 58.1 (French), 53.7 (Russian), 53.2 (Arabic) and 55.8 (Chinese). This indicates moderate agreement, at par with agreement obtained on related fine-grained semantic tasks (Pavlick et al., 2015). Comparison with monolingual hypernymy annotator agreement is not possible as, to the best of our knowledge, such numbers are not available for existing test sets. Dataset statistics are shown in Table 17.

It was observed that annotators were able to agree on pairs containing polysemous words where hypernymy holds for some sense. For instance, for the French-English pair (*avocat*, *professional*), the French word *avocat* can either mean *lawyer* or *avocado*, but the pair was annotated as a positive example.

[4] The extra negative pairs were sub-sampled so as to keep a balanced dataset for evaluation.

| Lang. Pair | #crowd-sourced | #pos (= #neg) |
|---|---|---|
| French–English | 2115 | 763 |
| Russian–English | 2264 | 706 |
| Arabic–English | 2144 | 691 |
| Chinese–English | 2165 | 806 |

Table 17: Crowd-sourced dataset statistics. #pos (#neg) denote the number of positives (negatives) in the evaluation set, and #crowd-sourced denote the number of crowd-sourced pairs. Negatives were deliberately under-sampled to have a balanced evaluation set. Some examples for positive and negative instances in French and Russian are: (pêcheur (en:fisher), worker) (vêtement (en:clothing), jeans) (замок (en:castle), structure) (флора (en:flora), sunflower)

**Two Evaluation Sets.** To verify if the crowd-sourced hyponyms are challenging negative examples we create two evaluation sets. Both share the (crowd-sourced) positive examples, but differ in the nature of the negative examples:

- Hyper-Hypo — negative examples are the crowd-sourced hyponyms.
- Hyper-Cohypo — negative examples are *cohyponyms* drawn from OMW.

Cohyponyms are words sharing a common hypernym. For instance, *bière* (“beer” in French) and *vodka* are cohyponyms since they share a common hypernym in *alcool*/*alcohol*. Cohyponyms were chosen for the second test set for two reasons. First, to perform well on the test set requires differentiating between a symmetric (word similarity) and an asymmetric relation (hypernymy). For instance, *bière* and *vodka* are similar words, yet, they do not have a hypernymy relationship. Second, cohyponyms are a popular choice of negative examples in other monolingual entailment datasets (Baroni and Lenci, 2011).

## 6.5. Experimental Setup

Training BiSparse-Dep requires a dependency parsed monolingual corpus, and a translation matrix for jointly aligning the monolingual vectors. The translation matrix is computed using word alignments derived from parallel corpora. The corpus statistics for both monolingual and parallel corpora are in Table 18. While we use parallel corpora to generate the translation matrix to be comparable to baselines (Section 6.5), we do not *require* it — the matrix can be obtained from any bilingual dictionary.

**Computing Dependency Co-occurrences.** The monolingual corpora are parsed using `Yara Parser` (Rasooli and Tetreault, 2015), trained on the corresponding treebank from version 1.4 of the Universal Dependency Treebank (McDonald et al., 2013). `Yara Parser` was chosen as it is fast, and competitive with state-of-the-art parsers (Choi et al., 2015). The monolingual corpora was POS-tagged using TurboTagger (Martins et al., 2013). Dependency contexts for words are induced by first thresholding the language vocabulary to the top 50,000 nouns, verbs and adjectives. A co-occurrence matrix is then computed over this vocabulary using the context types in Section 6.3.

**Inducing BiSparse-Dep Embeddings.** The entries of the word-context co-occurrence matrix are re-weighted using PPMI (Bullinaria and Levy, 2007) from Section 2.6. The resulting matrix is reduced to 1000 dimensions using SVD (Golub and Kahan, 1965). The embedding dimensionality was chosen based on preliminary experiments with {500, 1000, 2000, 3000} dimensional vectors for English–French. The output matrices are used as $\mathbf{X_e}, \mathbf{X_f}$ in the setup from Section 6.3 to generate 100 dimensional sparse bilingual embeddings.

**Evaluation.** Accuracy is used as the evaluation metric in all experiments, as it is easy to interpret when the classes are balanced (Turney and Mohammad, 2015). Both evaluation datasets — Hyper-Hypo and Hyper-Cohypo — are split into a development and a test set in the ratio 1:2 respectively.

**Tuning.** *BalAPinc* has two tunable parameters — 1) a threshold that indicates the *BalAPinc* score above which all examples are labeled as positive, 2) the maximum number of features to consider for each word. The tuning set is used to tune the two parameters as well as the various hyper-parameters associated with the models.

| Lang. | Parallel Data | #sent. | Monolingual Data | #sent. |
|---|---|---|---|---|
| English | – | – | Wackypedia (Baroni et al., 2009) | 43M |
| French | Europarl (Koehn, 2005)<br>NewsCommentary,<br>Wikipedia (Tiedemann, 2012) | 2.7M | Wikipedia | 20M |
| Russian | Yandex-1M | 1.6M | Wikipedia | 22M |
| Arabic | ISI (Munteanu and Marcu, 2007)<br>NewsCommentary,<br>Wikipedia (Tiedemann, 2012) | 1.1M | Gigaword 3.0 (Graff, 2007) | 17M |
| Chinese | FBIS (LDC2003E14) | 9.5M | Gigaword 5.0 (Parker, 2011) | 58M |

Table 18: Training data statistics for different languages. While we use parallel corpora for computing translation dictionaries, our approach can work with any bilingual dictionary.

### Contrastive Approaches

The BiSparse-Dep approach is compared with the following approaches:

(a) Mono-Dep **(Translation Baseline)** For a word pair ($p_f$, $q_e$) in the test data, $p_f$ is translated to English using the most common translation in the translation matrix. Entailment is then determined using sparse, dependency based embeddings in English.

(b) BiSparse-Lex **(Window-Based)** Predecessor of the BiSparse-Dep model from Vyas and Carpuat (2016). This model induces sparse, cross-lingual embeddings using window based context using the bilingual sparse coding objective.

(c) Bivec+ **(Window-Based)** Extension of the Bivec model of Luong et al. (2015b). Bivec generates dense, cross-lingual embeddings using window based context by substituting aligned word pairs within a window in parallel sentences. By default, Bivec only trains using parallel data, so to ensure fair comparison it is initialized with monolingually trained window based embeddings .

(d) Cl-Dep **(Dependency-Based)** The model from Vulić (2017), that induces dense, dependency-based cross-lingual embeddings using the `word2vecf`[5] toolkit, by translating syntactic word-context pairs using the most common translation.

### Evaluating Robustness of BiSparse-Dep

This set of experiments investigates how robust BiSparse-Dep is when exposed to data scarce settings. Evaluating on a truly low-resource language is complicated by the fact that obtaining an evaluation dataset for such a language is difficult. Therefore, such settings are simulated for the languages in our dataset in multiple ways.

**No Treebank.** If a treebank is not available in a language, dependency-contexts have to be induced using treebanks from other languages (Section 6.3), which can affect the quality of the dependency-based embeddings. To simulate this, a delexicalized parser is trained for each of the four languages. We use treebanks from Slovenian, Ukrainian, Serbian, Polish, Bulgarian, Slovak and Czech (40k sentences) for training the Russian parser, and treebanks from English, Spanish, German, Portuguese, Swedish and Italian (66k sentences) for training the French parser. UDT does not (yet) have languages in the same family as Arabic or Chinese, so for the sake of completeness, Arabic and Chinese delexicalized parsers are trained on treebanks of the language itself. After delexicalized training, the Labeled Attachment Score (LAS) on the UDT test set dropped by several points for all languages — from 76.6% to 60.0% for Russian, 83.7% to 71.1% for French, from 76.3% to 62.4% for Arabic and from 80.3% to 53.3% for Chinese. The monolingual corpora are then parsed with these weaker parsers, and dependency context co-occurrences are computed as before.

**Sub-sampling Monolingual Data.** To simulate low-resource behavior along another axis, we sub-sample the monolingual corpora used by BiSparse-Dep to induce monolingual vectors, $\mathbf{X_e}, \mathbf{X_f}$. Specifically, $\mathbf{X_e}$ and $\mathbf{X_f}$ are trained using progressively smaller corpora.

[5] `bitbucket.org/yoavgo/word2vecf/`

| en with → <br> Model ↓ | ru | zh | ar | fr | avg. |
|---|---|---|---|---|---|
| Translation Baseline | | | | | |
| Mono-Dep | 50.1 | 52.3 | 51.8 | 54.5 | 52.2 |
| Window Based | | | | | |
| BiSparse-Lex | 56.6 | 53.7 | 50.9 | 52.0 | 53.3 |
| Bivec+ | 55.8 | 52.0 | 51.5 | 53.4 | 53.2 |
| Dependency Based | | | | | |
| Cl-Dep | **60.2** | 54.4 | **56.7*** | 53.8 | **56.3** |
| BiSparse-Dep (Full) | 59.0 | 55.9 | 52.6 | 56.6 | 56.0 |
| BiSparse-Dep (Joint) | 53.8 | **57.0*** | 52.4 | **59.9*** | 55.8 |
| BiSparse-Dep (Unlab) | 55.9 | 51.2 | 53.3 | 55.9 | 54.1 |

Table 19: Comparing the different approaches from Section 6.5 with our BiSparse-Dep approach on Hyper-Hypo (random baseline= 0.5). **Bold** denotes the best score for each language, and the * on the best score indicates a statistically significant ($p < 0.05$) improvement over the next best score, using McNemar's test (McNemar, 1947).

**Quality of Bilingual Dictionary.** We study the impact of the quality of the bilingual dictionary used to create the translation matrix $\mathbf{S}$. This experiment involves using increasingly smaller parallel corpora to induce the translation dictionary.

## 6.6. Experiments

The experiments aim to answer the following questions — (a) Are dependency based embeddings superior to window based embeddings for cross-lingual hypernymy? (Section 6.6.1) (b) Are our models robust in data scarce settings? (Section 6.7) (c) Does directionality in the dependency context help cross-lingual hypernymy detection? (d) Is the answer to (a) predicated on choice of hypernymy detection measure?

### *6.6.1. Dependency v/s Window Contexts*

We compare the performance of models described in Section 6.5 with the BiSparse-Dep (Full and Joint) models. We evaluate the models on the two test splits described in Section 6.4 — Hyper-Hypo and Hyper-Cohypo.

| en with → <br> Model ↓ | ru | zh | ar | fr | avg. |
|---|---|---|---|---|---|
| Translation Baseline | | | | | |
| Mono-Dep | 58.7 | 50.0 | 65.1 | 56.9 | 57.7 |
| Window Based | | | | | |
| BiSparse-Lex | **63.8** | 55.8 | 65.8 | 63.2 | 62.2 |
| Bivec+ | 55.9 | 64.9 | 62.2 | 54.1 | 58.3 |
| Dependence Based | | | | | |
| Cl-Dep | 56.2 | 62.7 | 63.1 | 61.0 | 60.0 |
| BiSparse-Dep (Full) | 63.6 | **67.3** | **66.8*** | **66.7*** | **66.1** |
| BiSparse-Dep (Joint) | 60.6 | 63.6 | 65.9 | 64.9 | 63.8 |
| BiSparse-Dep (Unlab) | 58.6 | 66.7 | 62.4 | 61.5 | 62.4 |

Table 20: Comparing the different approaches from Section 6.5 with our BiSparse-Dep approach on Hyper-Cohypo (random baseline= 0.5). **Bold** denotes the best score for each language, and the * on the best score indicates a statistically significant ($p < 0.05$) improvement over the next best score, using McNemar's test (McNemar, 1947).

**Hyper-Hypo Results.** Table 19 shows the results on the Hyper-Hypo set. The benefit of directly modeling the cross-lingual relationship between words (as opposed to first translating to English) is evident in that almost all models (except Cl-Dep on French) outperform the translation baseline. Among dependency based models, BiSparse-Dep (Full) and Cl-Dep consistently outperform both window models, while BiSparse-Dep (Joint) outperforms them on all except Russian. This confirms that dependency context are more effective than window context for cross-lingual hypernymy detection. BiSparse-Dep (Joint) and Cl-Dep are the best models for French, Chinese and Arabic, with no statistically significant difference between them for Russian.

**Hyper-Cohypo Results.** The trends observed on Hyper-Hypo also hold on Hyper-Cohypo, i.e., dependency based models continue to outperform window based models (Table 20). Overall, BiSparse-Dep (Full) performs best in this setting, followed closely by BiSparse-Dep (Joint). This suggests that the sibling information encoded in Joint is useful to distinguish hypernyms from hyponyms (Hyper-Hypo results), while the dependency labels encoded in Full help to distinguish hypernyms from co-hyponyms. Also, all models improve significantly on the Hyper-Cohypo set, suggesting that discriminating hypernyms from co-hyponyms is easier than discriminating hypernyms from hyponyms.

While the BiSparse-Dep models were generally performing better than window models on both test sets, Cl-Dep was not as consistent (e.g., it was worse than the best window model on Hyper-Cohypo). One reason for this is that *BalAPinc* is designed for sparse embeddings and is likely to perform poorly with dense embeddings (Turney and Mohammad, 2015). This explains the relatively inconsistent performance of Cl-Dep.

### *6.6.2. Ablating Directionality in Context*

The context described by the Full and Joint BiSparse models encodes directional information (Section 6.3) either in the form of label direction (Full), or using sibling information (Joint). *Is directionality in the context useful to capture asymmetric relationship hypernymy?* To answer this, a third BiSparse model that uses Unlabeled dependency contexts is evaluated. The Unlabeled context is similar to the Full context, except it does not concatenate the label of the relation to the context word (parent or children). For instance, for the word *traveler* in Figure 12, Unlabeled contexts are *roamed* and *tired*.

Experiments on both Hyper-Hypo and Hyper-Cohypo (bottom row, Tables 19 and 20) highlight that directional information is essential — Unlabeled almost always performs worse than Full and Joint, and in many cases worse than even window based models.

### *6.6.3. Qualitative Analysis*

The predictions of the window and the dependency based models are analyzed qualitatively by examining the common errors made by both BiSparse-Dep (Full) and BiSparse-Lex models in Table 21. Often both models have difficulty capturing functional relations, such as *époux* (spouse in French) is a *relative* or *бас* (bass in Russian) is an *instrument*, though overall, the dependency based model makes fewer mistakes of this type. For false positives,

| False Negatives | False Positives |
|---|---|
| (pêcheur (en:fisher), worker) | (vêtement (en:clothing), jeans) |
| (époux (en:spouse), relative) | (cercueil (en:coffin), casket) |
| (équipe (en:team), unit) | (statut (en:status), barony) |
| (бас (en:bass), instrument) | (ткань (en:tissue), nerve) |
| (замок (en:castle), structure) | (флора (en:flora), sunflower) |
| (трейлер (en:trailer), vehicle) | (чувство (en:feeling), cynicism) |

Table 21: Common errors made by the different models described in Section 6.3.

both models often ignore the directional nature, predicting *vêtement* (clothing in French) is a *jeans* or *ткань* (Russian for tissue) is a *nerve* as positive instances.

Many test pairs were hard for both models due to translation ambiguity. For instance, in (*dessous*, *garment*) from the French dataset, *dessous* can be translated to *underneath*, however the apt translation (i.e., under which hypernymy holds) is *underwear* in this example. For another French pair (*poire*, *fruit*), the word *poire* can be translated to *perry*, which is a drink, but the apt translation is *pear*. Similarly, for the Russian pair (лук, *produce*), лук can be translated to *bow*, but the apt translation is *onion*. In (глава, leader), глава can be incorrectly translated to *chapter*, but the apt translation is *head*.

## 6.7. Evaluating Robustness of BiSparse-Dep

The experiments in the previous section highlight the robust behavior of the BiSparse models across languages when exposed to all available information. The experiments in this section investigate how the BiSparse-Dep models behave when exposed to various data scarce conditions. Three different settings are examined: (a) no dependency treebank in a language (b) small monolingual corpus (c) lower quality bilingual dictionary.

### *6.7.1. No Treebank*

These experiments are conducted with a version of BiSparse-Dep that used the Full context type for both English and the non-English (target) language, but the target language contexts are derived from a corpus parsed using a delexicalized parser (Section 6.5). This BiSparse-Dep model is compared against the best window-based model and the best

| en with →<br>Model ↓ | ru | zh | ar | fr | avg. |
|---|---|---|---|---|---|
| Hyper-Hypo | | | | | |
| Best Win. | 56.6 | 53.7 | 51.5 | 53.4 | 53.8 |
| Delex. | 59.1* | 55.1* | 54.6* | 56.1* | 56.2 |
| Best Dep. | 60.2 | 57.0* | 56.7* | 59.9* | 58.5 |
| Hyper-Cohypo | | | | | |
| Best Win. | 63.8 | 64.9 | 65.8 | 63.2 | 64.4 |
| Delex. | 59.4 | 65.7* | 67.5* | 66.3* | 64.7 |
| Best Dep. | 63.6* | 67.3* | 66.8* | 66.7 | 66.1 |

Table 22: **Robustness to absence of a treebank:** The delexicalized model is competitive to the best dependency based and the best window based models on both test sets. For each dataset, * indicates a statistically significant ($p < 0.05$) improvement over the next best model in that column, using McNemar's test (McNemar, 1947).

dependency-based model from the previous section. Results are in in Table 22.

The results show that this version of BiSparse-Dep compares favorably on all language pairs against the best window based and the best dependency based model. In fact, it almost consistently outperforms the best window based model by several points, and is only slightly worse than the best dependency-based model.

Further analysis revealed that the good performance of the delexicalized model is due to the relative robustness of the delexicalized parser on frequent contexts in the co-occurrence matrix. In French and Russian, the most frequent contexts were derived from `amod`, `nmod`, `nsubj` and `dobj` edges.[6] Of these, the `nmod` edge appears in 44% and 33% of the Russian and French contexts respectively. The delexicalized parser predicts both the label and direction of the `nmod` edge correctly with an F1 of 68.6 for Russian and 69.6 for French, compared to a fully-trained parser (76.7 F1 for Russian, 76.8 F1 for French).

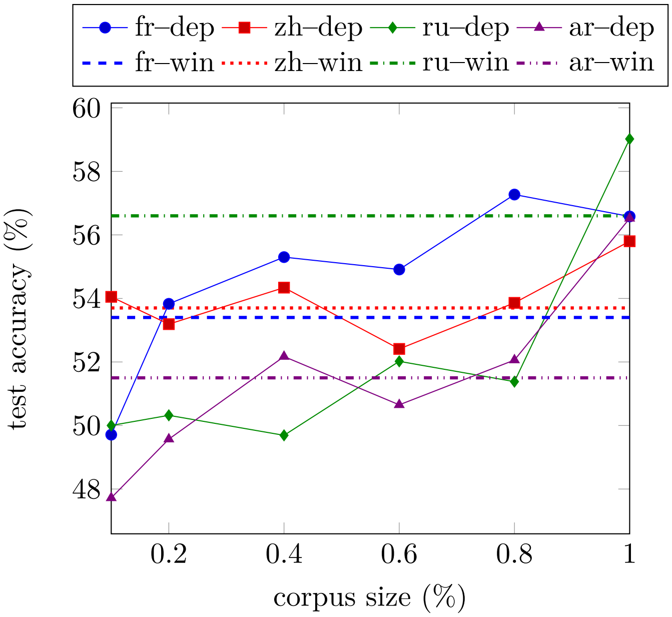


Figure 13: **Robustness to small corpus:** For most languages, BiSparse-Dep outperforms the corresponding best window based model on Hyper-Hypo, with about 40% of the monolingual corpora.

### *6.7.2. Small Monolingual Corpus*

In this experiment, increasingly smaller monolingual corpora (10%, 20%, 40%, 60% and 80%) sampled at random are used to induce the monolingual vectors for BiSparse-Dep (Full) model to evaluate its robustness to small corpora. Trends in Figure 13 indicate that BiSparse-Dep models that use only 40% of the original data remain competitive with the BiSparse-Lex model that has uses all the data. Robust performance with smaller monolingual corpora is desirable since large corpora are not always easily available.

### *6.7.3. Quality of Bilingual Dictionary*

Bilingual dictionaries derived from smaller amounts of parallel data are likely to be of lower quality compared to those derived from larger corpora. To analyze the impact of dictionary quality on BiSparse-Dep (Full), increasingly smaller parallel corpora are used to induce bilingual dictionaries as the score matrix **S** (Section 6.3). For this experiment, the top 10%,

[6] Together they make up at least 70% of the contexts.

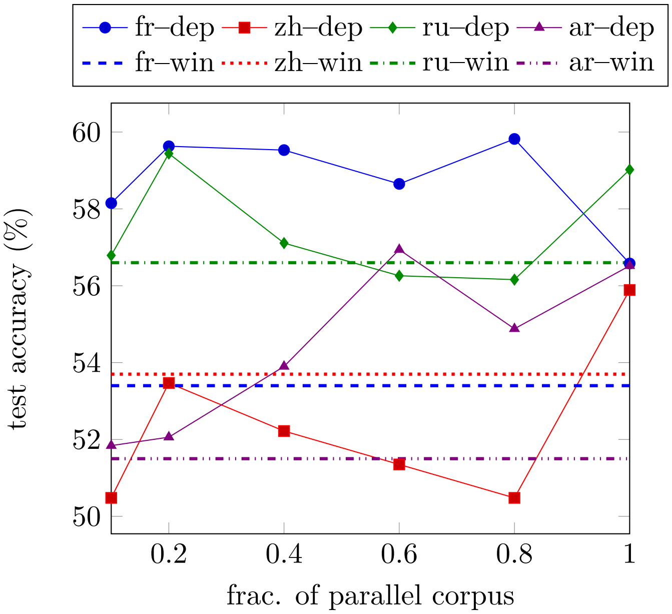


Figure 14: **Robustness to low quality dictionary:** For most languages, BiSparse-Dep outperforms the corresponding best window based model on Hyper-Hypo, with increasingly lower quality dictionaries.

20%, 40%, 60% and 80% sentences from the parallel corpora are used. Figure 14 show that even with a lower quality dictionary, BiSparse-Dep performs better than BiSparse-Lex.

### *6.7.4. Choice of Hypernymy Detection Measure*

To see if the conclusions drawn depend on the choice of the hypernymy detection measure, the measure is changed from *BalAPinc* to *SLQS* (Santus et al., 2014) and the experiments from Section 6.6.1 are redone. *SLQS* is based on the *distributional informativeness hypothesis*, that states that hypernyms are less "informative" than hyponyms because they occur in more general contexts. The informativeness $E_u$ of a word $u$ is defined to be the median entropy of its top $N$ dimensions, $E_u = median_{k=1}^{N} H(c_k)$, where $H(c_i)$ denotes the entropy of dimension $c_i$. The *SLQS* score for a pair $(u, v)$ is the relative difference in entropies,

$$SLQS(u \to v) = 1 - \frac{E_u}{E_v}$$

Recent work from Shwartz et al. (2017) has found *SLQS* to be more successful than other measures in monolingual hypernymy detection.

The results with *SLQS* are consistent with those in Section 6.6.1 — both BiSparse-Dep models still outperform window-based models. Also, the delexicalized version of BiSparse-Dep outperforms the window-based models, showing that the robust behavior demonstrated in Section 6.7 is also invariant across measures. It was found that using *BalAPinc* led to better results than *SLQS*. For both BiSparse-Dep models, *BalAPinc* wins across the board for two languages (Russian and Chinese), and wins half the time for the other two languages compared to *SLQS*.

## 6.8. Summary

This chapter demonstrated that cross-lingual representations can also reveal asymmetric relationships like hypernymy between words in different languages. This was shown using a new distributional approach, namely BiSparse-Dep, based on cross-lingual embeddings derived from dependency contexts. By appealing to the distributional inclusion hypothesis to probe these representations, hypernymy relationships between words in different languages was determined. Experimental results show that using BiSparse-Dep is superior to standard window based models and a translation baseline. The approach was also shown to be robust to various low-resource scenarios, especially when the syntactic contexts were derived using a weak dependency parser, an encouraging result for low-resource languages.

Several interesting future directions remain open — incorporating path-based information indirectly into the representation, or replacing the delexicalized parser with more sophisticated transfer strategies (Rasooli and Collins, 2017; Aufrant et al., 2016) might prove beneficial. It remains to be seen how our approach performs for other language pairs beyond simulated low-resource settings. The approach has the potential to complement existing work on creating cross-lingual ontologies such as BabelNet and the Open Multilingual WordNet, which are noisy because they are compiled semi-automatically, and have limited language coverage. In such settings, distributional approaches can help refine ontology construction for any language where sufficient resources are available.

This chapter assumed that the hypernymy relationship can be detected out-of-context, i.e., without paying heed to the context in which the words appear. This is a common assumption made in most monolingual lexical semantic tasks (Kotlerman et al., 2010; Baroni and Lenci, 2011; Baroni et al., 2012). However, this may not be true for polysemous words like *bat* (that can take both *mammal* and *equipment* as hypernyms). This criticism is related to that of static word representations discussed in Section 5.8. An important future direction is considering the cross-lingual hypernymy detection task in context, as has been done recently in the monolingual setting by Shwartz and Dagan (2015) and Vyas and Carpuat (2017).

# CHAPTER 7 : Multilingual Supervision for Cross-lingual Entity Linking

## 7.1. Introduction

The web is a valuable source of knowledge, expressed through text written either in English or other languages.[1] As most of text on the web is unstructured, information extraction tasks (e.g., coreference resolution, named entity recognition etc.) have been developed to extract structured information to facilitate better text understanding. Additionally, collective efforts to organize knowledge about entities and concepts have lead to creation of *knowledge bases* (KB) like Wikipedia, DBPedia (Auer et al., 2007), Freebase (Bollacker et al., 2008), and YAGO (Hoffart et al., 2013). Nevertheless, to be able to keep up with continuously generated new information about entities (old and new), one has to find a way to bridge the unstructured raw text and the structured knowledge bases.

A key step to bridging this gap is to identify and ground named entities expressed in raw text to their corresponding entries in the KB. This task is fraught with *ambiguity* and *variability* of natural language — the same string can refer to multiple entities (e.g., *St. Mary's Church* can refer to over 500 different churches over the world) and the same entity can be described in many ways (e.g., *New York City*, *the Big Apple*, *NYC*, *The Five Boroughs* all refer to the entity `New_York`). Furthermore, grounding entities in multilingual text adds the difficulty of comparing contexts in different languages. Developing approaches to facilitate this task in multilingual texts is the focus of this and the next chapter.

In this chapter, I will discuss how we can leverage cross-lingual representations to *share* supervision between languages, to perform a downstream information extraction task, namely, entity linking in the cross-lingual setting.

I will first briefly describe the entity linking task and its history, including a discussion of the challenges involved in the cross-lingual version of the task.

[1] >40% of the Web is non-English content (`en.wikipedia.org/wiki/Internet_languages`).

## 7.2. History of the Entity Linking Task

The task of entity linking involves grounding a span of text in a document to some background structured knowledge representation. More formally, given a mention $m$ in a document $\mathcal{D}$, entity linking (EL) involves linking $m$ to its gold entity $e^*$ in a KB, $\mathcal{K} = \{e_1, \cdots, e_n\}$, for all mentions $m \in \mathcal{D}$. For instance, for the following document containing mentions Jay Cutler and Dolphin and the English Wikipedia as the target KB, EL involves grounding Jay Cutler to the Wikipedia article `https://en.wikipedia.org/wiki/Jay_Cutler` and Dolphin to `https://en.wikipedia.org/wiki/Miami_Dolphins`.

[**Jay Cutler**] passed his physical and is officially a [**Dolphin**].

Understanding raw text often requires appealing to background knowledge about entities. For instance, in the example above, we need to know that Dolphin is the nickname for a player of the Miami Dolphins football team, and not (say) the aquatic mammal.

By augmenting a piece of text with such background knowledge, EL helps a human reader access relevant factual knowledge to complement the information expressed in the text. This not only aids understanding, but also serves as a useful component in other NLP systems. For instance, EL has been used for building question answering systems (Khalid et al., 2008; Yih et al., 2015; Sun et al., 2015), incorporating knowledge into coreference resolution systems (Ratinov and Roth, 2012; Hajishirzi et al., 2013; Rahman and Ng, 2011) and extracting new facts for knowledge base population (Ji and Grishman, 2011).

**Relation to Other NLP Tasks.** Entity linking is a closely related task to (a) Cross-Document Coreference Resolution (CDCR) (Mayfield et al., 2009) that aims to resolve coreference relationships between entity mentions *across* documents, and (b) Word Sense Disambiguation (WSD) (Navigli, 2009) discussed earlier in Chapter 5. The relationship with CDCR can be seen when we consider the union of the KB $\mathcal{K}$ and document $\mathcal{D}$ as a collection of documents — entity linking now reduces to resolving coreference relationships between entities across documents. The relation with WSD is more obvious one; the object to be disambiguated in WSD is a word instead of an entity mention, and the KB is a sense inventory instead of an encyclopedic resource like Wikipedia. In principle, both EL and WSD are dealing with the ambiguity problem in natural language. A major difference between the tasks is that named entities are often ambiguous, whereas most words assume only a single sense (i.e., monosemous) (Hachey et al., 2013). In this respect, EL is a harder disambiguation task that WSD.

### *7.2.1. Monolingual Entity Linking*

Previous work on entity linking has predominantly examined the problem with respect to English documents and an English KB like Wikipedia or Freebase (Bunescu and Paşca, 2006; Mihalcea and Csomai, 2007; Ratinov et al., 2011; Hoffart et al., 2011; Cheng and Roth, 2013; Shen et al., 2015, inter alia). All such entity linking systems involve two steps: *candidate generation* and *context-sensitive inference*, that together deal with ambiguity and variability of entities in text. I describe them below.

#### **Candidate Generation**

Considering all entities in the knowledge base as possible disambiguation for a mention $m$ is impractical, as the KB $\mathcal{K}$ can contain millions of entities. Therefore, it is prudent to filter out irrelevant entities that the mention is unlikely to refer to. Candidate Generation identifies a small set $\mathcal{C}(m)$ of plausible entities that a given mention $m$ can link to. Given $m$, candidate generation outputs a list of candidate entities $\mathcal{C}(m) = \{e_1, e_2, \cdots, e_K\}$ of size at most $K$, each associated with a prior probability $\mathrm{Pr}_{\mathrm{prior}}(e_i \mid m)$ indicating the probability of $m$ referring to $e_i$, given only $m$'s surface.

To do this, first a dictionary mapping mention surfaces (strings) to entities that they can refer to, is compiled. For instance, such a dictionary will map *chicago* to the set of entities {`Chicago`, `University_of_Chicago`, `Chicago_Cubs`, $\cdots$ }. One approach to compile such a dictionary is to crawl a large collection of hyperlinked documents and computing the fre-

| String | Entity (e) | $\Pr_{\text{prior}}(e \mid m)$ | counts/total |
|---|---|---|---|
| *chicago* | `Chicago` | 0.275 | 62939/228690 |
| *chicago* | `University_of_Chicago` | 0.075 | 17186/228690 |
| *chicago* | `Chicago_Cubs` | 0.045 | 10332/228690 |
| *chicago* | `Chicago_Tribune` | 0.041 | 9457/228690 |
| ··· | `Chicago_White_Sox` | 0.040 | 9184/228690 |
| ··· | `Chicago_Bears` | 0.039 | 9018/228690 |
| ··· | `Art_Institute_of_Chicago` | 0.015 | 3538/228690 |
| ··· | `Chicago_(band)` | 0.005 | 1225/228690 |
| *chicago* | `Loyola_University_Chicago` | 0.005 | 1057/228690 |

Table 23: Part of a dictionary compiled from Wikipedia hyperlinks showing the entities that the string *chicago* can refer to, along with the respective prior probabilities and counts.

quency with which an anchor text (or mention surface) links to a page in Wikipedia, such as the one shown in Table 23. From the frequencies, one can estimate the conditional probability $\Pr_{\text{prior}}$. For instance, if the string *chicago* appears hyperlinked 228690 times in the document collection, and links to `Chicago` and `University_of_Chicago` 62939 and 17186 times respectively, then the probability of a mention [**chicago**] referring to `Chicago` and `University_of_Chicago` is 0.275 and 0.075 respectively. In the literature, this collection of documents can either be hyperlinked text from the Web (Spitkovsky and Chang, 2012), or simply the articles in Wikipedia itself (Ratinov et al., 2011).

### Context-Sensitive Inference

Having identified a small set of plausible entities, the context-sensitive inference step predicts one of $\hat{e} \in C(m)$ as the disambiguation for mention $m$. For this, a compatibility score $\phi(m \to e)$ of the $m$ grounding to a candidate entity $e$ is computed using features extracted from the context of $m$ and information about $e$ expressed in the KB $\mathcal{K}$. The compatibility score is trained using *grounded mentions*, i.e., mentions of entities that are grounded, as supervision, such that the one shown in Figure 15. An inference problem is then solved to find the best assignment for $m$. The different approaches for performing context-sensitive inference vary in these two steps: choice of the features in compatibility score (*hand-crafted* or *learnt*) and inference strategy (*local inference* or *global inference*).

**Choice of Features in $\phi(m \rightarrow e)$.** The score $\phi(m \rightarrow e)$ can be computed using *hand-crafted* or *learnt* features. Ratinov et al. (2011) use hand-crafted features comparing the content of entity $e$'s page and mention $m$'s context. For instance, the cosine similarity of the TF-IDF vectors of $e$'s page and $m$'s context All these manually defined features are compiled into a feature vector $\mathbf{f}(e, m)$, such that $\phi(m \rightarrow e) = \mathbf{w}^{\intercal}\mathbf{f}(e, m)$ where $\mathbf{w}$ is a learnt weight vector.[2] On the other hand, approaches like (Gupta et al., 2017; Sil et al., 2018) learn feature representations $\mathbf{e}$ for the entity and $\mathbf{g}$ for the mention's context, so that $\phi(m \rightarrow e) = \mathbf{e}^{\intercal}\mathbf{g}$. The representations $\mathbf{e}$ and $\mathbf{g}$ are learnt end-to-end by optimizing a loss defined for the disambiguation task using back-propagation (Rumelhart et al., 1986).

**Local Inference.** Local inference resolves each mention $m \in \mathcal{D}$ in isolation, with the assignment to mention $m_i$ having no influence to assignment to mention $m_j$ where $i \neq j$.

$$\hat{e}_i = \underset{e_i \in C(m_i)}{\arg\max} \quad \phi(m_i \rightarrow e_i) \qquad \forall\, m_i \in \mathcal{D} \tag{7.1}$$

Local inference is a popular approach (Mihalcea and Csomai, 2007; Durrett and Klein, 2014; Lazic et al., 2015; Francis-Landau et al., 2016), owing to its simplicity.

**Global Inference.** The local inference approach does not take into account relationship between candidate entities of different mentions in the same document. For instance, if `Steven_Gerrard` is a candidate for mention $m_i \in \mathcal{D}$ and `Liverpool_F.C.` is a candidate for mention $m_j \in \mathcal{D}$, then assigning $m_i \rightarrow$ `Steven_Gerrard`, $m_j \rightarrow$ `Liverpool_F.C.` is *coherent* because these two entities are related.[3] This correlation between entities is expressed through a pairwise coherence score $\psi(m_i \rightarrow e_i, m_j \rightarrow e_j)$ that is used along with the learnt compatibility score $\phi(m \rightarrow e)$ to formulate a global inference problem,

$$(\hat{e}_1, \hat{e}_2, \cdots, \hat{e}_n) = \underset{e_i \in C(m_i)}{\arg\max} \sum_i \phi(m_i, e_i) + \sum_{i \neq j} \psi(m_i \rightarrow e_i, m_j \rightarrow e_j) \qquad \forall\, m_i \in \mathcal{D} \tag{7.2}$$

[2] A non-linear kernel (e.g., polynomial kernel) can also be used.

[3] `Steven_Gerrard` captained `Liverpool_F.C.` from 2003-15.

சுரேஸ் **[லிவர்பூல்]** மற்றும் உருகுவே விளையாடுகிறார்.

Everton won against **[Liverpool]** in an FA Cup match.

Figure 15: Tamil and English mention contexts containing [**mentions**] of the entity `Liverpool_F.C.` from the respective Wikipedias. Tamil Wikipedia only has 9 mentions referring to `Liverpool_F.C.`, whereas English Wikipedia has 5303 such mentions. Clearly, there is a need to augment the limited contextual evidence in low-resource languages (like Tamil) with evidence from high-resource languages (like English). The Tamil sentence translates to "Suarez plays for [**Liverpool**] and Uruguay."

This global inference problem is NP-hard (Cucerzan, 2007), so approximate inference approaches have been proposed that decompose the inference problem to smaller but tractable inference problems using a technique like message passing (Globerson et al., 2016; Ganea and Hofmann, 2017). The pairwise coherence score $\psi$ itself can either be hand crafted relational scores between entities imposed only at inference time (Cheng and Roth, 2013) or learnt end-to-end with the rest of the model (Ganea and Hofmann, 2017).

### *7.2.2. Cross-lingual Entity Linking*

The task of Cross-lingual Entity Linking (XEL) (McNamee et al., 2011; Ji et al., 2015; Tsai and Roth, 2016) involves grounding entity mentions written in *any* language (i.e., the document $\mathcal{D}$ containing the query mention $m$) to an English Knowledge Base (e.g., the English Wikipedia). For instance, Figure 15 shows a Tamil (a language with >70 million speakers) and an English mention (shown [**enclosed**]) and their mention contexts. XEL involves grounding the Tamil mention (which translates to 'Liverpool') to the football club `Liverpool_F.C.`, and not the city or the university. XEL enables knowledge acquisition directly from documents in any language, without resorting to machine translation.

Unlike monolingual entity linking for English, the cross-lingual entity linking is relatively less explored. The cross-lingual nature of the problem introduces new challenges to the entity linking problem. Firstly, candidate generation has to operate with mention whose surfaces are in a different writing script. For instance, in Figure 15, candidates have to be generated for the Tamil surface form (written in the Tamil script) that translates to *Liverpool*. Another challenge in the cross-lingual setting is to find supervision for the context-sensitive inference component. In this chapter, I will examine the latter challenge, and discuss an approach to address the former challenge in Chapter 8.

Existing approaches have taken two main directions to obtain supervision for learning XEL models — **(a)** using mention contexts appearing in the target language (McNamee et al., 2011; Tsai and Roth, 2016), or **(b)** using mention contexts appearing only in English (Pan et al., 2017; Sil et al., 2018). We describe these directions and their limitations below.

**Target Supervision Only.** The first direction, exemplified by approaches like McNamee et al. (2011) and Tsai and Roth (2016) utilizes supervision available in the target language to train the XEL model. McNamee et al. (2011) use annotation projection via parallel corpora to generate supervision in the target language. They generate mention contexts in the target language by first running an entity linking system on the English side of a parallel corpus, and then projecting the resulting annotations via word alignments to the target language. Such annotation projection approaches are not scalable as parallel data is expensive to obtain. On the other hand, Tsai and Roth (2016) uses mention contexts from the target language Wikipedia to learns separate XEL models for each language. Both these approach have scalability issues for languages with limited resources. Another limitation of these approaches is that they train separate models for each language, which is inefficient when working with multiple languages.

**English Supervision Only.** Other approaches, like that of Pan et al. (2017) and Sil et al. (2018) only use mention contexts from English. While Pan et al. (2017) compute entity coherence statistics from English Wikipedia, and uses other mentions in the document to jointly rank candidate entities in an unsupervised manner. Similarly, Sil et al. (2018) perform zero-shot XEL for Chinese and Spanish by using multilingual embeddings to transfer a pre-trained English EL model. Sil et al. (2018) use the term *zero-shot* in the sense that the model was not trained on the target language. However, intuitively, mention contexts in the target language are a valuable source of language-specific contextual information, and should also be used if available. Indeed, a recent study (Lewoniewski et al., 2017) found that for language sensitive topics, the quality of information can be better in the relevant language version of Wikipedia than the English version.

### *7.2.3. Our Work*

As discussed earlier, training an EL model requires grounded mentions, i.e., mentions of entities that are grounded to a Knowledge Base (KB), as supervision (Figure 15). While millions of such mentions are available in English, by virtue of hyperlinks in the English Wikipedia, this is not the case for most languages. This makes learning XEL models challenging, especially for languages with limited resources (e.g., the Tamil Wikipedia is only 1% of the English Wikipedia in size). To overcome this, it is desirable to augment the limited contextual evidence available in the language with evidence from high-resource languages like English.

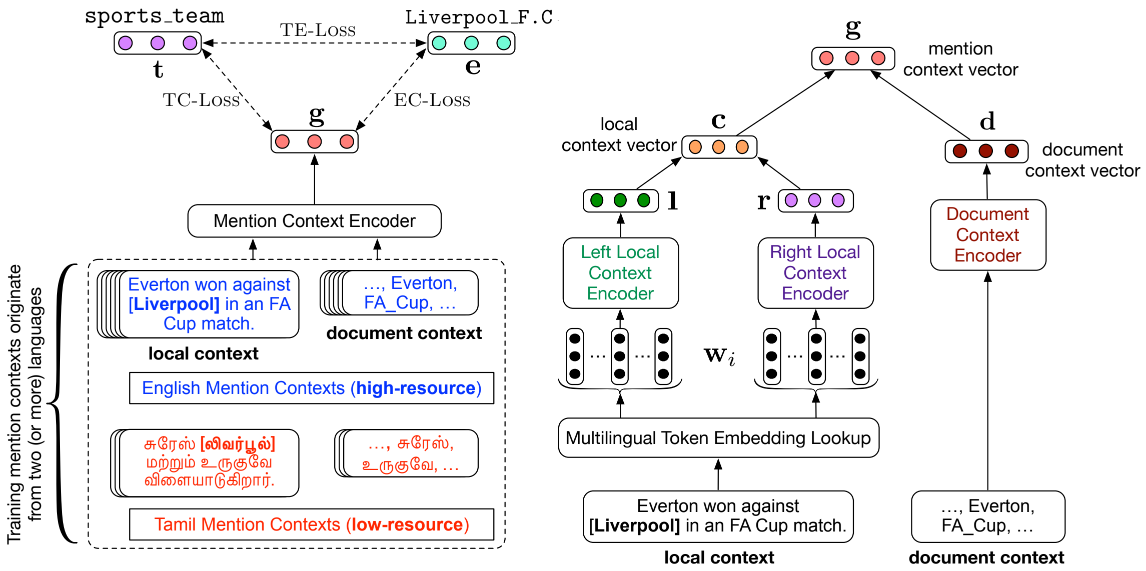
In this chapter, I will describe XELMS (XEL with Multilingual Supervision) (Section 7.3), the first approach that fulfills the above desiderata by using multilingual supervision to train an XEL model. XELMS represents the mention contexts of the same entity from different languages in the same semantic space using a single context encoder (Section 7.3.1). Language-agnostic entity representations are jointly learned with the relevant mention context representations, so that an entity and its context share similar representations. Additionally, by encoding freely available structured knowledge, like fine-grained entity types, the entity and context representations can be further improved (Section 7.3.2). The architecture of XELMS is inspired by several monolingual entity linking systems (Francis-Landau et al., 2016; Nguyen et al., 2016b; Gupta et al., 2017), approaches that use type information to aid entity linking (Ling et al., 2015; Gupta et al., 2017; Raiman and Raiman, 2018), and the recent success of multilingual embeddings for several tasks (Ammar et al., 2016a; Duong et al., 2017b).

The ability to use multilingual supervision enables XELMS to learn XEL models for target languages with limited resources by exploiting freely available supervision from high resource languages (like English). The experiments show that XELMS outperforms existing state-of-the-art approaches that only use target language supervision, across 3 benchmark datasets in 8 languages (Section 7.6.1). Moreover, while previous XEL models (McNamee et al., 2011; Tsai and Roth, 2016) train separate models for different languages, XELMS can train a *single* model for performing XEL in multiple languages (Section 7.6.2).

One of the goals of XEL is to enable understanding of languages with limited resources. To evaluate this, experiments are performed in two such settings. Experiments in the *zero-shot setting* (Section 7.7.1), where *no* supervision is available in the target language, show that the good performance of zero-shot XEL approaches (Sil et al., 2018) can be attributed to the use of prior probabilities. As described earlier (Section 7.2.1), these probabilities are computed from large amount of grounded mentions, which are not available in realistic zero-shot settings. Experiments in the *low-resource setting* (Section 7.7.2), where some supervision is available in the target language, show that even when only a fraction of the available supervision in the target language is provided, XELMS can achieve competitive performance by exploiting supervision from English.

## 7.3. Cross-lingual EL with XELMS

An overview of XELMS is shown in Figure 16a. XELMS computes the conditional probability, $\text{Pr}_{\text{context}}(e \mid m)$, of a mention $m$ referring to entity $e \in \mathcal{K}$ using a mention context vector $\mathbf{g} \in \mathbb{R}^h$ representing $m$'s context, and an entity vector $\mathbf{e} \in \mathbb{R}^h$, representing the entity $e \in \mathcal{K}$ (one vector per entity). XELMS can also incorporate structured knowledge like fine-grained entity types (Section 7.3.2) using a multi-task learning approach (Caruana, 1998), by learning a type vector $\mathbf{t} \in \mathbb{R}^h$ for each possible type $t$ (e.g., `sports_team`) associated with the entity $e$. The entity vector $\mathbf{e}$, context vector $\mathbf{g}$ and the type vector $\mathbf{t}$ are jointly trained, and interact through appropriately defined pairwise loss terms — an Entity-Context loss (EC-LOSS), Type-Entity loss (TE-LOSS) and a Type-Context loss (TC-LOSS).


sports_team
TE-Loss
Liverpool_F.C
t
e
TC-Loss
EC-Loss
g
Mention Context Encoder
Training mention contexts originate from two (or more) languages
Everton won against [Liverpool] in an FA Cup match.
..., Everton, FA_Cup, ...
local context
document context
English Mention Contexts (high-resource)
சுரேஸ் [லிவர்பூல்] மற்றும் உருகுவே விளையாடுகிறார்.
..., சுரேஸ், உருகுவே, ...
Tamil Mention Contexts (low-resource)
g
mention context vector
local context vector
c
d
document context vector
l
r
Left Local Context Encoder
Right Local Context Encoder
Document Context Encoder
$\mathbf{w}_i$
Multilingual Token Embedding Lookup
Everton won against [Liverpool] in an FA Cup match.
..., Everton, FA_Cup, ...
local context
document context

(a) Overview of XELMS. Mentions are [**enclosed**].

(b) Mention Context Encoder.

Figure 16: (**a**) XELMS uses grounded mentions from two or more languages (English and Tamil shown) as supervision. The context **g**, entity **e** and type **t** vectors interact through Entity-Context loss (EC-LOSS), Type-Context loss (TC-LOSS) and Type-Entity loss (TE-LOSS). The Tamil sentence is the same as in Figure 15, and other mentions in it translate to [**Suarez**] and [**Uruguay**]. (**b**) The Mention Context Encoder (Section 7.3.1) encodes the local context (neighboring words) and the document context (surfaces of other mentions in the document) of the mention into **g**. Internal view of local context encoder is in Figure 17.

The mention context vector **g** is computed using a mention context encoder (Section 7.3.1), shown in Figure 16b. The encoder is a function of the *mention context* of mention $m$ in a document $\mathcal{D}$, which consists of: **(a)** neighboring words around the mention, which is referred to as its *local context* and, **(b)** surfaces of other mentions appearing in $\mathcal{D}$, which is referred to as its *document context*.

XELMS is trained using grounded mentions in multiple languages (English and Tamil in Figure 16a) that is derived from Wikipedia (Section 7.5).

### *7.3.1. Mention Context Representation*

To learn from mention contexts in multiple languages, mention context representations are generated using a language-agnostic mention context encoder. An overview of the mention context encoder is shown in Figure 16b. The components of the mention context encoder, namely multilingual word embeddings, a local context encoder and a document context encoder, are described below.

**Multilingual Word Embeddings.** Multilingual word embeddings jointly encode words in multiple ($\geq$2) languages in the same vector space such that semantically similar words in the same language and translationally equivalent words in different languages are close per cosine similarity (Ammar et al., 2016b; Smith et al., 2017; Duong et al., 2017b). These embeddings generalize bilingual embeddings (such as those discussed in Chapter 3), which do the same for two languages *only*.

XELMS uses FASTTEXT (Bojanowski et al., 2017; Smith et al., 2017) multilingual embeddings that aligns monolingual embeddings of multiple languages in the same space using a small bilingual dictionary ($\sim$2500 pairs) from each language to English. Both the embeddings and the dictionary can be easily obtained for languages with limited resources.

The multilingual word embeddings for tokens $\{w_1, w_2, \cdots, w_n\}$ are denoted by $\mathbf{w}_{1:n} = \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_n\}$, where each $\mathbf{w}_i \in \mathbb{R}^d$.

**Local Context Representation.** The local context of a mention $m$, spanning tokens $i$ to $j$, consists of left context (tokens $i - W$ to $j$) and right context (tokens $i$ to $j + W$). For example, for the mention [**Liverpool**] in Figure 16b, the left and right contexts are "Everton won against Liverpool" and "Liverpool in a FA Cup match" respectively. The local context encoder (Figure 17) encodes the left and the right contexts into vectors $\mathbf{l} \in \mathbb{R}^h$ and $\mathbf{r} \in \mathbb{R}^h$ respectively, using a convolutional neural network (CNN). These two vectors are then combined to generate the local context vector $\mathbf{c} \in \mathbb{R}^h$ (Figure 16b).

The CNN convolves continuous spans of $k$ tokens using a filter matrix $\mathbf{F} \in \mathbb{R}^{kd \times h}$ to project the concatenation ($\oplus$ operator) of the token embeddings in the span. The resulting vector is passed through a ReLU unit[4] to generate convolutional output $\mathbf{O}_i$. The outputs $\{\mathbf{O}_i\}$

[4] A ReLU activation unit (Nair and Hinton, 2010) is defined to be the function $f(x) = \max(0, x)$

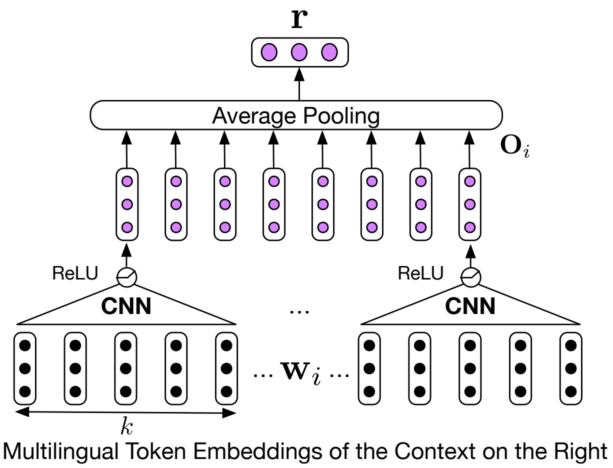


Figure 17: Local Context Encoder, for the right context. Figure 16b shows how it fits inside Mention Context Encoder.

are pooled by averaging (AVG) to obtain the final encoding $\textsc{Enc}(\mathbf{w}_{1:n})$,

$$\mathbf{O}_i = \textsc{relu}(\mathbf{F}^{\mathsf{T}}(\mathbf{w}_i \oplus \cdots \oplus \mathbf{w}_{i+k-1})) \tag{7.3}$$

$$\textsc{Enc}(\mathbf{w}_{1:n}) = \textsc{avg}(\mathbf{O}_1, \cdots, \mathbf{O}_{n-k+1}) \tag{7.4}$$

Left and right context vectors $\mathbf{l}$ and $\mathbf{r}$ are computed using respective $\textsc{Enc}(.)$ layers,

$$\mathbf{l} = \textsc{Enc}_{\text{left}}(\mathbf{w}_{i-W} \cdots \mathbf{w}_j) \tag{7.5}$$

$$\mathbf{r} = \textsc{Enc}_{\text{right}}(\mathbf{w}_i \cdots \mathbf{w}_{j+W}) \tag{7.6}$$

These vectors together generate the local context vector $\mathbf{c} = \mathcal{F}_{2h,h}(\mathbf{l} \oplus \mathbf{r})$. Here $\mathcal{F}_{d_i,d_o} : \mathbf{v}_i \rightarrow \mathbf{v}_o$ denotes a feed-forward layer that takes $\mathbf{v}_i \in \mathbb{R}^{d_i}$ as input, and outputs $\mathbf{v}_o \in \mathbb{R}^{d_o}$.

**Document Context Representation.** Presence of certain mentions in a document can help disambiguate other mentions. For example, “Suarez”, “Everton” in a document can help disambiguate “Liverpool”. To incorporate this, the document context $d_m$ of a mention $m$ appearing in document $\mathcal{D}$ is defined to be the bag of all other mentions in $\mathcal{D}$. The document context $d_m$ is encoded into a dense document context vector $\mathbf{d} \in \mathbb{R}^h$ by a feed-forward layer $\mathbf{d} = \mathcal{F}_{|V|,h}(d_m)$. Here $V$ is the set containing all mention surfaces seen during training. When training jointly over multiple languages, $V$ consists of mention surfaces seen in all languages (e.g., all English and Tamil mention surfaces) during training. This enables parameter sharing by embedding mention surfaces in different languages in the same low-dimensional space.

The local and document context vectors $\mathbf{c}$ and $\mathbf{d}$ are combined to get the mention context vector $\mathbf{g}$ using a feed-forward layer $\mathbf{g} = \mathcal{F}_{2h,h}(\mathbf{c} \oplus \mathbf{d})$.

**Context Conditional Probability.** The probability of a mention $m$ linking to entity $e$ is computed using its context vector $\mathbf{g}$ and the entity vector $\mathbf{e}$,

$$\Pr_{\text{context}}(e \mid m) = \frac{\exp(\mathbf{g}^{\intercal}\mathbf{e})}{\sum\limits_{e' \in C(m)} \exp(\mathbf{g}^{\intercal}\mathbf{e}')} \tag{7.7}$$

where $C(m)$ denotes candidate entities of the mention $m$ (Section 7.4.1 explains how $C(m)$ is generated). The negative log-likelihood of $\Pr_{\text{context}}(e \mid m)$ is minimized with respect to the gold entity $e^*$ against the candidate entities $C(m)$. This loss term is called the Entity-Context loss (EC-Loss) in the rest of the chapter,

$$\text{EC-Loss} = -\log \frac{\Pr_{\text{context}}(e^* \mid m)}{\sum\limits_{e' \in C(m)} \Pr_{\text{context}}(e' \mid m)} \tag{7.8}$$

### *7.3.2. Including Type Information*

Incorporating the fine-grained types of a mention $m$ can help rank entities of the appropriate type higher than others (Ling et al., 2015; Gupta et al., 2017; Raiman and Raiman, 2018). For instance, knowing `sports_team` is the type of [**Liverpool**] and constraining linking to entities with the relevant type, encourages disambiguation to the correct entity.

To make the mention context representation $\mathbf{g}$ type-aware, the model predicts the set of fine-grained types of $m$, $\mathbf{T}(m) = \{t_1, ..., t_{|\mathbf{T}(m)|}\}$ using $\mathbf{g}$. Each $t_i$ belongs to a pre-defined type vocabulary $\Gamma$.[5] The probability of a type $t$ belonging to $\mathbf{T}(m)$ given the mention context is defined as $\Pr(t \mid m) = \sigma(\mathbf{t}^{\intercal}\mathbf{g})$, where $\sigma$ is the sigmoid function and $\mathbf{t}$ is the learnable embedding for type $t$. For this task, a Type-Context loss (TC-Loss) can be defined as,

$$\text{TC-Loss} = \text{BCE}(\mathbf{T}(m), \Pr(t \mid m)) \tag{7.9}$$

where BCE is the Binary Cross-Entropy loss,

$$-\sum_{t \in \mathbf{T}(m)} \log \Pr(t \mid m) - \sum_{t \notin \mathbf{T}(m)} \log(1 - \Pr(t \mid m)) \tag{7.10}$$

The entity-type information is also incorporated in the entity representations, by defining a similar Type-Entity loss (TE-Loss).

To identify the gold types $\mathbf{T}(m)$ of a mention $m$, the model makes the distant supervision assumption (same as Ling et al. (2015)) and assigns the types of the gold entity $e^*$ to be the types of the mention. Gold fine-grained types of the entities can be acquired from resources like Freebase (Bollacker et al., 2008) or YAGO (Hoffart et al., 2013).

## 7.4. Training and Inference

This section explains how Xelms generates candidate entities, performs inference, and combines the different training losses.

### *7.4.1. Candidate Generation*

Xelms adopts Tsai and Roth (2016)'s candidate generation strategy with some minor modifications. $\Pr_{\text{prior}}(e \mid m)$ is computed using a surface-title index by counting how often the surface of $m$ links to entity $e$ in Wikipedia hyperlinks. Redirect title surfaces are also treated as hyperlink surfaces and added to the surface to link counts. To generate candidates for mentions in another language (e.g., Chinese), first candidates are generated as described above in the Wikipedia of that language (Chinese Wikipedia), and then mapped

[5] The type vocabulary contains 112 fine-grained types from Ling and Weld (2012) (i.e., $|\Gamma| = 112$).

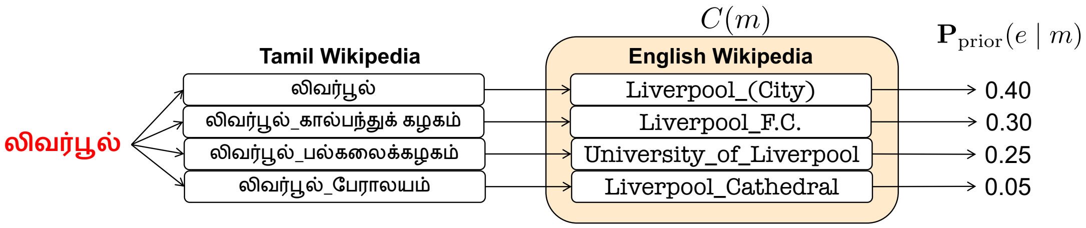


Figure 18: Candidate generation using inter-language links in Wikipedia. Prior probabilities are computed from surface to index statistics computed over target language Wikipedia.

to English titles using inter-language links. To improve recall, counts of a version of the hyperlink surface with its non-ASCII characters replaced (e.g., "Algebre" vs "Algèbre") are also maintained. For Chinese, counts are also kept for a version of the string converted to simplified Chinese. For foreign languages with a Latin script (e.g., Spanish, Turkish), a BACKOFF-TO-ENGLISH strategy is used — if querying the target language surface-to-link index for the mention surface does not generate any candidates, the English surface-to-link index is queried. At most $K$=20 candidates are generated for each mention.

The above candidate generation approach works well for the languages studied in this chapter. In Chapter 8, I will describe a limitation of this approach and an improved candidate generation strategy that uses transliteration.

### *7.4.2. Inference*

The context conditional entity probability $\mathrm{Pr}_{\mathrm{context}}(e \mid m)$ computed in Equation (7.7) and the prior probability $\mathrm{Pr}_{\mathrm{prior}}(e \mid m)$ are combined by taking their union:

$$\begin{aligned}\mathrm{Pr}_{\mathrm{model}}(e \mid m) = \mathrm{Pr}_{\mathrm{prior}}(e \mid m) + \mathrm{Pr}_{\mathrm{context}}(e \mid m) \\ - \mathrm{Pr}_{\mathrm{prior}}(e \mid m) \times \mathrm{Pr}_{\mathrm{context}}(e \mid m)\end{aligned} \tag{7.11}$$

Inference for the mention $m$ picks the entity,

$$\hat{e} = \arg\max_{e \in \mathcal{C}(m)} \mathrm{Pr}_{\mathrm{model}}(e \mid m) \tag{7.12}$$

### 7.4.3. Training Objective

When only training the mention context encoder and entity vectors, the EC-Loss averaged over all training mentions is minimized. When using the two type-aware losses, a weighted sum of EC-Loss, TE-Loss, and TC-Loss is minimized, using the weighing scheme of Kendall et al. (2018),

$$\begin{aligned} &\frac{\text{EC-Loss}}{2\lambda_{\text{EC}}^2} + \frac{\text{TE-Loss}}{2\lambda_{\text{TE}}^2} + \frac{\text{TC-Loss}}{2\lambda_{\text{TC}}^2} \\ &+ \log \lambda_{\text{EC}}^2 + \log \lambda_{\text{TE}}^2 + \log \lambda_{\text{TC}}^2 \end{aligned} \tag{7.13}$$

Here $\lambda_i$ are learnable scalar weighing parameters, and the respective $\frac{1}{2\lambda_i^2}$ and $\log \lambda_i^2$ term ensure that $\lambda_i^2$ does not grow unboundedly. This way, the model learns the relative weight for each loss term.

During training, mentions from different languages are mixed using *inverse-ratio mini-batch mixing* strategy. That is, if two languages have training data sizes proportional to $\alpha : \beta$, at any time during training, mini-batches seen from them are in the ratio $\frac{1}{\alpha} : \frac{1}{\beta}$. This strategy prevents languages with more training data from overwhelming languages with less training data. Though simple, we found this strategy yielded good results.

## 7.5. Experimental Setup

This section describes the training and evaluation datasets, and the previous XEL approaches from the literature used in the experiments for comparison.

**Training Mentions.** Following previous work, hyperlinks from Wikipedia (dumps dated 05/20/2017) are used as the source of grounded mentions for supervision. As described earlier (Section 4.3), Wikipedias in different languages contain articles describing the same entity, which can be resolved by using inter-language links. For instance, article लिवरपूल in the Hindi Wikipedia resolves to `Liverpool` in English. Training mentions statistics are shown in Table 24.

| Lang. | # Train Mentions | Size Relative to # English Mentions |
|---|---|---|
| German (de) | 22.6M | 43.7% |
| Spanish (es) | 13.8M | 26.7% |
| French (fr) | 16.2M | 31.3% |
| Italian (it) | 11.5M | 22.2% |
| Chinese (zh) | 5.9M | 11.4% |
| Arabic (ar) | 3.1M | 6.0% |
| Turkish (tr) | 1.8M | 3.5% |
| Tamil (ta) | 473k | 0.9% |

Table 24: Number of train mentions (from Wikipedia) in each language, with % size relative to English (51.7M mentions). Train mentions from Wikipedias like Arabic, Turkish and Tamil are <10% the size of those from the English Wikipedia.

The evaluation spans 8 languages — German (de), Spanish (es), Italian (it), French (fr), Chinese (zh), Arabic (ar), Turkish (tr) and Tamil (ta), each of which has varying amount of grounded mentions from the respective Wikipedia (Table 24). Note that our method is applicable to any of the 293 Wikipedia languages as a target language.

**Evaluation Datasets.** XELMS is evaluated on the following benchmark datasets, spanning 8 different languages, thus providing an extensive evaluation.

(a) **McN-Test** Dataset from (McNamee et al., 2011). The test set was collected by using parallel document collections, and then crowd-sourcing the ground truths. All the test mentions in this dataset consists of person-names only.

(b) **TH-Test** A subset of the dataset used in (Tsai and Roth, 2016), derived from Wikipedia.[6] The mentions in the dataset fall in two categories — *easy* and *hard*, where hard mentions are those for which the most likely candidate according to the prior probability (i.e., $\arg\max \Pr_{\text{prior}}(e \mid m)$) is *not* the correct title. Indeed, most Wikipedia mentions can be correctly linked by selecting the most likely candi-

[6] Pan et al. (2017) also created a dataset using Wikipedia, but did not categorize mentions like Tsai and Roth (2016). Preliminary experiments on their dataset showed XELMS consistently beat Pan et al. (2017)'s model. TH-TEST was chosen for more controlled experiments.

date (Ratinov et al., 2011). All the hard mentions from Tsai and Roth (2016)'s test splits, henceforth called TH-Test, are used for each language.

(c) **TAC15-Test** Dataset from TAC-KBP 2015 Trilingual Entity Linking Track (Ji et al., 2015) for Chinese and Spanish. It contains 84 news and 82 discussion forum articles in Chinese (total 166 documents) and 84 news and 83 discussion forum articles in Spanish (total 167 documents).

All models are evaluated using *linking accuracy* on gold mentions, which is the fraction of test mentions that are linked to the correct entity. It is assumed gold mentions are provided at test time, following common practice (Tsai and Roth, 2016; Ganea and Hofmann, 2017; Gupta et al., 2017). Table 25 shows the source domains of the evaluation datasets.

| Dataset | Lang. | Source |
|---|---|---|
| TH-Test | de, es, fr, it,<br>zh, ar, tr, ta | Wikipedia |
| McN-Test | de, es, fr, it,<br>zh, ar, tr | News,<br>Parliament Proceedings |
| TAC15-Test | es, zh | News,<br>Discussion Forums |

Table 25: Evaluation datasets used in the cross-lingual entity linking experiments.

**Implementation and Tuning.** All models were implemented using PyTorch.[7] The ADAM (Kingma and Ba, 2014) optimizer was used with a learning rate of 1e-3 in all experiments. the candidate generator was limited to output the top-20 candidates for all experiments. Local context window was set to $W = 25$ tokens. The convolutional filter width was set to $k = 5$. The mention surface vocabulary $V$ was limited to size 1M for both monolingual and joint training. The multilingual embeddings ($d$=300) were scaled to a fixed norm $R$ (=5.0), and were not updated during training. Dropout (Srivastava et al., 2014) was separately applied to local context and document context feature, each being tuned over $\{0.4, 0.45, \cdots, 0.7\}$. The size of entity, type and context vectors was fixed to

[7] `github.com/pytorch`

$h = 100$. Batch size was tuned over $\{128, 256, 512, 1024\}$.

The Wikipedia dumps were parsed using the WikiExtractor script.[8] Stanford segmenter was used for Arabic (Monroe et al., 2014) and Chinese segmentation (Tseng et al., 2005).

Any dataset-specific tuning is avoided by tuning on a development set and applying the same parameters across all datasets. All tunable parameters were tuned on a development set containing the hard mentions from the train split released by Tsai and Roth (2016).

**Comparative Approaches.** We compare against the following state-of-the-art (SoTA) approaches, described with the language from which they use mention contexts in **(.)**,

(a) **Tsai and Roth (2016) (Target Only)** trains a separate model for each language using mention contexts from the target language Wikipedia only. Current SoTA on TH-TEST.

(b) **Pan et al. (2017) (English Only)** uses entity coherence statistics from English Wikipedia and the document context of a mention for XEL. Current SoTA on MCN-TEST, except for Italian and Turkish, for which it's McNamee et al. (2011).

(c) **Sil et al. (2018) (English Only)** uses multilingual embeddings to transfer a pre-trained English entity linking model to perform XEL for Spanish and Chinese. Prior probabilities $\text{Pr}_{\text{prior}}$ are used as a feature. Current SoTA on TAC15-TEST.

## 7.6. Experiments

The experiments in this section aim to evaluate: **(a)** if XELMS can train a better entity linking model for a target language, by exploiting additional data from a high resource language like English? (Section 7.6.1) **(b)** how does a *single* XEL model (trained using XELMS) for multiple related languages compare to individually trained models for each language? (Section 7.6.2) **(c)** if adding additional type information through a multi-tasking

[8] `github.com/attardi/wikiextractor`

loss to XELMS improves performance? (Section 7.6.3)

In all experiments, the linking accuracy of XELMS is reported, averaged over 5 different runs, and marked with * the statistical significance ($p < 0.01$) of the best result (shown **bold**) against the state-of-the-art (SoTA) using Student's one-sample t-test.

### *7.6.1. Monolingual and Joint Models*

The first experiment compares XELMS(mono), which uses monolingual supervision in the target language only, and XELMS(joint), which uses supervision from English in addition to the monolingual supervision, against the state-of-the-art (SoTA) approaches. The results are in Table 26 and 27 respectively.

It can be seen that XELMS(mono) achieves similar or better scores than respective SoTA on all datasets. The SoTA for McN-TEST in Turkish and Chinese enhances the model by using transliteration for candidate generation, explaining their superior performance. XELMS(joint) performs substantially better than XELMS(mono) on all datasets, proving that using additional supervision from a high resource language like English leads to better linking performance. In particular, XELMS(joint) outperforms the SoTA on all languages in TH-TEST, on Spanish in TAC15-Test, and on 4 of the 7 languages in McN-TEST.

### *7.6.2. Multilingual Training*

XELMS is the first approach that can train a *single* XEL model for multiple languages. To demonstrate this capability, a model, henceforth referred as XELMS(multi), is trained *jointly* on 5 related languages — Spanish, German, French, Italian and English. XELMS(multi) is compared to the respective XELMS(joint) model for each language. The results are in Table 27 and 28.

The results show that XELMS(multi) is better (or at par) than XELMS(joint) on all datasets. This suggests that XELMS(multi) can make more efficient use of available supervision in related languages than previous approaches that trained separate models per language.

| **Dataset** → | TH-Test | | | McN-Test | | |
|---|---|---|---|---|---|---|
| **Lang.** ↓ | SoTA | Xelms | | SoTA | Xelms | |
| | | mono | joint | | mono | joint |
| de | 53.3 | 53.7 | **55.6**$^*$ | 89.7 | 90.9 | **91.5** |
| es | 54.5 | 54.9 | **56.6**$^*$ | **91.5** | 91.2 | 91.4 |
| fr | 47.5 | 48.5 | **49.9**$^*$ | 92.1 | 92.6 | **92.7** |
| it | 48.3 | 48.4 | **51.9**$^*$ | 85.9 | 87.0 | **87.8**$^*$ |
| zh | 57.6 | 58.1 | **61.3**$^*$ | **91.2**$^*$ | 87.4 | 88.2 |
| ar | 62.1 | 62.6 | **63.8**$^*$ | 80.2 | 80.3 | **83.1**$^*$ |
| tr | 60.2 | 61.0 | **61.7**$^*$ | **95.3**$^*$ | 91.0 | 91.9 |
| ta | 54.1 | 54.7 | **59.7**$^*$ | n/a | n/a | n/a |
| avg. | 54.7 | 55.2 | **57.6** | 89.4 | 88.6 | **89.5** |

Table 26: Xelms(joint) improves upon Xelms(mono) and the current State-of-The-Art (SoTA) on TH-Test and McN-Test, showing the benefit of using additional supervision from English. The best score is shown **bold** and $^*$ marks statistical significance of best against SoTA. Refer Section 7.5 for details on SoTA.

### *7.6.3. Adding Fine-grained Type Information*

In this experiment, the effect of adding fine-grained type information is studied by comparing Xelms(mono) and Xelms(joint) to Xelms(mono$^{+\text{type}}$) and Xelms(joint$^{+\text{type}}$) respectively, that are versions of Xelms(mono) and Xelms(joint) trained with the type-aware losses. The results are in Table 27.

Xelms(mono$^{+\text{type}}$) and Xelms(joint$^{+\text{type}}$) both improve compared to Xelms(mono) and Xelms(joint) on McN-Test and TH-Test (compare Table 29 to Table 26), showing the benefit of using structured knowledge in the form of fine-grained types. Similar trends are also seen on TAC15-Test (Table 27), where Xelms(joint$^{+\text{type}}$) improves on the SoTA.

## 7.7. Experiments with Limited Resources

One of the motivations of Xelms is to exploit supervision from high-resource languages like English to aid XEL for languages with limited resources. Two such scenarios are examined in this section,

| | Lang. → <br> Model ↓ | es | zh |
|---|---|---|---|
| | (Tsai and Roth, 2016) | 82.4 | 85.1 |
| | (Sil et al., 2018) (SoTA) | 83.9 | 85.9 |
| Xelms | mono | 83.3 | 84.4 |
| | mono$^{+\text{type}}$ | 83.5 | 84.8 |
| | joint | 84.1 | 85.5 |
| | joint$^{+\text{type}}$ | **84.4**$^*$ | **86.0** |
| | multi | 83.9 | n/a |
| | multi$^{+\text{type}}$ | **84.4**$^*$ | n/a |

Table 27: Linking accuracy on TAC15-Test. Numbers for Sil et al. (2018) from personal communication.

| **Dataset →** | TH-Test | | | McN-Test | | |
|---|---|---|---|---|---|---|
| | | Xelms | | | Xelms | |
| **Lang. ↓** | SoTA | joint | multi | SoTA | joint | multi |
| de | 53.3 | **55.6**$^*$ | 55.2 | 89.7 | **91.5** | 91.4 |
| es | 54.5 | 56.6 | **56.8**$^*$ | **91.5** | 91.4 | 91.4 |
| fr | 47.5 | 49.9 | **51.0**$^*$ | 92.1 | **92.7** | 92.6 |
| it | 48.3 | 51.9 | **52.3**$^*$ | 85.9 | 87.8 | **87.9**$^*$ |
| avg. | 50.9 | 53.5 | **53.8** | 89.8 | **90.8** | **90.8** |

Table 28: Linking accuracy of a *single* Xelms(multi) model for four languages — German, Spanish, French and Italian. For comparison, individually trained Xelms(joint) scores are also shown. The best score is shown **bold** and $^*$ marks statistical significance of **best** against SoTA. Refer Section 7.5 for details on SoTA.

**(a)** *Zero-shot setting*, that is, no supervision available in the target language. Analysis in Section 7.7.1 reveals the limitations of zero-shot XEL approaches and finds that the prior probabilities play an important role in achieving good performance, which are unavailable in realistic zero-shot scenarios.

**(b)** *Low-resource setting*, that is, some supervision available in the target language. Section 7.7.2 shows that by combining supervision from a high-resource language, Xelms can achieve competitive performance with a fraction of available supervision in the target language.

| **Dataset** → | TH-Test | | | McN-Test | | |
|---|---|---|---|---|---|---|
| **Lang.** ↓ | SoTA | Xelms | | SoTA | Xelms | |
| | | mono$^{+\text{type}}$ | joint$^{+\text{type}}$ | | mono$^{+\text{type}}$ | joint$^{+\text{type}}$ |
| de | 53.3 | 54.0 | **55.9**$^*$ | 89.7 | 91.2 | **91.5** |
| es | 54.5 | 55.1 | **57.2**$^*$ | **91.5** | 91.0 | 91.2 |
| fr | 47.5 | 49.0 | **50.6**$^*$ | 92.1 | 92.6 | **92.7** |
| it | 48.3 | 49.2 | **52.2**$^*$ | 85.9 | 87.4 | **87.9**$^*$ |
| zh | 57.6 | 58.9 | **61.5**$^*$ | **91.2**$^*$ | 87.6 | 88.4 |
| ar | 62.1 | 63.0 | **64.0**$^*$ | 80.2 | 81.1 | **84.0**$^*$ |
| tr | 60.2 | 61.5 | **62.0**$^*$ | **95.3**$^*$ | 91.2 | 92.1 |
| ta | 54.1 | 56.0 | **59.9**$^*$ | n/a | n/a | n/a |
| avg. | 54.7 | 55.8 | **57.9** | 89.4 | 88.9 | **89.7** |

Table 29: Adding fine-grained type information further improves linking accuracy (compare to Table 26). The best score is shown **bold** and $^*$ marks statistical significance of best against SoTA. Refer Section 7.5 for details on SoTA.

### *7.7.1. Zero-shot Setting*

We first explain how Xelms can perform zero-shot XEL, the implications of our zero-shot setting, and how it is more realistic than previous work.

**Zero-shot XEL with XELMS.** Xelms performs zero-shot XEL by training a model using English supervision and multilingual embeddings for English, and directly applying it to the test data in another language using the respective multilingual word embedding instead of English embeddings.

**No Prior Probabilities.** Prior probabilities (or prior), i.e., $\text{Pr}_{\text{prior}}$ have been shown to be a reliable indicator of the correct disambiguation in entity linking (Ratinov et al., 2011; Tsai and Roth, 2016). These probabilities are estimated from counts over the training mentions in the target language. In the absence of training data for the target language, as in the zero-shot setting, these prior probabilities are not available to an XEL model.

**Comparison to Previous Work.** The only other model capable of zero-shot XEL is that of Sil et al. (2018). However, Sil et al. (2018) use prior probabilities and coreference

| **Dataset →** | TAC15-Test | | TH-Test | McN-Test |
|---|---|---|---|---|
| **Approach ↓** | (es) | (zh) | (avg) | (avg) |
| Xelms (Z-S w/ prior) | 80.3 | 83.9 | 43.5 | 88.1 |
| Xelms (Z-S w/o prior) | 53.5 | 55.9 | 41.1 | 86.0 |
| SoTA | 83.9 | 85.9 | 54.7 | 89.4 |

Table 30: Linking accuracy of the zero-shot (Z-S) approach on different datasets. Zero-shot (w/ prior) is close to SoTA for datasets like TAC15-Test, but performance drops in the more realistic setting of zero-shot (w/o prior) (Section 7.7.1) on all datasets, indicating most of the performance can be attributed to the presence of prior probabilities. The slight drop in McN-Test is due to trivial mentions that only have a single candidate.

chains for the target language in their zero-shot experiments, both of which will not be available in a realistic zero-shot scenario. Compared to Sil et al. (2018), we evaluate the performance of zero-shot XEL in more realistic setting, and show it is adversely affected by absence of prior probabilities.

**Is zero-shot XEL really effective?** To evaluate the effectiveness of the zero-shot XEL approach, we perform zero-shot XEL using Xelms on all datasets. Table 30 shows zero-shot XEL results on all datasets, both with and without using the prior during inference. Note that zero-shot XEL (with prior) is close to SoTA (Sil et al. (2018)) on TAC15-Test, that also uses the prior for zero-shot XEL. However, for zero-shot XEL (without prior), performance drops by more than 20% for TAC15-Test, 2.4% for TH-Test and by 2.1% for McN-Test. This drop indicates that zero-shot XEL is not effective in a realistic zero-shot setting (i.e., when the prior is unavailable for inference).

The prior is indeed a strong indicator of the correct disambiguation. For instance, simply selecting the most likely candidate using the prior achieved 77.2% and 78.8% for Spanish and Chinese respectively in the TAC15-Test dataset. Note that both versions of zero-shot XEL (with and without prior) perform worse than the best possible model on TH-Test, because TH-Test was constructed to ensure prior probabilities are not strong indicators (Tsai and Roth, 2016). On McN-Test, an average of 75.9% mentions have only one (the correct) candidate, making them trivial to link, regardless of the absence of priors.

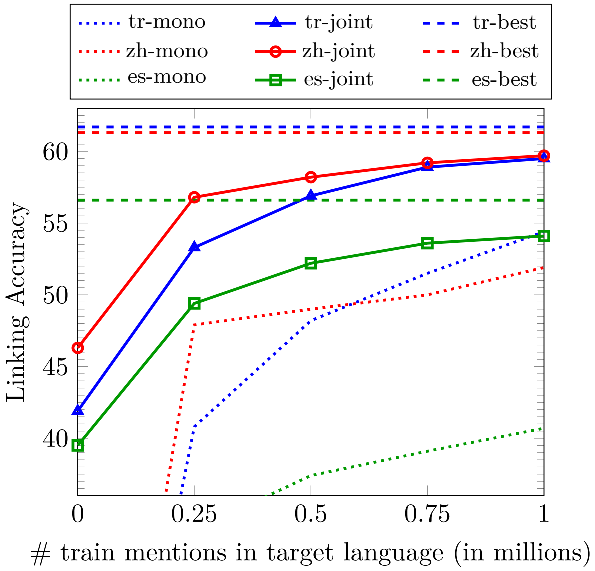


Figure 19: Linking accuracy v/s the number of train mentions in the target language L (= Turkish (tr), Chinese (zh) and Spanish (es)). We compare both XELMS(mono) and XELMS(joint) to the best results using all available supervision, denoted by L-best. To discount the effect of the prior, all results above are without it. For number of train mentions = 0, XELMS(joint) is equivalent to zero-shot without prior. Best viewed in color.

The results show that most of the XEL performance in zero-shot settings can be attributed to availability of prior probabilities for the candidates. It is evident that zero-shot XEL in a realistic setting (i.e., when prior probabilities are not available) is still challenging.

### *7.7.2. Low-resource Setting*

This set of experiments analyzes the behavior of XELMS in a low-resource setting, i.e., when some supervision is available in the target language. The aim of this setting is to estimate how much supervision from the target language is needed to get reasonable performance when using it jointly with supervision from English.

For this experiment, a XELMS(joint) model is trained by gradually increasing the number of mention contexts for target language L (= Spanish, Chinese and Turkish) that are available for supervision. Figure 19 plots the results on the TH-Test dataset. Figure 19 also shows the best results achieved using all available target language supervision (denoted by L-best). For comparison with the mono-lingually supervised model, the performance of XELMS(mono), which only uses the target language supervision, is also plotted. To discount the effect of prior probabilities, all results reported here are without the prior.

Figure 19 shows that after training on 0.75M mentions from Turkish and Chinese (and 1.0M mentions from Spanish), the Xelms(joint) model is within 2-3% of the respective L-best model that uses all training mentions in the target language. This indicates that Xelms(joint) can reach competitive performance even with a fraction of the full target language supervision. For comparison, a Xelms(mono) model that was trained on the same number of training mentions is 5-10% behind the respective Xelms(joint) model, showing better utilization of target language supervision by Xelms(joint).

## 7.8. Summary

In this chapter, I described an approach that can combine supervision from multiple languages to train an XEL model. Key to this approach were multilingual embeddings, that were the feature space upon which language agnostic mention representations were built for the XEL task. The results in the Section 7.6 illustrated the benefits of this shared representation space through extensive evaluation on different benchmarks. Another benefit of our approach is that can train a *single* model for multiple related languages, making more efficient use of available supervision than previous approaches that trained separate models.

Further improvements for XEL can be made using a joint inference framework that enforces coherent predictions (Cheng and Roth, 2013; Globerson et al., 2016; Ganea and Hofmann, 2017), or by combining supervision between languages in a more principled way (e.g., choice of related languages). Similar techniques as ours can be applied to other information extraction tasks like relation extraction to extend them to multilingual settings. An assumption made in this chapter was that the entity mentions are already identified for the EL system. That is, the EL system is only responsible for disambiguation. This assumption may lead to un-recoverable errors due to incorrectly detected mention spans. Several monolingual entity linking systems have shown that this issue can be remedied by performing these two steps *jointly* (Sil and Yates, 2013; Nguyen et al., 2016a; Kolitsas et al., 2018). Extending cross-lingual entity linking systems in a similar manner, such that they can directly operate on raw text by jointly detecting and disambiguating the mentions, is also a useful direction to pursue. Another key challenge for all XEL approaches is the task of candidate generation, that is currently limited by existence of a target language Wikipedia. I explore this problem in detail in the next chapter.

# CHAPTER 8 : Bootstrapping Transliteration with Constrained Discovery

## 8.1. Introduction

The performance of an entity linking system is a function of the quality of candidate generation, as poor candidate generation means that the correct candidate might not be in the list of candidates. The previous chapter discussed how one of the challenges of the cross-lingual entity linking problem is generating candidates for mentions in a language that has a different writing script than English. The candidate generation approach in the previous chapter dealt with this challenge by using inter-language links between the target language Wikipedia and the English Wikipedia. However, this approach has an obvious coverage issue when the inter-language link does not exist between some articles.[1]

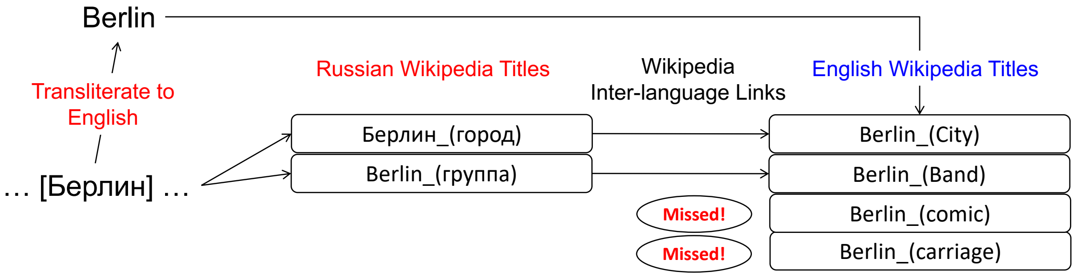


Figure 20: Limitation of the candidate generation approach described in Figure 18. For the Russian mention (that transliterates to *Berlin*), using inter-language links misses two plausible candidate entities in the English Wikipedia.

For example, any approach that relies on Wikipedia inter-language links will not generate all plausible candidates for the Russian mention in Figure 20 (which transliterates to Berlin), because of missing inter-language links for entities like `Berlin_(comic)` and `Berlin_(carriage)`. This problem can be exacerbated for low-resource languages, for which the corresponding Wikipedia contains few articles (and thus few inter-language links). Figure 20 also shows how by first *transliterating* the mention surface (i.e., rewriting the string in a different writing script) into English, this issue can be avoided.

[1] This can happen either because the corresponding article does not exist in the target language Wikipedia or because the article exist, but the inter-language link was missing in the available meta-data.

Existing approaches for generating transliterations require thousands of name pairs (i.e., source and its transliteration) as supervision, which may not be suitable for low-resource scenarios. *How can one learn to transliterate when only a few hundred name pairs are available as supervision?* This chapter proposes a new bootstrapping algorithm to iteratively improve a weak transliteration generation model trained on a few hundred name pairs. The approach presented in this chapter marries two threads of work in the literature — transliteration *generation* and transliteration *discovery*, which I describe below.

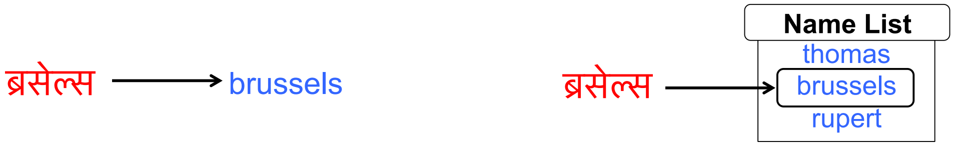
## 8.2. History of the Transliteration Task

*Transliteration* is the process of transducing a name from one writing system to another (e.g., ओबामा in Devanagari to *Obama* in Latin script) while preserving its pronunciation (Knight and Graehl, 1998; Karimi et al., 2011).

Historically, the transliteration task emerged as a component of machine translation systems, to handle the translation of names and technical terms across languages. Such lexical items will have poor coverage in a bilingual dictionary or phrase translation table, thus making it necessary to replace them with their (approximate) phonetic equivalent (Knight and Graehl, 1998). However, transliteration has utility beyond as a sub-routine in machine translation. For instance, transliterating names from foreign languages to English[2] helps in multilingual knowledge acquisition tasks like named entity recognition (Darwish, 2013) and information retrieval (Virga and Khudanpur, 2003; Jaleel and Larkey, 2003).

Two tasks feature prominently in the transliteration literature: *generation* (Knight and Graehl, 1998) that involves producing an appropriate transliteration for a given word in an open-ended way, and *discovery* (Sproat et al., 2006; Klementiev and Roth, 2008) that involves selecting an appropriate transliteration for a word from a list of candidates. This section briefly review the limitations of existing generation and discovery approaches, and provides an overview of how our work addresses them.

[2] Also referred to as *back-transliteration*.



(a) Generation

(b) Discovery

Figure 21: Generation treats transliteration as a sequence transduction task, while discovery aims to select the correct transliteration from a given list of names.

### *8.2.1. Transliteration Generation*

Generation aims to transliterate an input word $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ in the source writing script without relying on a translation lexicon, by generating a sequence of characters $\boldsymbol{y} = (y_1, y_2, \cdots, y_m)$ in the target writing script (Haizhou et al., 2004; Jiampojamarn et al., 2009; Ravi and Knight, 2009; Jiampojamarn et al., 2010; Finch et al., 2015, inter alia). Figure 21a shows how a generation model will process the Hindi string ब्रसेल्स.

Transliteration generation is often formulated as a *machine translation (MT) problem*, by treating the characters $x_i$ in the input word as "words", the input word $\boldsymbol{x}$ as the "sentence", and the target output $\boldsymbol{y}$ as the "translation" (Virga and Khudanpur, 2003; Irvine et al., 2010). This allows one to use available MT toolkits such as Moses (Koehn et al., 2007) or Joshua (Li et al., 2009) for training a transliteration system. However, these approaches may not exploit structure that is specific to the transliteration task.

Other approaches treat generation as a *sequence labeling problem* (Ganesh et al., 2008; Reddy and Waxmonsky, 2009; Ammar et al., 2012). Each character $x_i$ in the input is assigned a label $t_i = y_p, \cdots, y_q$ which is a sequence of one or more characters from the target writing script. For instance, the Hindi character थ can be labeled as "tha" when transliterating Hindi→English. One issue with this formulation is that the size of output label space can be quite large, depending on how many different spans $y_p, \cdots, y_q$ are seen in the training data. For instance, Ammar et al. (2012) reported that Arabic → English transliteration had $>$ 1200 labels, while Thai → English had $>$ 1700 labels.

Both the above formulations require generous amount of name pairs ($\approx$ 5–10k) as supervision. One popular way to obtain this supervision is identifying name pairs using interlanguage links in Wikipedia (Irvine et al., 2010; Tsai and Roth, 2018). However, a truly low-resource language (like Tigrinya) is likely to have limited Wikipedia presence as well, and thus low resource transliteration remains a challenging problem.

### *8.2.2. Transliteration Discovery*

On the other hand, the task of transliteration discovery deals with selecting the correct transliteration $\boldsymbol{y}$ for a word $\boldsymbol{x}$ from a relatively small list of candidates $\mathcal{N}$. The presence of the name list makes discovery essentially a *ranking* problem — rank all names in $\mathcal{N}$ to identify the most appropriate transliteration for a given word in the source script. Figure 21b shows how a discovery model will process the Hindi string ब्रसेल्स.

Discovery is a considerably easier task than generation, owing to the restricted search space. Indeed, for $|\mathcal{N}| \sim 50k$, unsupervised discovery approaches that use constraints derived from romanization tables have been shown to be successful (Chang et al., 2009). Notice that for a given input $\boldsymbol{x}$, many names in $\mathcal{N}$ can be filtered out quite easily, so the effective search space is even smaller. In general, the hardness of discovery problem increases with $|\mathcal{N}|$, say, when $\mathcal{N}$ contains millions of names.

A key limitation of discovery is the assumption that the correct transliteration(s) is in the list of candidates $\mathcal{N}$. But it is unlikely that $\mathcal{N}$ will be exhaustive, as new names are constantly being introduced in language.[3] If no correct transliteration is present in $\mathcal{N}$, a discovery model will end up selecting *some* name from $\mathcal{N}$, and produce false positives. Of course, one can partially remedy this issue by using a large list of candidates $\mathcal{N}$, at the cost of increased difficulty. Another issue with transliteration discovery approaches is that they often exploit features derived from resources that are unlikely to be available for low-resource languages, like temporally aligned comparable corpora (Sproat et al., 2006; Klementiev and

[3] The honorific *Khaleesi* from the TV series “Game of Thrones” became a popular baby name in 2016.

Roth, 2008). To overcome these limitations, it is prudent to develop generation models that can handle input for which the transliteration does not belong in $\mathcal{N}$, and operate in low-resource scenarios.

The work described in this chapter develops transliteration generation approaches for low-resource languages, by using constrained discovery to drive the learning.

### *8.2.3. Our Work*

Existing generation models require supervision in the form of source-target name pairs ($\approx$ 5–10k) that are often collected from names in Wikipedia inter-language links (Irvine et al., 2010). However, most languages that use non-Latin scripts are under-represented in terms of such resources. Table 31 illustrates this issue. A generation model that requires 50k name pairs as supervision can only support 6 languages, while one that needs 500 could support 56. For a model to be widely applicable, it must function in low-resource settings.

| # Name Pairs in Wikipedia | | Languages | Scripts |
|---|---|---|---|
| $> 50,000$ | | 6 | 5 |
| $> 10,000$ | | 18 | 14 |
| $> 5,000$ | *Previous Work* | 24 | 15 |
| $> 1,000$ | | 45 | 22 |
| $> 500$ | *Our Approach* | 56 | 23 |
| $> 0$ | | 93 | 30 |

Table 31: Cumulative number of person name pairs in Wikipedia inter-language links. For instance, 45 languages (that cover 22 scripts) have at least 1000 name pairs that can be extracted from inter-language links. While previous approaches for transliteration generation were applicable to only 24 languages (spanning 15 scripts), our approach is applicable to 56 languages (23 scripts). When counting scripts we exclude variants (e.g., all Cyrillic scripts and variants count as one).

We show that a weak generation model can be iteratively improved using constrained discovery. In particular, our work uses a weak generation model to discover new training pairs, using constraints to drive the bootstrapping. Table 31 shows the extra coverage one can achieve by extending to low-resource languages using our approach. The practicality of our approach is demonstrated in truly low-resource scenarios and downstream applications through two case studies. First, experiments in Section 8.8.1 shows that one can obtain the initial supervision from a *single* human annotator within a few hours for two languages — Armenian and Punjabi. This is a realistic scenario where language access is limited to a single native informant. Second, experiments in Section 8.8.2 shows that our approach benefits a typical downstream application, namely candidate generation for cross-lingual entity linking, by improving recall on two low-resource languages — Tigrinya and Macedonian. To appreciate the difficulty of the transliteration task, an analysis of the inherent challenges of transliteration, and the trade-off between native (i.e., source) and foreign (i.e., target) vocabulary is also presented in Section 8.7.

The generation model presented in this chapter is inspired by the success of sequence to sequence generation models (Sutskever et al., 2014; Bahdanau et al., 2015) particularly variants used for string transduction tasks like morphological inflection and derivation generation (Faruqui et al., 2016; Cotterell et al., 2017; Aharoni and Goldberg, 2017; Makarov et al., 2017). The bootstrapping framework can be viewed as an instance of constraint-driven learning (Chang et al., 2007, 2012).

## 8.3. Transliteration with Hard Monotonic Attention

We view generation as a string transduction task and use a sequence to sequence (Seq2Seq) generation model that uses *hard monotonic attention* (Aharoni and Goldberg, 2017), henceforth referred to as Seq2Seq(HMA). During generation, Seq2Seq(HMA) directly models the *monotonic* source-to-target sequence alignments, using a pointer that attends to a *single* input character at a time. Monotonic attention is a natural fit for transliteration because even though the number of characters needed to represent a sound in the source and target language vary, the sequence of sounds is presented in the same order.[4] We review Seq2Seq(HMA) below, and describe how it can be applied to transliteration generation.

[4] Many Indic scripts, that sometimes write vowels before the consonants they are pronounced after, seem to violate this claim, but their Unicode representations actually preserve the consonant-vowel order.

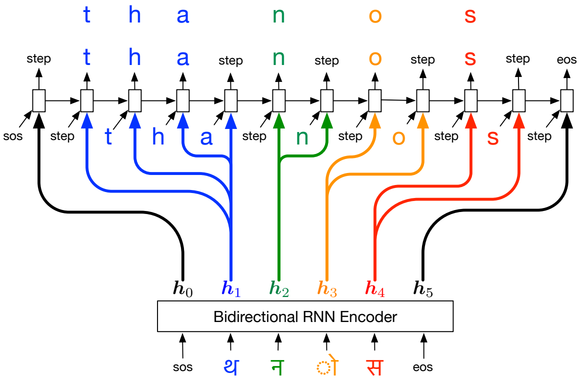


Figure 22: Transliteration using Seq2Seq transduction with Hard Monotonic Attention, or Seq2Seq(HMA). The figure shows how decoding proceeds for transliterating थनोस to *thanos*. During decoding, the model attends to a source character (e.g.,थ shown in blue) and outputs target characters (*t*, *h*, *a*) until a *step* action is generated, which moves the attention position forward by one character (to न), and so on.

**Encoding the Input Word.** Let $\Sigma_f$ be the source alphabet and $\Sigma_e$ be the English alphabet. Let $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ denote an input word where each character $x_i \in \Sigma_f$. The characters are first encoded using an embedding matrix $\mathbf{W} \in \mathbb{R}^{|\Sigma_f| \times d}$ to get character embeddings $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_n$ where each $\mathbf{w}_i \in \mathbb{R}^d$. These embeddings are fed into a bidirectional RNN encoder (Figure 22) to generate encoded vectors $\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_n$ where each $\boldsymbol{h}_i \in \mathbb{R}^{2k}$, and $k$ is the size of output vector of the forward (and backward) encoder. The encoded vectors $\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_n$ are then fed into the decoder.

**Monotonic Decoding with Hard Attention.** Figure 22 illustrates the decoding process. The decoder RNN generates a sequence of actions $\{s_1, s_2, \cdots\}$, such that each $s_i \in \Sigma_e \cup \{\textit{step}\}$. The *step* action controls an attention position $a$, attending on input character $x_a$, with encoded vector $\boldsymbol{h}_a$. Each action $s_i$ is embedded into $\boldsymbol{s}_i \in \mathbb{R}^d$ using an output embedding matrix $\mathbf{A} \in \mathbb{R}^{(|\Sigma_e|+1) \times d}$. At any time during decoding, the decoder uses its last hidden state, the embedding of the previous action $\boldsymbol{s}_i$ and the encoded vector $\boldsymbol{h}_a$ of the current attended position to generate the next action $s_{i+1}$. If the generated action is *step*, the decoder increments the attention position by one. This ensures that the decoding is monotonic, as the attention position can only move forward or stay at the same position during generation. We use $\texttt{Inference}(G, \boldsymbol{x})$ to refer to the above decoding process for a trained generation model $G$ and input word $\boldsymbol{x}$.

**Training.** Training requires the oracle action sequence $\{s_i\}$ for input $\boldsymbol{x}_{1:n}$ that generates the correct transliteration $\boldsymbol{y}_{1:m}$. The oracle sequence is generated using the train name pairs and Algorithm 1 in Aharoni and Goldberg (2017), with the character-level alignment between $\boldsymbol{x}_{1:n}$ and $\boldsymbol{y}_{1:m}$ being generated using the algorithm in Cotterell et al. (2016).

**Inference Strategies.** An unconstrained and a constrained inference strategy can be used to select the best transliteration $\hat{\boldsymbol{y}}$ from a beam $\{\boldsymbol{y}_i\}_{i=1}^{k}$ of transliteration hypotheses, sorted in descending order by likelihood. The constrained strategy uses a name dictionary $\mathcal{N}$, to guide the inference. These strategies are applicable to any generation model.

- **Unconstrained (U)** selects the most likely item $\boldsymbol{y}_1$ in the beam as $\hat{\boldsymbol{y}}$.
- **Dictionary-Constrained (DC)** selects the highest scoring hypothesis that is present in $\mathcal{N}$, and defaults to $\boldsymbol{y}_1$ if none are in $\mathcal{N}$.

It is tempting to disallow the model from generating hypotheses that are not in the dictionary $\mathcal{N}$. However, dictionaries are always incomplete, and restricting the search to generate from $\mathcal{N}$ inevitably leads to incorrect predictions if the correct transliteration is not in $\mathcal{N}$. This is essentially the same as the problem inherent to discovery models.

**Other Strategies in Previous Work.** A related constrained inference strategy was proposed by Lin et al. (2016), who use an entity linking system (Wang et al., 2015) to correct and re-rank hypotheses. Our constrained inference strategy is much simpler, requiring only a name dictionary $\mathcal{N}$. These approaches are compared in the experiments.

## 8.4. Low-Resource Bootstrapping

Low-resource languages will have a limited number of name pairs for training a generation model. This section proposes a new bootstrapping algorithm to learn a good generation model in this setting, that uses *constrained discovery* to mine name pairs to re-train the generation model. The algorithm requires a small ($\approx 500$) seed list of name pairs $\mathcal{S}$ for supervision, a dictionary $\mathcal{N}$ containing names in English, and a list of words $\mathcal{V}_f$ in the foreign script. The algorithm and the constraints used in discovery are described below.

### *8.4.1. The Bootstrapping Algorithm*

Algorithm 3 shows the code of the bootstrapping procedure. First, a weak generation model $G_0$ is initialized using a seed list of name pairs $\mathcal{S}$ (line 1). At iteration $t$, the current generation model $G_t$ produces the top-$k$ transliteration hypotheses $\{\boldsymbol{y}_i\}_{i=1}^{k}$ for each word $\boldsymbol{x} \in \mathcal{V}_f$ (line 5). A source word and hypothesis pair $(\boldsymbol{x}, \boldsymbol{y}_i)$, is added to the set of mined name pairs $\mathcal{B}$ if they satisfy a set of discovery constraints (described below) (line 8). A new generation model $G_{t+1}$ is trained for the next iteration using the union of the seed list $\mathcal{S}$ and the mined name pairs $\mathcal{B}$ (line 12). $\mathcal{B}$ is purged after every iteration (line 3) to prevent $G_{t+1}$ from being influenced by possibly incorrect name pairs mined in earlier iterations. The algorithm converges when accuracy@1 stops increasing on a development set. Note that our bootstrapping approach is applicable to any transliteration generation model.

To ensure that high quality name pairs are added to the mined set $\mathcal{B}$ during bootstrapping, the following discovery constraints are used.

### *8.4.2. Discovery Constraints*

A word-transliteration pair $(\boldsymbol{x}, \boldsymbol{y})$ is added to the set of mined pairs $\mathcal{B}$, only if all the following constraints are satisfied,

1. $\boldsymbol{y} \in \mathcal{N}$. i.e., $\boldsymbol{y}$ is a known name in the dictionary.

2. $\mathrm{P}(\boldsymbol{y} \mid \boldsymbol{x}) > \delta^{min}$. The model is sufficiently confident about the transliteration.

3. The ratio of lengths $\frac{|\boldsymbol{y}|}{|\boldsymbol{x}|}$ should be close to the average ratio estimated from $\mathcal{S}$ (Matthews, 2007). This is encoded using the constraint $|\frac{|\boldsymbol{y}|}{|\boldsymbol{x}|} - r(\mathcal{S})| \leq \epsilon$, where $\epsilon$ is a tunable tolerance and $r(\mathcal{S})$ is the average ratio in $\mathcal{S}$.

**Algorithm 3** Bootstrapping Transliteration Generation via Constrained Discovery

```
Input:
    English name dictionary $\mathcal{N}$; Seed training pairs $\mathcal{S}$;
    Vocabulary in the target language $\mathcal{V}_f$.
Hyper-parameters:
    initial minimum length threshold $L_0^{min}$;
    minimum likelihood threshold $\delta^{min}$;
    length ratio tolerance $\epsilon$.
Output: Generation model $G_T$
$G_0 = \texttt{train}(\mathcal{S})$                              ▷ Initialize a weak generation model.
while not converged do
    $\mathcal{B} = \emptyset$                                     ▷ Purge mined set.
    for $\boldsymbol{x}$ in $\mathcal{V}_f$ do
        $\{\boldsymbol{y}_i\}_{i=1}^{k} = \text{argtop}_k\ \texttt{Inference}(G_t, \boldsymbol{x})$    ▷ Generate top-k hypotheses.
        for $\boldsymbol{y}_i$ in $\{\boldsymbol{y}_i\}_{i=1}^{k}$ do
            if $(\boldsymbol{x}, \boldsymbol{y}_i)$ satisfies constraints in Section 8.4.2 then
                $\mathcal{B} = \mathcal{B} \cup \{(\boldsymbol{x}, \boldsymbol{y}_i)\}$        ▷ Add name pair to mined set.
            end if
        end for
    end for
    $G_{t+1} = \texttt{train}\ (\mathcal{S} \cup \mathcal{B})$
    $L_{t+1}^{min} = L_t^{min} - 1$                               ▷ Reduce length threshold.
    $t = t + 1$                                                   ▷ Track iteration.
end while
```

4. $|\boldsymbol{y}| > L_t^{min}$. It was found that false positives were more likely to be short hypotheses in early iterations. As the model improves with each iteration, $L_t^{min}$ is lowered to allow more new pairs to be mined.

Notice that the bootstrapping algorithm can be formulated as an instance of constraint driven learning (Chang et al., 2007, 2012).

## 8.5. Experimental Setup

All generation models are evaluated using accuracy at 1 (acc@1), by comparing the best model prediction $\hat{\boldsymbol{y}}$ against the reference transliteration $\boldsymbol{y}^*$, unless otherwise specified.

**Training and Evaluation.** The train and development sets from the Named Entities Workshop 2015 (Duan et al., 2015) (NEWS2015) for Hindi (hi), Kannada (kn), Bengali (bn), Tamil (ta) and Hebrew (he) are used as our train and evaluation set.[5] The size of the train set was ∼12k, 10k, 14k, 10k and 10k respectively, and all evaluation sets were ∼1k.

For the low-resource experiments, 500 examples are sub-sampled from each train set in the NEWS2015 dataset using five random seeds, and the averaged results on these is reported. 1k name pairs from the corresponding NEWS2015 train set of each language is set aside as development data. The foreign script portion of the remaining train data is used as $\mathcal{V}_f$ in the bootstrapping algorithm.

**Model and Tuning Details.** Seq2Seq(HMA) is implemented using PyTorch.[6] 50 dimensional character embeddings, and single layer GRU (Cho et al., 2014) encoder with 20 hidden states are used for all experiments. The Adam (Kingma and Ba, 2014) optimizer was used with default hyper-parameters, a learning rate of 0.001, a batch size of 1, and maximum of 20 iterations in all experiments. Beam search used a width of 10. For low-resource experiments, all bootstrapping parameters were tuned on the development data set aside above. $L_0^{min}$ is chosen from $\{10, 15, 20, 25\}$.

**Name Dictionary $\mathcal{N}$.** A name dictionary of 1.05 million names is constructed from the English Wikipedia (dump dated 05/20/2017) by taking the list of title tokens in Wikipedia sorted by frequency, and removing tokens that appears only once. The same dictionary is used in all experiments.

**Comparisons.** The following generation models are used for comparison:

(a) **P&R (Pasternack and Roth, 2009)** A probabilistic transliteration generation approach that learns latent alignments between sub-strings in the source and the target words. The model is trained to score all possible segmentation and their alignments, using an EM-like algorithm.

[5] Test set was not available since shared task concluded.

[6] github.com/pytorch

(b) **DirecTL+ (Jiampojamarn et al., 2009)** A HMM-like discriminative string transduction model that predicts the output transliteration using many-to-many alignments between the source word and target transliteration. Following Jiampojamarn et al. (2009), the m2m-aligner (Jiampojamarn et al., 2007) is used to generate the many-to-many alignments, and the public implementation of DirecTL+ to train models.[7]

(c) **RPI-ISI (Lin et al., 2016)** A transliteration approach that uses an entity linking system (Wang et al., 2015) to jointly correct and re-rank the hypotheses produced by the generation model. The experiments compare to both the unconstrained inference (U) approach and the entity linking constrained inference (+EL) approach.

(d) **Seq2Seq w/ Att** A sequence to sequence generation model that uses soft attention as described in (Bahdanau et al., 2015). This model does not enforce monotonicity at inference time, and serves as direct comparison for Seq2Seq(HMA).

## 8.6. Experiments

The experiments analyze how effective is: **(a)** Seq2Seq(HMA) for transliteration generation when provided all available supervision (Section 8.6.1)? **(b)** the bootstrapping algorithm in the low-resource setting when only 500 examples are available (Section 8.6.2)?

### *8.6.1. Full Supervision Setting*

This experiment compares Seq2Seq(HMA) with the generation approaches described in Section 8.5, when provided all available supervision, to see how it fares under standard evaluation.

Results in the unconstrained inference (U) setting, shown in the top 5 rows of Table 32, demonstrate that Seq2Seq(HMA) (denoted by "Ours(U)") outperforms previous approaches on Hindi, Kannada, and Bengali, with almost 3-4% gains. Improvements over the Seq2Seq with Attention (Seq2Seq w/ Att) model demonstrate the benefit of imposing the mono-

[7] `https://code.google.com/p/directl-p`

| **Lang.** → <br> **Approach** ↓ | hi | kn | bn | ta | he | Avg. |
|---|---|---|---|---|---|---|
| **Full Supervision Setting (5–10k examples)** | | | | | | |
| Seq2Seq w/ Att (U) | 35.5 | 33.4 | 46.1 | 17.2 | 20.3 | 30.5 |
| P&R (U) | 37.4 | 31.6 | 45.4 | **20.2** | 18.7 | 30.7 |
| DirecTL+ (U) | 38.9 | 34.7 | 48.4 | 19.9 | 16.8 | 31.7 |
| RPI-ISI (U) | 40.3 | 29.8 | 49.4 | **20.2** | 21.5 | 32.2 |
| Ours(U) | **42.8** | **38.9** | **52.4** | **20.5** | **23.4** | **35.6** |
| **Approaches using Constrained Inference** | | | | | | |
| RPI-ISI + EL | 44.8 | 37.6 | 52.0 | **29.0** | **37.2** | 40.1 |
| Ours(DC) | **51.8** | **43.3** | **56.6** | 28.0 | 36.1 | **43.2** |
| **Low-Resource Setting (500 examples)** | | | | | | |
| Seq2Seq w/ Att (U) | 17.0 | 13.6 | 14.5 | 6.0 | 9.5 | 12.1 |
| P&R (U) | 21.1 | 16.6 | 34.2 | 9.4 | 13.0 | 18.9 |
| DirecTL+ (U) | 26.6 | 25.3 | 35.5 | 11.8 | 10.7 | 22.0 |
| Ours(U) | 29.1 | 27.7 | 37.7 | 11.5 | 16.2 | 24.4 |
| **Ours(U) + Boot.** | **40.1** | **35.1** | **50.3** | **17.8** | **22.8** | **33.2** |

Table 32: Comparing different generation approaches on the NEWS 2015 dataset using accuracy@1 as the evaluation metric for five languages — Hindi (hi), Kannada (kn), Bengali (bn), Tamil (ta) and Hebrew (he) – in full supervision and low-resource settings. "Ours" denotes the Seq2Seq(HMA) model, with (.) denoting the inference strategy. The rest of the approaches are described in Section 8.5. Numbers for RPI-ISI are from Lin et al. (2016).

tonicity constraint in the generation model. On Tamil and Hebrew, Seq2Seq(HMA) is at par with the best approaches, with negligible gap (∼0.3) in scores. Overall, Seq2Seq(HMA) achieves better (and sometimes competitive) scores than state-of-the-art approaches in full supervision settings. When comparing constrained inference approaches (Table 32, rows 6 and 7), it can be seen that using dictionary-constrained inference (as in "Ours(DC)") is more effective than using a entity-linking model for re-ranking (RPI-ISI + EL).

### *8.6.2. Low-Resource Setting*

In Table 32 (rows under "Low-Resource Setting"), the different models are evaluated in a low-resource setting when provided only 500 name pairs as supervision. Results are averaged over 5 different random sub-samples of 500 examples.

The results clearly demonstrate that all generation models suffer a drop in performance when provided limited training data. Note that models like Seq2Seq with Attention suffer a larger drop than those which enforce monotonicity, suggesting that incorporating monotonicity into the inference step in the low-resource setting is essential. After bootstrapping our weak generation model using Algorithm 3, the performance improves substantially (last row in Table 32). On almost all languages, the generation model improves by at least 6%, with performance for Hindi and Bengali improving by more than 10%. Bootstrapping results for the languages are within 2-4% of the best model trained with all available supervision.

To better analyze the progress of the transliteration model during bootstrapping, the accuracy@1 of the current transliteration model is plotted after each bootstrapping iteration for each of the languages (solid lines in Figure 23). For reference, the best performance for a generation model using all available supervision from Section 8.6.1 is also shown as dotted horizontal lines in Figure 23. From Figure 23, we can see that almost after 5 bootstrapping iterations, the generation model attains competitive performance to respective state-of-the-art models trained with full supervision.

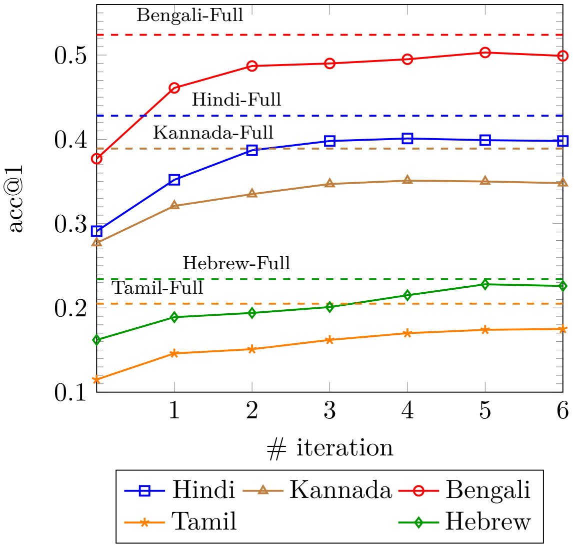


Figure 23: Plot showing accuracy@1 after each bootstrapping iteration for Hindi, Kannada, Bengali, Tamil and Hebrew, starting with only 500 training pairs as supervision. For comparison, the accuracy@1 of a model trained with all available supervision is also shown (respective dashed lines, marked X-Full).

### *8.6.3. Error Analysis*

Though our model is state of the art, it does present a few weaknesses. We have found that the dictionary sometimes misleads the model during constrained inference. For example, the correct transliteration of the Hindi word विद्युल, is *vidyul*, which is not present in the dictionary, but an incorrect hypothesis *vidul* is present, leading to incorrect prediction. Another issue comes from the proportion of native (i.e., from the source language) and foreign (i.e., from English or other languages) names in the training data. It is usually not the case that the source and target scripts have the same transliteration rules. For example, य in Hindi might represent *ya* in English or Hindi names, but *ja* in German. Similarly, while अ should be *a* in Hindi names, it could be any of a few vowels in English. The NEWS2015 dataset does not report a native/foreign ratio, but by our estimation, it is about 70/30 for each language. This native and foreign names dichotomy are some of the inherent challenges in transliteration, that are discussed in detail in the next section.

## 8.7. Challenges Inherent to Transliteration

The fact that all models in Table 32 perform well or poorly on the same languages suggests that most of the observed performance variation is the result of factors *intrinsic* to the specific languages. Here we analyze some challenges that are inherent to the transliteration task, and explain why the performance ceiling is well under 100% for all languages, and lower for languages like Tamil and Hebrew than the others.[8]

### *8.7.1. Source and Target-Specific Issues*

**Source-Driven.** Some transliteration errors are due to ambiguities in the source scripts. For instance, the Tamil script uses a single character to denote {*ta*, *da*, *tha*, *dha*}, a single character for {*ka*, *ga*, *kha*, *gha*}, etc., while the rest of the Indian scripts have unique characters for each of these. Thus, names like *Har**t**ley* and *Har**d**ley* are entirely indistinguishable in Tamil but are distinguishable in the other scripts. We illustrate this problem by transliterating back and forth between Tamil and Hindi. When transliterating Hindi → Tamil, the model achieves an accuracy of 31%, that drops to 15% when transliterating Tamil → Hindi, suggesting that the Tamil script is more ambiguous.

The Hebrew script also introduces error because it tends to omit vowels or write them ambiguously, leaving the model to guess between plausible choices. For example, the word מלך could be transliterated *melech* (meaning *king*) just as easily as *malach* (meaning *he ruled*). When Hebrew does write vowels, it reuses consonant letters, again ambiguously. For example, ה can be used to express *a* or *e*, so שמונה can be either *shmona* or *shmone* ( either a *masculine eight* or a *feminine eight*). The script also does not reliably distinguish *b* from *v* or *p* from *f*, among others.

All languages run into problems when they are faced with writing sounds that they do not natively distinguish. For example, Hindi does not make a distinction between *w* and *v*, so both *vest* and *west* are written as वेस्ट in its script.

[8] A similar analysis to ours was presented in (Merhav and Ash, 2018).

These script-specific deficiencies explains why all models struggle on Tamil and Hebrew relative to the others. These issues cannot be completely resolved without memorizing individual source-target pairs and leveraging context.

**Target-Driven.** Some errors arise from the challenges presented by target script (here Latin script for English). To handle English's notoriously convoluted orthography, a model has to infer silent letters, decide whether to use *f* or *ph* for /f/; use *k*, *c*, *ck*, *ch*, or *q* for /k/, and so on. The problem is made worse because English is not the only language that uses Latin script. For example, German names like *Schmidt* should be written with *sch* instead of *sh*, and for French names like *Margot* and *Margeau* (which are pronounced the same), we have to resort to memorization. The arbitrariness extends into borrowings from the source languages as well. For example, the Indian name *Bangalore* is written with a silent-*e*, and the name *Lakshadweep* contains *ee*, instead of the expected *i*.

### *8.7.2. Disparity between Native and Foreign*

Certain names are well-integrated into the source language etymologically (Indian names like *Jasodhara* or *Ramanathan* for Hindi), while some are not (French *Grenoble* or Japanese *Honshu* for Hindi). We refer to the former as *native* names for that language, and the latter as *foreign* names. The above datasets include an unspecified mix of native and foreign names. How does the transliteration performance vary across these two name types?

To quantify the effect of this, we annotate native and foreign names in the test split of the four Indian languages, and evaluate performance for both categories. Table 33 shows that the performance is significantly better on native names for all the languages. A possible reason for is that the source scripts were designed for writing native names (e.g., Tamil script lacks separate {*ta*, *da*, *tha*, *dha*} characters because Tamil does not distinguish these sounds). Furthermore, foreign names have a wide variety of origins with their own conventions (Section 8.7.1). The performance gap is proportionally greatest for Tamil, likely due to its script-specific deficiencies.

| Language | Native | Foreign | Ratio |
|---|---|---|---|
| Hindi | 45.1 | 31.4 | 1.44 |
| Bengali | 63.1 | 20.1 | 3.14 |
| Kannada | 42.6 | 23.1 | 1.84 |
| Tamil | 24.3 | 05.2 | 4.67 |

Table 33: Accuracy@1 for native and foreign words for four languages (Section 8.7.2). Ratio is native performance relative to foreign.

This disparity between native and foreign names is a problem since any model must learn essentially separate transliteration schemes for each name type.

## 8.8. Case Studies

The practical utility of our approach in low-resource settings and for downstream applications is evaluated through two case studies. First, Section 8.8.1 evaluates if obtaining an adequate seed list is possible with a few hours of manual annotation from a *single* human annotator. Then, Section 8.8.2 evaluates the impact of our approach on a downstream task of candidate generation for Tigrinya and Macedonian entity linking.

| Language | Monolingual Corpus | Vocabulary |
|---|---|---|
| Punjabi | Corpus ILCI-II♠ | 30k |
| Armenian | TED♣ | 50k |
| Tigrinya | Habit Project♦ | 225k |
| Macedonian | TED♣ | 60k |

♦=`habit-project.eu/wiki/TigrinyaCorpus`, ♠=`tdil-dc.in`,

♣=`github.com/ajinkyakulkarni14/TED-Multilingual-Parallel-Corpus`

Table 34: Corpora used for obtaining foreign vocabulary $\mathcal{V}_f$ for bootstrapping in the case studies in Section 8.8.1 and Section 8.8.2.

### *8.8.1. Manual Annotation*

The manual annotation exercises simulate a low-resource setting with only a single human annotator is available. This experiment judges the usability of the annotations by training models on them and evaluating the models on test sets of 1000 names each, obtained from Wikipedia inter-language links. For bootstrapping experiments, the corpora shown in Table 34 is used to obtain foreign vocabulary $\mathcal{V}_f$.

**Languages Studied.** The evaluation involves two languages: Armenian and Punjabi.

Spoken in Armenia and Turkey, Armenian is an Indo-European language with no close relatives. It has Eastern and Western dialects with different spelling conventions. One issue was that the Armenian Wikipedia is primarily written in the Eastern dialect, while our annotator was a native Western speaker. The annotator produced Western Armenian name-pairs, that were mechanically mapped to “Eastern” Armenian by swapping five Armenian character pairs: դ/տ, պ/բ , ք/կ , ծ/ձ, ճ/ջ, so that evaluation can be performed against name pairs from Armenian Wikipedia.

Punjabi is an Indic language from Northwest India and Pakistan that is closely related to Hindi. Our annotator grew up primarily speaking Hindi.

**Annotation Guidelines.** Annotators were given two tasks. First, they were asked to write two names and their English transliterations for each letter in the source script: one beginning with the letter and another containing it elsewhere. (e.g. *__J__ulia* and *Ben__j__amin* for the letter *j*, if the source were English). The is done to ensure good coverage over the alphabet. Next, annotators were shown a list of English words and were asked to provide plausible transliteration(s) into the target script. The list had a mix of recognizable foreign (e.g., *Clinton*, *Helsinki*) and native names (e.g., *Sarkessian*, *Yerevan* for Armenian).

About 600 and 500 annotated pairs were collected for Armenian and Punjabi respectively. Table 35 shows that the performance of the models trained on the annotated data is compa-

| **Lang.** → <br> **Approach** ↓ | Punjabi | Armenian |
|---|---|---|
| Ours(U) | 33.4 | 49.9 |
| Ours(U) + Bootstrapping | 44.5 | 55.8 |
| Annotation Time (hours) | 5 | 4 |

Table 35: Accuracy@1 of Seq2Seq(HMA) model supervised using human annotated seed set in Punjabi and Armenian (with and without bootstrapping). Both languages perform well relative to the other languages investigated so far. Both annotation sub-tasks took roughly the same time.

rable to that on the standard test corpora for other languages. This show that our approach is robust to human inconsistencies and regional spelling variations, and that obtaining an adequate seed list is possible with just a few hours of manual annotation.

### *8.8.2. Candidate Generation*

Since transliteration is an intermediate step in many downstream multilingual information extraction tasks (Darwish, 2013; Kim et al., 2012; Jeong et al., 1999; Virga and Khudanpur, 2003; Chen et al., 2006), it is possibly to gauge its performance extrinsically by the impact it has on such tasks. This section uses the task of *candidate generation* (CG), which is a key step in cross-lingual entity linking, as the subject.

The goal of cross-lingual entity linking (McNamee et al., 2011; Tsai and Roth, 2016; Upadhyay et al., 2018a) is to ground spans of text written in any language to an entity in a knowledge base (KB). For instance, grounding [**Chicago**] in the following German sentence to `Chicago_(band)`.[9]

[**Chicago**] *wird in Woodstock aufzutreten.*

CG in entity linking identifies a set of plausible entities that a given mention can link to, while ensuring the correct KB entity belongs to that set (Section 7.2.1). For the above German sentence, CG would provide a list of possible KB entities for the string *Chicago*:

[9] Translation: Chicago will perform at Woodstock.

`Chicago_(band)`, `Chicago_(city)`, `Chicago_(font)`, etc., so that context-sensitive inference (Section 7.2.1) can select the band. Foreign scripts pose an additional challenge for CG because they must be transliterated before they are passed on to candidate generation. For instance, any mention of *Chicago* in Amharic must first be transliterated from **ቺካጎ**.

Most approaches for CG use Wikipedia inter-language links to generate the lists of candidates (Tsai and Roth, 2016). While recent approaches such as Tsai and Roth (2018) have used name translation for CG, they require over 10k examples for languages with non-Latin scripts, which is prohibitive for low-resource languages with little Wikipedia presence.

**Candidate Generation with Transliteration.** We evaluate the extent to which our approach improves recall of a naive CG baseline that generates candidates by performing exact name match. For each span of text to be linked (or *query mention*), first it is checked if the naive name matching strategy finds any candidates in the KB. If none are found, the query mention is back-transliterated to English, and at most 20 candidates are generated using a inverted-index from English names to KB entities. The evaluation metric is recall@20, i.e., if the gold KB entity is in the top 20 candidates. The effect of including transliteration on CG is evaluated for two test languages — Tigrinya and Macedonian.

**Tigrinya** is a South Semitic language related to Amharic, written in the Ethiopic script, and spoken primarily in Eritrea and northern Ethiopia. The Tigrinya Wikipedia has <200 articles, so inter-language links (∼7.5k) from the Amharic Wikipedia are used instead to extract 1k name pairs for the seed set. The monolingual corpus in Table 34 is used for bootstrapping. The model is evaluated on the *unsequestered set* provided under the NIST LoReHLT evaluation, containing 4,630 query mentions.

The Ethiopic script is an alphasyllabary, where each character is consonant-vowel pair. For example, the character **መ** is *mä*, **ሚ** with a tail is *mi*, and **ሞ** with a line is *mo*. With 26 consonants and 8 vowels, this leads to a set of >200 characters creating a sparsity problem since each character has its own Unicode code point. However, the code points are organized

| Approach | Recall@20 |
|---|---|
| **Tigrinya** | |
| Name match (baseline) | 31.4 |
| Ours | 35.6 |
| Ours (CV-split) | 41.3 |
| **Ours (CV-split) + Bootstrapping** | **46.2** |
| **Macedonian** | |
| Name match (baseline) | 33.6 |
| Ours | 72.2 |
| **Ours + Bootstrapping** | **76.8** |

Table 36: Comparing candidate recall@20 for different approaches on Tigrinya and Macedonian. CV-split refers to consonant-vowel splitting. Using our transliteration generation model with bootstrapping yields the highest recall, improving significantly over a name match baseline.

so that they can be automatically split[10] into unique consonant and vowel codes *without explicitly understanding the script.* Arbitrary ASCII codes are assigned to each consonant and vowel so that መ/*mä* becomes “D 1” and ሞ/*mo* becomes “D 6.” This consonant-vowel splitting (CV-split) reduces the number of unique input characters to 55.

**Macedonian** is a South Slavic language closely related to the languages of the former Yugoslavia and written in a local variant of the Cyrillic alphabet. The experiment uses the Macedonian test set constructed by McNamee et al. (2011) containing 1956 query mentions. A seed set of 1k name pairs was obtained from the Wikipedia inter-language links for Macedonian, and the monolingual corpus from Table 34 was used for bootstrapping.

**Candidate Generation Results.** Table 36 shows the results of the candidate generation experiments in the two languages. For Tigrinya, candidate generation with transliteration improves on the baseline by 4.2%. Splitting the characters (CV-split) gives another 5.7%, and adding bootstrapping gives 4.9% more. Our approach yields an overall 14.8% improvement in recall over the baseline, showing that the little available supervision can be

[10] Consonant = Unicode / 8; Vowel = Unicode % 8

effectively exploited by bootstrapping. Macedonian yields more dramatic results, where transliteration provides 38.6% improvement (more than double the baseline), with bootstrapping providing another 4.6%. The differences between Tigrinya and Macedonian is likely due both to their test sets, corpora and writing systems.

## 8.9. Summary

The chapter examined the transliteration problem, that serves as an important component of candidate generation for cross-lingual entity linking. We presented a new transliteration generation model, and a new bootstrapping algorithm that can iteratively improve a weak generation model using constrained discovery. The model presented here achieves state-of-the-art results on standard training set sizes, and more importantly, works well in a low-resource setting with the aid of the bootstrapping algorithm. The key benefit of the bootstrapping approach is that it can "recover" most of the performance lost in the low-resource setting when little supervision is available by training with a smaller seed set, an English name dictionary, and a list of un-annotated words in the target script. Through case studies, it was shown that collecting an adequate seed list is practical with a few hours of annotation. The benefit of incorporating our transliteration approach in a downstream task, namely candidate generation for cross-lingual entity linking, was also demonstrated. Challenges inherent to the transliteration task were also discussed, particularly the impact of the native/foreign distinction in the train data and the difficulties posed by specific scripts or pairs of scripts.

There are several interesting directions for future work. Performing model combination, either by developing hybrid transliteration models (Nicolai et al., 2015) or by ensembling (Finch et al., 2016), can further improve low-resource transliteration. Jointly leveraging similarities between related languages, such as writing systems or phonetic properties (Kunchukuttan et al., 2018), also shows promise for low-resource settings. Our analysis suggests value in revisiting "transliteration in context" approaches (Goto et al., 2003; Hermjakob et al., 2008), especially for languages like Hebrew.

# CHAPTER 9 : Conclusion

So far, progress in NLP has largely benefited just a select few languages. For its fruits to be accessible to more of the world's diverse set of languages, it is essential to enable NLP in multilingual settings. However, such endeavors are beset with the challenges of working with no or limited supervision in the language of interest.

Throughout this thesis, I discussed different approaches that either transfer, share or exploit knowledge across languages to achieve this goal. The recurring theme was using cross-lingual signals to developing approaches that facilitate NLP in English and other languages. The key ingredient in all approaches was some form of cross-lingual word representation, which made sharing or transfer of supervision possible.

## 9.1. Summary of Contributions

The thesis began by motivating the need for developing NLP technology for languages other than English in Chapter 1. A large population of the world produces and consumes information in a variety of languages, evident from the growing size of the non-English content on the Web. Multilingual NLP can help us leverage this relatively untapped source of knowledge, and also serve as a means for making social impact in urgent situations like disaster relief. From a scientific standpoint, working in Multilingual NLP scenarios also serves as a test-bed for evaluating existing NLP techniques, revealing their strengths or limitations. Nevertheless, progress in Multilingual NLP is hindered by the lack of annotated data in most languages, posing a challenge for traditional supervised learning approaches. I discussed why existing annotation-based or translation-based approaches are ill-equipped to deal with this challenge, and argued why sharing resources developed for popular languages like English is a more viable strategy.

Chapter 2 gave brief history of representation learning approaches used in NLP. The driving force behind most of these approaches were two hypotheses rooted in the distributional

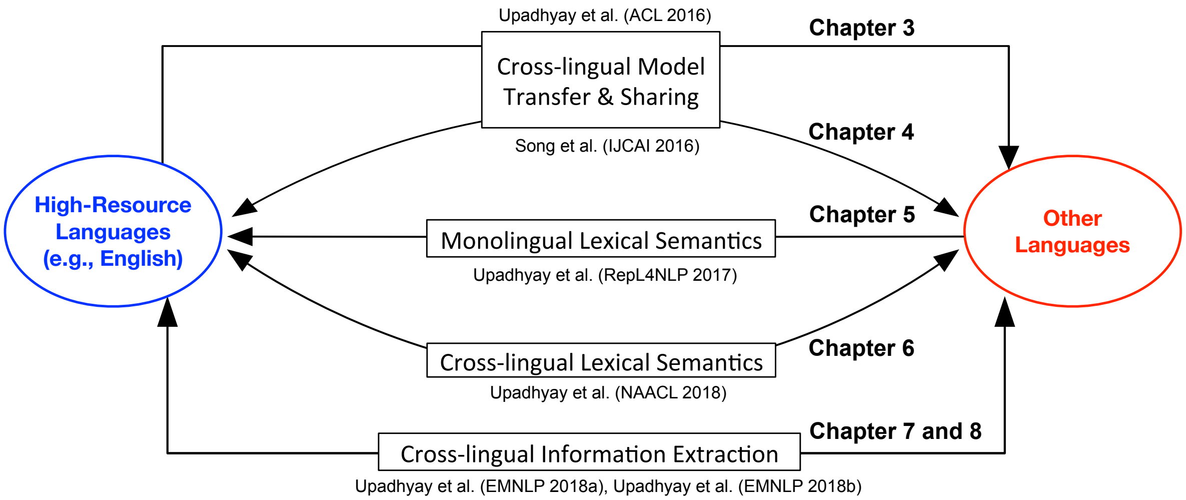

Upadhyay et al. (ACL 2016)
Cross-lingual Model Transfer & Sharing
Song et al. (IJCAI 2016)
Chapter 3
Chapter 4
High-Resource Languages (e.g., English)
Other Languages
Chapter 5
Monolingual Lexical Semantics
Upadhyay et al. (RepL4NLP 2017)
Chapter 6
Cross-lingual Lexical Semantics
Upadhyay et al. (NAACL 2018)
Chapter 7 and 8
Cross-lingual Information Extraction
Upadhyay et al. (EMNLP 2018a), Upadhyay et al. (EMNLP 2018b)

Figure 24: Summary of contributions made in the thesis. Different chapters demonstrated how cross-lingual representations can help achieve model transfer and sharing (Chapter 3 and 4), aid in lexical semantics tasks either monolingually (Chapter 5) or cross-lingually (Chapter 6), and facilitate cross-lingual information extraction (Chapter 7 and 8).

semantics literature. By appealing to these hypotheses in different ways, one can derive word representations directly from raw text, an attractive alternative to manually building lexical semantic resources. I trace the history of different word representation paradigms, starting with cluster-based representations (à la Brown Clustering) to the more efficient and compact vector-based representations. Different variants of popular vector space representations were discussed, laying the groundwork for understanding their cross-lingual counterparts in later chapters. The chapter ends with a discussion of different aspects where these word representations fall short, from issues arising either from modeling assumptions or limitations of the distributional hypothesis itself. Importantly, I discussed the inability of monolingual representations to capture cross-lingual relationships between words in different languages, making it difficult to share resources across languages.

The next few chapters were devoted to demonstrate how cross-lingual representations can help alleviate this limitation, thereby enabling sharing and transfer of resources across languages. Figure 24 summarizes the tasks used as evidence to support this claim.

The next chapter described different approaches to learn cross-lingual representations to overcome the above limitation of monolingually trained distributional representations. By viewing them as a continuous approximation of discrete translation dictionaries that can "translate" features used in a model, I motivated why using cross-lingual representations can aid model transfer across languages. The unifying motif of all approaches was the use of some form of cross-lingual alignment — word-level, sentence-level, document-level, or a combination of these — in addition to the monolingual distributional signal. The chapter also made a comparison of these approaches on various tasks, both intrinsic and extrinsic, and evaluated the suitability of the different cross-lingual alignments for each.

Traditional distributional representations require lots of data to learn relationships that might be explicitly stated in manually constructed resources like Wikipedia. In Chapter 4, I discussed ESA, a representation learning approach that derives word representations from Wikipedia by appealing to the bag-of-words hypothesis. Then I showed how ESA can be extended by utilizing metadata that connects Wikipedias in different languages to obtain cross-lingual word representations. Unlike the representation in Chapter 2, these cross-lingual representations were interpretable, where the dimensions correspond to concepts in Wikipedia to which the word is relevant. Another attractive property of these representations is that they can be used in unsupervised classification paradigms (referred to as "dataless" in Chapter 4), and achieve accuracies comparable to supervised classification.

In Chapter 5, I discussed how these cross-lingual signals can be used to overcome another limitation of traditional word representations — that they cannot model polysemy. I motivated why data-driven approaches learn multi-sense word representations hold more appeal compared to approaches that use lexical resources, as the latter are inherently limited to the senses listed in those resources. Building on the line of work that exploits translational information to tease apart word senses, I described an data-driven approach to learn sense representations by exploiting multilingual translational information using Bayesian non-parametrics. The experiments showed that the proposed approach yielded comparable performance to a monolingual model trained on more data.

In Chapter 6, I discussed a new lexical semantic task of identifying asymmetric relationships like hypernymy between words in different languages. Building on work in monolingual hypernymy detection, I proposed an unsupervised approach to probe the distributional representations of the words to detect if the relationship exists. These representations were derived from syntactic contexts, and aligned cross-lingual using a small bilingual dictionary. Experimental evaluations showed that the model performance was robust to various low-resource scenarios, especially when the syntactic contexts were derived using a weak dependency parser, a encouraging result for languages without a dependency treebank. The proposed approach has the potential to complement efforts to semi-automatically construct cross-lingual ontologies such as BabelNet and the Open Multilingual WordNet.

The overarching theme of Chapters 4, 5, 6 shows different applications of cross-lingual representations for lexical semantics tasks. Chapter 7 and 8 examine the downstream information extraction task of entity linking in the cross-lingual setting. Chapter 7 motivated the cross-lingual entity linking problem as a means for acquiring knowledge expressed in multilingual text, similar to the function that monolingual entity linking performs for tasks like question answering and knowledge base population. Apart from dealing with the ambiguity and variability of language, in this task one has to deal with the additional challenge of working with limited supervision in the language of interest. To address this challenge, I show how cross-lingual representations can be used to leverage supervision from multiple languages jointly for the task of cross-lingual entity linking. Another benefit of the proposed approach is that it can train a *single* model for multiple related languages, making more efficient use of available supervision than previous approaches that trained separate models. Later in Chapter 8, I examined a key component of XEL, namely candidate generation, and show how its coverage for the XEL task can be improved by using transliteration. For this, I described a transliteration generation approach that can operate in low-resource settings using a little supervision and an English KB to bootstrap. The key benefit of bootstrapping is that it can "recover" most of the performance lost in the low-resource setting.

## 9.2. Future Directions

There are several directions in cross-lingual NLP that warrant further research; I briefly describe a few of them below.

**Static vs Dynamic Representations.** Section 5.8 touched upon a fundamental limitation of static word representations (either monolingual or cross-lingual), in that they divorce the context and the word's representation by compiling a fixed representation for the word. Recently, approaches that use language modeling to learn dynamic, context-sensitive word representations, like ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018), have been proposed to overcome this issue in the monolingual setting. Related research questions can be asked in this direction. For instance, how can one extend these dynamic, context-sensitive representations cross-lingually? Are there other similar auxiliary tasks like language modeling for learning contextual representations?

**Cross-lingual Understanding and Reasoning.** One of the fundamental limitations of distributional representations discussed in Chapter 2 was its inability to account for the compositional nature of language, motivating the need to marry formal semantics and distributional approaches. A similar argument can be made for the need for compositional semantics in cross-lingual settings. Developing approaches for learning compositional meaning representations in cross-lingual settings with little or no language specific annotation is an active area of research. Such approaches can pave the way for multilingual versions of language understanding tasks like semantic parsing (Duong et al., 2017a; Susanto and Lu, 2017; Zou and Lu, 2018; Zhang et al., 2018a,b), which so far have been confined to English.

**Code-mixed and Code-switched NLP.** In all the cross-lingual tasks studied in this thesis, an implicit assumption was that the input text was in a single language, and said language was known beforehand. However, often a speaker alternates between two or more languages in the same conversation, a phenomenon referred as *code-switching.* Indeed, when conversing with another multilingual, a multilingual person can interchange seamlessly between two or more languages, either at the word, phrase or sentence level. For instance, a person who is fluent in both Hindi and English, can utter the following sentence “tum mujhe dinner ke liye fifty-five dollars owe karte ho” (translates to “you owe me fifty-five dollars for dinner”), which is a mix of Hindi (written in Latin script) and English. In such situations, it is necessary to develop ways to either identify which language’s feature representation to use, or operate at a feature representation that smoothens the difference in the two languages.

Overall, a wide variety of problems remain to be explored, with some (such as semantic parsing) being cross-lingual counterparts of traditional NLP problems, while others (such as code-mixing) being unique to the multilingual setting. These uncharted territories are attractive directions for future work in developing truly language-independent approaches, and finding better ways of leveraging existing supervision available in other languages.

# APPENDIX

## A.1. Statistical Measures used in the Thesis

**Spearman Rank Correlation Coefficient (Spearman, 1904)** The Spearman Rank Correlation Coefficient (SRCC) is a way to compare the "similarity" of rankings produced by two different models. For a set of $n$ items, suppose models $A$ and $B$ produce scores such that item $i$ gets a rank $r_i$ according to model $A$'s scores and rank $R_i$ according to model $B$'s scores. Define $a_{ij} = (r_i - r_j)$ as the difference in the rank of item $i$ and $j$ according to model $A$. A similar term $b_{ij} = (R_i - R_j)$ can be defined suitably. Then, the SRCC is,

$$\tau = \frac{\sum_{i,j} a_{ij} b_{ij}}{\sqrt{\sum_{i,j} a_{ij}^2 \sum_{i,j} b_{ij}^2}} \tag{A.1}$$

The coefficient assumes a value between 1 and -1, with 1 indicating complete agreement in the rankings, while -1 indicating complete disagreement.

**Steiger's Test (Steiger, 1980; Williams, 1959)** Suppose the SRCC between the ranking produced by model $A$ and the reference ranking is 0.9 and the ranking produced by model $B$ and the reference ranking is 0.7. How to assess if this difference is statistically significant? Steiger's test is used to calculate the statistical significance between two dependent correlation coefficients, i.e., correlation between random variables ($X$, $Y$) and ($X$,$Z$). For the situation described above, $X$ will be the reference ranking, and $Y$ and $Z$ are the rankings produced by the models. Suppose $r_{pq}$ is the rank correlation between variables $P$ and $Q$, then the $p$-value can be computed as,

$$p = 1 - cdf(|T_2|, N - 3) \tag{A.2}$$

$$\text{where} \qquad T_2 = d\sqrt{\frac{(N-1)(1+r_{yz})}{2\frac{N-1}{N-3}|R| + \bar{r}^2(1-r_{yz})^3}} \tag{A.3}$$

$$R = \begin{bmatrix} 1 & r_{xy} & r_{xz} \\ r_{xy} & 1 & r_{yz} \\ r_{xz} & r_{yz} & 1 \end{bmatrix} \qquad d = r_{xy} - r_{xz} \qquad \bar{r} = \frac{1}{2}(r_{xy} + r_{xz}) \tag{A.4}$$

where *cdf* is the cumulative distribution function of the t-distribution, $N$ is the number of items being ranked, and $|R| = 1 - r_{xy}^2 - r_{yz}^2 - r_{xz}^2 + 2r_{xy}r_{yz}r_{xz}$ is the determinant of the correlation matrix $R$.

**McNemar's Test (McNemar, 1947)** McNemar's test is a non-parametric test to determine if the difference in the accuracies of two classifiers is statistically significant. McNemar's test considers the number $c_{01}$ of instances mis-classified by classifier 1 but correctly classified by classifier 2, and a similarly defined term $c_{10}$. Under the null hypothesis both classifiers have the same error rate, i.e., $c_{01} = c_{10}$. McNemar's test uses the $\chi^2$ test for goodness of fit comparing the observed counts and expected counts under the null hypothesis, where the $\chi^2$ statistic is computed as,

$$\frac{(|c_{01} - c_{10}| - 1)^2}{c_{01} + c_{10}} \tag{A.5}$$

where the -1 in the numerator is a correction term to account for the fact that the statistic is discrete while the $\chi^2$ distribution is continuous.

**Fleiss' Kappa (Fleiss, 1971)** A statistical measure of the reliability of inter-annotator agreement when $N$ items are assigned to $k$ classes by $n$ annotators. It generalizes Cohen's Kappa (Cohen, 1960), which measures the same for exactly 2 annotators. The statistic computes the probability of the observed agreement, discounting for chance agreement,

$$\kappa = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e} \tag{A.6}$$

where $\bar{P}_e$ is the probability of agreement arising out of chance, and $\bar{P}$ is the probability of observed agreement. If $n_{ij}$ is the number of annotators who assigned $i^{th}$ item to the $j^{th}$ class, these quantities can be computed as,

$$\bar{P} = \frac{1}{N}\sum_{i=1}^{N} P_i, \qquad \bar{P}_e = \sum_{j=1}^{k} p_j^2 \tag{A.7}$$

$$\text{where } p_j = \frac{1}{Nn}\sum_{i=1}^{N} n_{ij}, \qquad P_i = \frac{1}{n(n-1)}\sum_{j=1}^{k} n_{ij}(n_{ij}-1) \tag{A.8}$$

where $p_j$ is the observed probability of assigning to $j^{th}$ class and $P_i$ is the observed probability of annotators agreeing for $i^{th}$ item.

**Adjusted Rand Index (Hubert and Arabie, 1985)** Adjusted Rand Index (ARI) is a measure to compare the quality of a predicted clustering $X = \{X_1, \cdots\}$ against a reference clustering $Y = \{Y_1, \cdots\}$ , where $X_i$ is a subset with $a_i$ items, and $Y_j$ is a subset with $b_j$ items. It is derived from another measure, the Rand Index (RI) (Rand, 1971), by discounting for the expected score for a random clustering. If $a$ denotes the number of pairs that are in the same cluster in $X$ and $Y$, and $d$ denotes the number of pairs that are in different clusters in $X$ and $Y$ then RI is $\frac{a+d}{\binom{n}{2}}$. It can be shown that $a + d$ can be simplified to a linear transformation of $e = \sum_{ij}\binom{n_{ij}}{2}$, where $n_{ij}$ is the number of objects in the common to cluster $X_i$ and $Y_j$ (Yeung and Ruzzo, 2001). Using $e$, ARI can be defined as,

$$ARI = \frac{e - \tilde{e}}{e_{max} - \tilde{e}} \tag{A.9}$$

where $\tilde{e}$ is expected value of $e$ for a random clustering, $e_{max} = \frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right]$ is the maximum possible value of $e$, and $n$ is the number of items being clustered. By assuming the hyper-geometric model, Hubert and Arabie (1985) showed that $\tilde{e} = \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}$. ARI has a maximum value of 1 (for perfect agreement) and a expected value of 0, unlike RI which has a non-zero expected value, making ARI a more sensitive index.

## A.2. BiSparse — Bilingual Sparse Coding

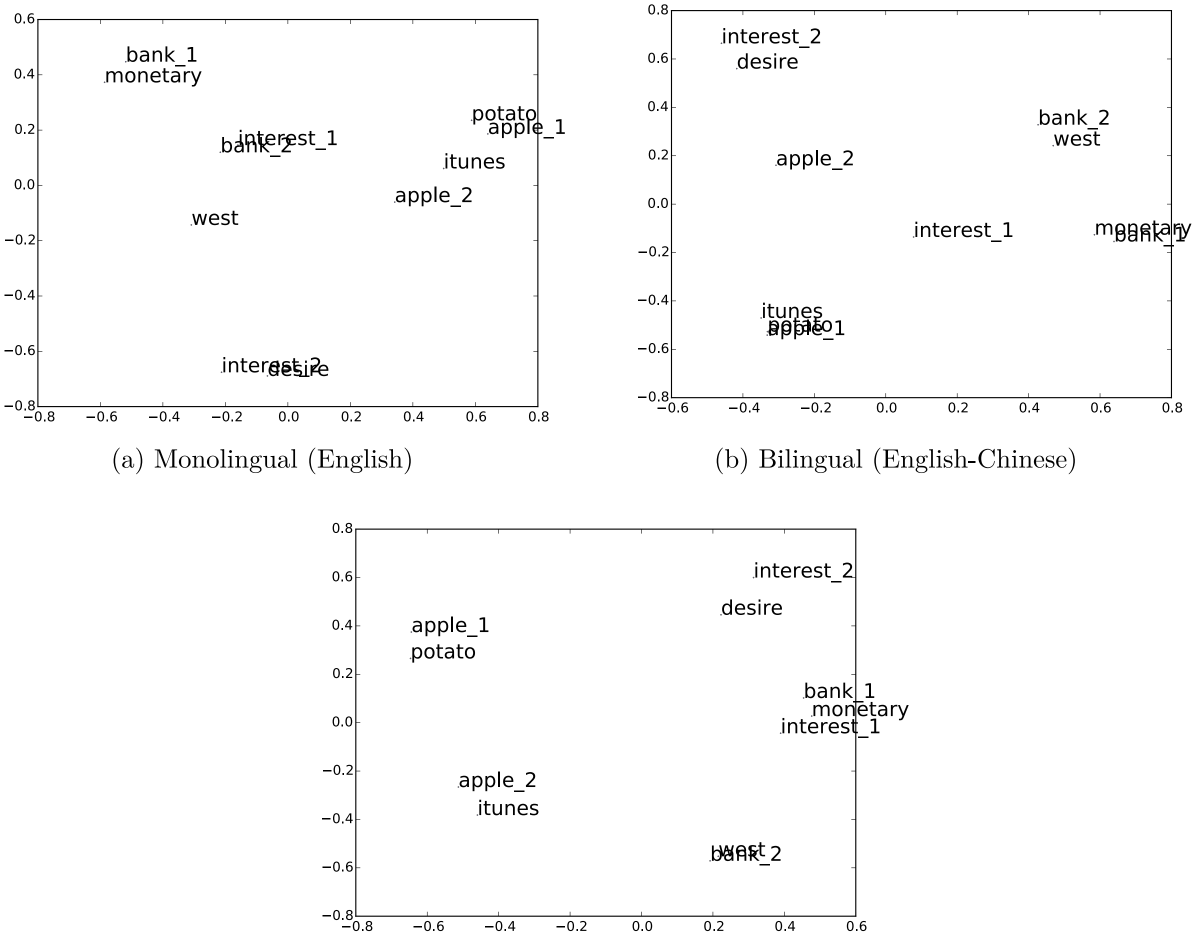
For two languages with vocabularies $v_e$ and $v_f$, and monolingual dependency embeddings $\mathbf{X_e}$ and $\mathbf{X_f}$, BiSparse solves the following objective:

$$\begin{aligned}\underset{\mathbf{A_e},\mathbf{D_e},\mathbf{A_f},\mathbf{D_f}}{\operatorname{argmin}} \sum_{i=1}^{v_e} & \frac{1}{2}||\mathbf{A_e}_i\mathbf{D_e}^{\mathrm{T}} - \mathbf{X_e}_i||_2^2 \ + \lambda_e||\mathbf{A_e}_i||_1 \\ + \sum_{j=1}^{v_f} & \frac{1}{2}||\mathbf{A_f}_j\mathbf{D_f}^{\mathrm{T}} - \mathbf{X_f}_j||_2^2 \ + \lambda_f||\mathbf{A_f}_j||_1 \\ + \sum_{i,j} & \frac{1}{2}\lambda_x\mathbf{S}_{ij}||\mathbf{A_e}_i - \mathbf{A_f}_j||_2^2 \\ \text{s.t. } \mathbf{A_k} > \mathbf{0} \qquad & \|\mathbf{D_k}_i\|_2^2 \leq 1 \qquad \mathbf{k} \in \{\mathbf{e}, \mathbf{f}\}\end{aligned} \tag{A.10}$$

The first two rows and the constraints in Equation A.10 encourage sparsity and non-negativity in the embeddings, by solving a sparse coding problem where ($\mathbf{D_e}$, $\mathbf{D_f}$) represent the dictionary matrices and ($\mathbf{A_e}, \mathbf{A_f}$) the final sparse representations. The third row imposes bilingual constraints, weighted by the regularizer $\lambda_x$, so that words that are strongly aligned according to $\mathbf{S}$ have similar representations. The above optimization problem can be solved using a proximal gradient method such as FASTA (Goldstein et al., 2014).

## A.3. Qualitative Analysis for Multi-sense Embeddings

Figure 25 shows PCA plots for 11 sense vectors for 9 words using monolingual, bilingual and multilingual models. With monolingual training (Fig 25a) the model infers two senses for *bank* (denoted by *bank_1* and *bank_2*), but both are close to financial terms, suggesting their distinction was not recognized. The same observation holds for the senses of *apple*. With bilingual training (Fig 25b), the model infers two senses of *bank* correctly, and two sense of *apple* become more distant. The model can still improve e.g., pulling *interest* towards the financial sense of *bank*, and pulling *itunes* towards *apple_2*. Finally, with multilingual training (Fig 25c), all senses of the words are more clearly clustered — the senses of *apple*, *interest*, and *bank* are well separated, and are close to sense-specific words.



(a) Monolingual (English)

(b) Bilingual (English-Chinese)

(c) Multilingual (English-{French,Chinese})

Figure 25: PCA plots for the vectors for {*apple, bank, interest, itunes, potato, west, monetary, desire}* with multiple sense vectors for *apple* (*apple_1* and *apple_2*), *interest* (*interest_1* and *interest_2*) and *bank* (*bank_1* and *bank_2*) obtained using monolingual (25a), bilingual (25b) and multilingual (25c) training.

# BIBLIOGRAPHY

Ž. Agić, D. Hovy, and A. Søgaard. If all you have is a bit of the Bible: Learning POS taggers for Truly Low-resource Languages. In *Proceedings of ACL-IJCNLP*, volume 2, pages 268–272, 2015.

R. Aharoni and Y. Goldberg. Morphological Inflection Generation with Hard Monotonic Attention. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, July 2017.

W. Ammar, C. Dyer, and N. A. Smith. Transliteration by Sequence Labeling with Lattice Encodings and Reranking. In *Proceedings of the 4th Named Entity Workshop*, pages 66–70. Association for Computational Linguistics, 2012.

W. Ammar, G. Mulcaire, M. Ballesteros, C. Dyer, and N. Smith. Many Languages, One Parser. *Transactions of the Association for Computational Linguistics*, 4:431–444, 2016a.

W. Ammar, G. Mulcaire, Y. Tsvetkov, G. Lample, C. Dyer, and N. A. Smith. Massively Multilingual Word Embeddings. *arXiv preprint arXiv:1602.01925*, 2016b.

S. Arora, Y. Li, Y. Liang, T. Ma, and A. Risteski. Linear Algebraic Structure of Word Meanings, with Applications to Polysemy. *TACL*, pages 1–25, 2018.

M. Artetxe, G. Labaka, and E. Agirre. Learning Bilingual Word Embeddings with (almost) No Bilingual Data. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 451–462, Vancouver, Canada, July 2017. Association for Computational Linguistics.

S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A Nucleus for a Web of Open Data. *The Semantic Web*, pages 722–735, 2007.

L. Aufrant, G. Wisniewski, and F. Yvon. Zero-resource Dependency Parsing: Boosting Delexicalized Cross-lingual Transfer with Linguistic Knowledge. In *Proc. the International Conference on Computational Linguistics (COLING)*, 2016.

D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2015.

C. F. Baker, C. J. Fillmore, and J. B. Lowe. The Berkeley FrameNet Project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING-ACL 1998*, pages 86–90, 1998.

C. Bannard and C. Callison-Burch. Paraphrasing with Bilingual Parallel Corpora. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 597–604, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

M. Bansal, J. DeNero, and D. Lin. Unsupervised Translation Sense Clustering. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, 2012.

M. Bansal, K. Gimpel, and K. Livescu. Tailoring Continuous Word Representations for Dependency Parsing. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2014.

M. Baroni and A. Lenci. Distributional Memory: A General Framework for Corpus-based Semantics. *Computational Linguistics*, 36(4):673–721, Dec. 2010. ISSN 0891-2017.

M. Baroni and A. Lenci. How we BLESSed Distributional Semantic Evaluation. In *Proc. of the GEMS Workshop*, 2011.

M. Baroni, S. Bernardini, A. Ferraresi, and E. Zanchetta. The WaCky Wide Web: A Collection of Very Large Linguistically Processed Web-Crawled Corpora. *Proc. of the International Conference on Language Resources and Evaluation (LREC)*, 2009.

M. Baroni, R. Bernardi, N.-Q. Do, and C.-c. Shan. Entailment Above the Word Level in Distributional Semantics. In *Proc. of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2012.

M. Baroni, G. Dinu, and G. Kruszewski. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, volume 1, pages 238–247, 2014.

S. Bartunov, D. Kondrashkin, A. Osokin, and D. Vetrov. Breaking Sticks and Ambiguities with Adaptive Skip-gram. *Proc. of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

I. Beltagy, S. Roller, P. Cheng, K. Erk, and R. J. Mooney. Representing Meaning with a Combination of Logical and Distributional Models. *Computational Linguistics*, 42(4): 763–808, 2016.

E. M. Bender. Linguistically naïve!= language independent: Why NLP needs linguistic typology. In *Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous?*, pages 26–32. Association for Computational Linguistics, 2009.

E. M. Bender. On Achieving and Evaluating Language-Independence in NLP. *Linguistic Issues in Language Technology*, 6(3):1–26, 2011.

Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A Neural Probabilistic Language Model. *Journal of Machine Learning Research (JMLR)*, 3:1137–1155, 2003.

Y. Bengio, A. C. Courville, and P. Vincent. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1798–1828, 2013.

A. Bennett, T. Baldwin, J. H. Lau, D. McCarthy, and F. Bond. LexSemTM: A Semantic Dataset Based on All-words Unsupervised Sense Distribution Learning. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2016.

P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics (TACL)*, 5, 2017.

K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data.* ACM, 2008.

F. Bond and R. Foster. Linking and Extending an Open Multilingual Wordnet. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2013.

G. Bouma. Normalized (Pointwise) Mutual Information in Collocation Extraction. *Proceedings of GSCL*, pages 31–40, 2009.

P. Brown, S. D. Pietra, V. D. Pietra, and R. Mercer. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 1993.

P. F. Brown, V. J. D. Pietra, P. V. de Souza, J. C. Lai, and R. L. Mercer. Class-Based n-gram Models of Natural Language. *Computational Linguistics*, 18(4):467–479, 1992.

J. A. Bullinaria and J. P. Levy. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, pages 510–526, 2007.

R. Bunescu and M. Paşca. Using Encyclopedic Knowledge for Named Entity Disambiguation. In *Proc. of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2006.

C. Callison-Burch. *Paraphrasing and Translation.* PhD thesis, University of Edinburgh, 2007.

C. Callison-Burch, P. Koehn, C. Monz, and O. F. Zaidan. Findings of the 2011 Workshop on Statistical Machine Translation. In *WMT Shared Task*, 2011.

J. Camacho-Collados and M. T. Pilehvar. From Word to Sense Embeddings: A Survey on Vector Representations of Meaning. *CoRR*, abs/1805.04032, 2018.

J. Camacho-Collados, M. T. Pilehvar, and R. Navigli. NASARI: Integrating Explicit Knowledge and Corpus Statistics for A Multilingual Representation of Concepts and Entities. *Artificial Intelligence*, 240:36–64, 2016.

V. I. S. Carmona and S. Riedel. How Well Can We Predict Hypernyms from Word Embeddings? A Dataset-Centric Analysis. *Proc. of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, page 401, 2017.

M. Carpuat, Y. Vyas, and X. Niu. Detecting Cross-Lingual Semantic Divergence for Neural Machine Translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 69–79, Vancouver, August 2017. Association for Computational Linguistics.

R. Caruana. Multitask Learning. In *Learning to Learn*, pages 95–133. Springer, 1998.

A. Cauchy. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536–538, 1847.

M.-W. Chang, L. Ratinov, and D. Roth. Guiding Semi-Supervision with Constraint-Driven Learning. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 2007.

M.-W. Chang, L. Ratinov, D. Roth, and V. Srikumar. Importance of Semantic Representation: Dataless Classification. In *Proc. of the Conference on Artificial Intelligence (AAAI)*, September 2008.

M.-W. Chang, D. Goldwasser, D. Roth, and Y. Tu. Unsupervised Constraint Driven Learning For Transliteration Discovery. In *Proc. of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2009.

M.-W. Chang, L. Ratinov, and D. Roth. Structured Learning with Constrained Conditional Models. *Machine Learning*, 88(3):399–431, June 2012.

H.-H. Chen, W.-C. Lin, C. Yang, and W.-H. Lin. Translating-Transliterating Named Entities for Multilingual Information Access. *Journal of the Association for Information Science and Technology*, 57(5), 2006.

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

E. Chersoni, E. Santus, A. Lenci, P. Blache, and C.-R. Huang. Representing Verbs with Rich Contexts: an Evaluation on Verb Similarity. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2016.

T. Chklovski and P. Pantel. VerbOcean: Mining the Web for Fine-grained Semantic Verb Relations. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2004.

K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, Oct. 2014.

J. D. Choi, J. R. Tetreault, and A. Stent. It Depends: Dependency Parser Comparison Using A Web-based Evaluation Tool. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2015.

C. Christodouloupoulos and M. Steedman. A Massively Parallel Corpus: The Bible in 100 Languages. In *Proc. of the International Conference on Language Resources and Evaluation (LREC)*, pages 375–395, 2015.

K. W. Church and P. Hanks. Word Association Norms, Mutual Information, and Lexicography. *Computational Linguistics*, 16(1):22–29, Mar. 1990. ISSN 0891-2017.

D. Clarke. Context-theoretic Semantics for Natural Language: an Overview. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics.* Association for Computational Linguistics, 2009.

J. Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

R. Collobert and J. Weston. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Proc. of ICML*, 2008.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12: 2493–2537, 2011.

R. Cotterell, C. Kirov, J. Sylak-Glassman, D. Yarowsky, J. Eisner, and M. Hulden. The SIGMORPHON 2016 Shared Task—Morphological Reinflection. In *Proc. of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 2016.

R. Cotterell, E. Vylomova, H. Khayrallah, C. Kirov, and D. Yarowsky. Paradigm Completion for Derivational Morphology. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, Sept. 2017.

J. Coulmance, J.-M. Marty, G. Wenzek, and A. Benhalloum. Trans-gram, Fast Cross-lingual Word-embeddings. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2015.

S. Cucerzan. Large-scale Named Entity Disambiguation based on Wikipedia Data. In *Proceedings of EMNLP-CoNLL*, 2007.

I. Dagan and A. Itai. Word Sense Disambiguation using a Second Language Monolingual Corpus. *Computational Linguistics*, 1994.

I. Dagan, O. Glickman, and B. Magnini. The PASCAL Recognising Textual Entailment Challenge. In *MLCW*, 2005.

I. Dagan, D. Roth, M. Sammons, and F. M. Zanzoto. Recognizing Textual Entailment: Models and Applications. *Synthesis Lectures on Human Language Technologies*, 2013.

K. Darwish. Named Entity Recognition using Cross-lingual Resources: Arabic as an Example. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2013.

G. de Melo and G. Weikum. Towards a Universal Wordnet by Learning from Combined Evidence. In *Proc. of the ACM Conference on Information and Knowledge Management (CIKM)*, 2009.

S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 1990.

T. Demeester, T. Rocktäschel, and S. Riedel. Lifted Rule Injection for Relation Embeddings. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, November 2016.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*, 2018.

P. Dhillon, J. Rodu, D. Foster, and L. Ungar. Two step CCA: A New Spectral Method for Estimating Vector Models of Words. In *Proc. of the International Conference on Machine Learning (ICML)*, 2012.

P. S. Dhillon, D. P. Foster, and L. H. Ungar. Eigenwords: Spectral Word Embeddings. *Journal of Machine Learning Research (JMLR)*, 16:3035–3078, 2015.

M. Diab and P. Resnik. An Unsupervised Method for Word Sense Tagging using Parallel Corpora. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2002.

X. Duan, R. E. Banchs, M. Zhang, H. Li, and A. Kumaran, editors. *Proc. of the Fifth Named Entity Workshop.* Association for Computational Linguistics, July 2015.

L. Duong, H. Afshar, D. Estival, G. Pink, P. Cohen, and M. Johnson. Multilingual Semantic Parsing And Code-Switching. In *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*, August 2017a.

L. Duong, H. Kanayama, T. Ma, S. Bird, and T. Cohn. Multilingual Training of Crosslingual Word Embeddings. In *Proc. of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 894–904, Valencia, Spain, Apr. 2017b. Association for Computational Linguistics.

G. Durrett and D. Klein. A Joint Model for Entity Analysis: Coreference, Typing, and Linking. *Transactions of the Association for Computational Linguistics*, 2:477–490, 2014.

C. Dyer. Notes on Noise Contrastive Estimation and Negative Sampling. *arXiv preprint arXiv:1410.8251*, 2014.

C. Dyer, V. Chahuneau, and N. A. Smith. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, 2013.

C. Eckart and G. Young. The Approximation of One Matrix by Another of Lower Rank. *Psychometrika*, 1(3):211–218, 1936.

A. Eisele and Y. Chen. MultiUN: A Multilingual Corpus from United Nation Documents. In *Proc. of the International Conference on Language Resources and Evaluation (LREC)*, 2010.

J. L. Elman. Finding Structure in Time. *Cognitive Science*, 14(2):179–211, 1990.

O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised Named-Entity Extraction from the Web: An Experimental Study. *Artificial Intelligence*, 165(1):91–134, 2005.

Y. Even-Zohar and D. Roth. A classification approach to word prediction. In *Proc. of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 124–131, 2000.

Y. Even-Zohar, D. Roth, and D. Zelenko. Word clustering via classification. In *Proc. of the Bar-Ilan Symposium on Foundations of Artificial Intelligence (BISFAI)*, Bar-Ilan, Israel, 1999.

S. Evert. *The Statistics of Word Cooccurrences: Word Pairs and Collocations*. PhD thesis, University of Stuttgart, 2005.

S. Evert. Corpora and Collocations. *Corpus linguistics. An International Handbook*, 2: 1212–1248, 2008.

M. Faruqui and C. Dyer. Improving Vector Space Word Representations Using Multilingual Correlation. In *Proc. of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2014.

M. Faruqui, J. Dodge, S. K. Jauhar, C. Dyer, E. Hovy, and N. A. Smith. Retrofitting Word Vectors to Semantic Lexicons. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, 2015a.

M. Faruqui, Y. Tsvetkov, D. Yogatama, C. Dyer, and N. A. Smith. Sparse Overcomplete Word Vector Representations. In *Proceedings of ACL-IJCNLP*, July 2015b.

M. Faruqui, Y. Tsvetkov, G. Neubig, and C. Dyer. Morphological Inflection Generation Using Character Sequence to Sequence Learning. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, June 2016.

T. S. Ferguson. A Bayesian Analysis of Some Nonparametric Problems. *The Annals of Statistics*, 1973.

A. Finch, L. Liu, X. Wang, and E. Sumita. Neural Network Transduction Models in Transliteration Generation. In *Proc. of the Fifth Named Entity Workshop*, 2015.

A. Finch, L. Liu, X. Wang, and E. Sumita. Target-Bidirectional Neural Models for Machine Transliteration. In *Proc. of the Sixth Named Entity Workshop*, 2016.

J. R. Firth. The Technique of Semantics. *Transactions of the Philological Society*, 34(1): 36–73, 1935.

J. R. Firth. A Synopsis of Linguistic Theory 1930-1955. *Studies in Linguistic Analysis*, pages 1–32, 1957.

J. L. Fleiss. Measuring Nominal Scale Agreement among Many Raters. *Psychological Bulletin*, 76(5):378, 1971.

M. Francis-Landau, G. Durrett, and D. Klein. Capturing Semantic Similarity for Entity Linking with Convolutional Neural Networks. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, 2016.

D. Freitag. Trained Named Entity Recognition using Distributional Clusters. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2004.

Y. Freund and R. E. Schapire. Large Margin Classification Using the Perceptron Algorithm. *Machine Learning*, 37(3):277–296, 1999.

R. Fu, J. Guo, B. Qin, W. Che, H. Wang, and T. Liu. Learning Semantic Hierarchies via Word Embeddings. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2014.

G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. Statistical semantics: Analysis of the potential performance of key-word information systems. *The Bell System Technical Journal*, 62(6):1753–1806, 1983.

E. Gabrilovich and S. Markovitch. Wikipedia-based Semantic Interpretation for Natural Language Processing. *Journal of AI Research (JAIR)*, 34(2):443, 2009.

O.-E. Ganea and T. Hofmann. Deep Joint Entity Disambiguation with Local Neural Attention. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2017.

S. Ganesh, S. Harsha, P. Pingali, and V. Verma. Statistical Transliteration for Cross Language Information Retrieval using HMM Alignment Model and CRF. In *Proceedings of the 2nd Workshop on Cross Lingual Information Access*, 2008.

J. Ganitkevitch, B. Van Durme, and C. Callison-Burch. PPDB: The Paraphrase Database. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, 2013.

M. Gardner, K. Huang, E. Papalexakis, X. Fu, P. Talukdar, C. Faloutsos, N. Sidiropoulos, and T. Mitchell. Translation Invariant Word Embeddings. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2016.

D. Garrette, K. Erk, and R. Mooney. Integrating Logical Representations with Probabilistic Information using Markov Logic. In *Proc. of the International Conference on Computational Semantics (IWCS)*, pages 105–114. Association for Computational Linguistics, 2011.

M. Geffet and I. Dagan. The Distributional Inclusion Hypotheses and Lexical Entailment. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2005.

J. Giles. Internet Encyclopaedias Go Head to Head, 2005.

R. Girju, A. Badulescu, and D. I. Moldovan. Automatic Discovery of Part-Whole Relations. *Computational Linguistics*, 32:83–135, 2006.

A. Globerson, N. Lazic, S. Chakrabarti, A. Subramanya, M. Ringaard, and F. Pereira. Collective Entity Resolution with Multi-Focal Attention. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2016.

Y. Goldberg. Neural Network Methods for Natural Language Processing. *Synthesis Lectures on Human Language Technologies*, 10(1):1–309, 2017.

T. Goldstein, C. Studer, and R. Baraniuk. A Field Guide to Forward-Backward Splitting with a FASTA Implementation. *arXiv eprint*, abs/1411.3, 2014.

G. Golub and W. Kahan. Calculating the Singular Values and Pseudo-inverse of a Matrix. *Journal of the SIAM*, 1965.

I. Goto, N. Kato, N. Uratani, and T. Ehara. Transliteration Considering Context Information based on the Maximum Entropy Method. In *Proc. of MT-Summit IX*, volume 125132, 2003.

T. Gottron, M. Anderka, and B. Stein. Insights into Explicit Semantic Analysis. In *Proc. of the ACM Conference on Information and Knowledge Management (CIKM)*, pages 1961–1964. ACM, 2011.

S. Gouws, Y. Bengio, and G. Corrado. Bilbowa: Fast Bilingual Distributed Representations without Word Alignments. In *Proc. of the International Conference on Machine Learning (ICML)*, 2015.

D. Graff. Arabic Gigaword 3rd Edition, LDC2003T40, 2007.

E. Grefenstette. Towards a Formal Distributional Semantics: Simulating Logical Calculi with Tensors. In *Proc. of the Joint Conference on Lexical and Computational Semantics (*SEM)*, June 2013.

E. Grefenstette and M. Sadrzadeh. Experimental Support for a Categorical Compositional Distributional Model of Meaning. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2011.

G. Grefenstette. Finding Semantic Similarity in Raw Text: The Deese Antonyms. In *Fall Symposium Series, Working Notes, Probabilistic Approaches to Natural Language*, pages 61–65, 1992.

J. Guo, W. Che, H. Wang, and T. Liu. Learning Sense-specific Word Embeddings By Exploiting Bilingual Resources. In *Proc. the International Conference on Computational Linguistics (COLING)*, 2014.

J. Guo, W. Che, D. Yarowsky, H. Wang, and T. Liu. Cross-lingual Dependency Parsing Based on Distributed Representations. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2015.

N. Gupta, S. Singh, and D. Roth. Entity Linking via Joint Encoding of Types, Descriptions, and Context. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

M. U. Gutmann and A. Hyvärinen. Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics. *Journal of Machine Learning Research (JMLR)*, 13(Feb):307–361, 2012.

B. Hachey, W. Radford, J. Nothman, M. Honnibal, and J. R. Curran. Evaluating Entity Linking with Wikipedia. *Artificial Intelligence*, 194:130–150, 2013.

L. Haizhou, Z. Min, and S. Jian. A Joint Source-Channel Model for Machine Transliteration. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2004.

H. Hajishirzi, L. Zilles, D. S. Weld, and L. Zettlemoyer. Joint Coreference Resolution and Named-Entity Linking with Multi-pass Sieves. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2013.

D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical Correlation Analysis: An Overview with Application to Learning Methods. *Neural Computation*, 16(12):2639–2664, 2004.

M. A. Hearst. Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*, 1992.

K. M. Hermann and P. Blunsom. Multilingual Models for Compositional Distributional Semantics. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2014a.

K. M. Hermann and P. Blunsom. Multilingual Distributed Representations without Word

Alignment. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2014b.

U. Hermjakob, K. Knight, and H. Daumé III. Name Translation in Statistical Machine Translation - Learning When to Transliterate. *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2008.

F. Hill, R. Reichart, and A. Korhonen. Simlex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation. *arXiv preprint arXiv:1408.3456*, 2014.

J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust Disambiguation of Named Entities in Text. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pages 782–792. Association for Computational Linguistics, 2011.

J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia (extended abstract). In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2013.

M. D. Hoffman, D. M. Blei, C. Wang, and J. W. Paisley. Stochastic Variational Inference. *Journal of Machine Learning Research (JMLR)*, 2013.

H. Hotelling. Relations Between Two Sets of Variates. *Biometrika*, 28(3/4):321–377, 1936. ISSN 00063444.

E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng. Improving Word Representations via Global Context and Multiple Word Prototypes. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2012.

L. Hubert and P. Arabie. Comparing Partitions. *Journal of Classification*, 1985.

R. Hwa, P. Resnik, A. Weinberg, C. Cabezas, and O. Kolak. Bootstrapping Parsers via Syntactic Projection across Parallel Texts. *Journal of Natural Language Engineering (NLE)*, 11:11–311, 2005.

A. Irvine. *Using Comparable Corpora to Augment Low Resource SMT Models.* PhD thesis, Johns Hopkins University, 2014.

A. Irvine and C. Callison-Burch. A Comprehensive Analysis of Bilingual Lexicon Induction. *Computational Linguistics*, 43(2):273–310, 2017.

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

N. A. Jaleel and L. S. Larkey. Statistical Transliteration for English-Arabic Cross Language Information Retrieval. In *Proc. of the ACM Conference on Information and Knowledge Management (CIKM)*, 2003.

S. K. Jauhar, C. Dyer, and E. Hovy. Ontologically Grounded Multi-sense Representation Learning for Semantic Vector Space Models. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, 2015.

K. S. Jeong, S.-H. Myaeng, J. S. Lee, and K.-S. Choi. Automatic Identification and Back-Transliteration of Foreign Words for Information Retrieval. *Information Processing & Management*, 35(4):523–540, 1999.

H. Ji and R. Grishman. Knowledge Base Population: Successful Approaches and Challenges. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 1148–1158. Association for Computational Linguistics, 2011.

H. Ji, J. Nothman, B. Hachey, and R. Florian. Overview of TAC-KBP2015 Tri-lingual Entity Discovery and Linking. In *Text Analysis Conference (TAC)*, 2015.

S. Jiampojamarn, G. Kondrak, and T. Sherif. Applying Many-to-Many Alignments and Hidden Markov Models to Letter-to-Phoneme Conversion. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, Apr. 2007.

S. Jiampojamarn, A. Bhargava, Q. Dou, K. Dwyer, and G. Kondrak. DirecTL: A Language-Independent Approach to Transliteration. In *Proc. of the 2009 Named Entities Workshop: Shared Task on Transliteration*, 2009.

S. Jiampojamarn, K. Dwyer, S. Bergsma, A. Bhargava, Q. Dou, M.-Y. Kim, and G. Kondrak. Transliteration Generation and Mining with Limited Training Resources. In *Proc. of the 2010 Named Entities Workshop*, 2010.

C. Jiang, H.-F. Yu, C.-J. Hsieh, and K.-W. Chang. Learning Word Embeddings for Low-Resource Languages by PU Learning. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, June 2018.

N. Kaji and M. Kitsuregawa. Using Hidden Markov Random Fields to Combine Distributional and Pattern-Based Word Clustering. In *Proc. the International Conference on Computational Linguistics (COLING)*, August 2008.

S. Karimi, F. Scholer, and A. Turpin. Machine Transliteration Survey. *ACM Computing Surveys*, 2011. doi: 10.1145/1922649.1922654.

D. Kartsaklis and M. Sadrzadeh. Distributional Inclusion Hypothesis for Tensor-based Composition. In *Proc. the International Conference on Computational Linguistics (COLING)*, Osaka, Japan, December 2016.

K. Kawakami and C. Dyer. Learning to Represent Words in Context with Multilingual Supervision. *Proc. of ICLR Workshop*, 2015.

A. Kendall, Y. Gal, and R. Cipolla. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

M. A. Khalid, V. Jijkoun, and M. De Rijke. The Impact of Named Entity Normalization on Information Retrieval for Question Answering. In *Proc. of the European Conference on Information Retrieval (ECIR)*, pages 705–710. Springer, 2008.

A. Kilgarriff. I don't believe in word senses. *Computers and the Humanities*, 1997.

S. Kim, K. Toutanova, and H. Yu. Multilingual Named Entity Recognition using Parallel Data and Metadata from Wikipedia. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2012.

D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2014.

A. Klementiev and D. Roth. Named Entity Transliteration and Discovery in Multilingual Corpora. In *Learning Machine Translation*. MIT Press, 2008.

A. Klementiev, I. Titov, and B. Bhattarai. Inducing Crosslingual Distributed Representations of Words. In *Proc. the International Conference on Computational Linguistics (COLING)*, 2012.

K. Knight and J. Graehl. Machine Transliteration. *Computational Linguistics*, 1998.

T. Kočiský, K. M. Hermann, and P. Blunsom. Learning Bilingual Word Representations by Marginalizing Alignments. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 224–229. Association for Computational Linguistics, 2014.

P. Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proc. of MT Summit*, 2005.

P. Koehn and R. Knowles. Six Challenges for Neural Machine Translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39. Association for Computational Linguistics, 2017.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics, 2007.

P. Kolachina, N. Cancedda, M. Dymetman, and S. Venkatapathy. Prediction of Learning Curves in Machine Translation. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 22–30, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

N. Kolitsas, O.-E. Ganea, and T. Hofmann. End-to-End Neural Entity Linking. In *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*, pages 519–529. Association for Computational Linguistics, 2018.

T. Koo, X. Carreras, and M. Collins. Simple Semi-supervised Dependency Parsing. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2008.

L. Kotlerman, I. Dagan, I. Szpektor, and M. Zhitomirsky-Geffet. Directional Distributional Similarity for Lexical Expansion. In *Proc. of the ACL-IJCNLP*, 2009.

L. Kotlerman, I. Dagan, I. Szpektor, and M. Zhitomirsky-Geffet. Directional Distributional Similarity for Lexical Inference. *Journal of Natural Language Engineering (NLE)*, 16(4): 359–389, 2010. ISSN 1351-3249.

Z. Kozareva and E. Hovy. A Semi-Supervised Method to Learn and Construct Taxonomies Using the Web. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pages 1110–1118, Cambridge, MA, October 2010. Association for Computational Linguistics.

A. Kunchukuttan, M. Khapra, G. Singh, and P. Bhattacharyya. Leveraging Orthographic Similarity for Multilingual Neural Transliteration. In *Transactions of the Association for Computational Linguistics (TACL)*, volume 6, 2018.

G. Lakoff and M. Johnson. *Metaphors we live by.* University of Chicago press, 1980.

G. Lakoff and M. Johnson. *Philosophy in the Flesh*, volume 4. New York: Basic Books, 1999.

T. K. Landauer and S. T. Dumais. A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. *Psychological Review*, 104(2):211, 1997.

K. Lang. Newsweeder: Learning to Filter Netnews. In *Proc. of the International Conference on Machine Learning (ICML)*, pages 331–339, 1995.

N. Lazic, A. Subramanya, M. Ringgaard, and F. Pereira. Plato: A Selective Context Model for Entity Resolution. *Transactions of the Association for Computational Linguistics*, 3: 503–515, 2015. ISSN 2307-387X.

R. Lebret and R. Collobert. Word Embeddings through Hellinger PCA. In *Proc. of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 482–490, Gothenburg, Sweden, April 2014. Association for Computational Linguistics.

D. B. Lenat. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

A. Lenci and G. Benotto. Identifying Hypernyms in Distributional Semantic Spaces. In *Proc. of the 6th Workshop on Semantic Evaluation*, 2012.

J. P. Levy and J. A. Bullinaria. Learning lexical properties from word usage patterns:

Which context words should be used? In *Connectionist models of learning, development and evolution*. Springer, 2001.

O. Levy and Y. Goldberg. Linguistic Regularities in Sparse and Explicit Word Representations. In *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*, 2014a.

O. Levy and Y. Goldberg. Dependency-Based Word Embeddings. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2014b.

O. Levy and Y. Goldberg. Neural Word Embedding as Implicit Matrix Factorization. In *Proc. of the Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 2177–2185, 2014c.

O. Levy, S. Remus, C. Biemann, and I. Dagan. Do Supervised Distributional Methods Really Learn Lexical Inference Relations? In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, 2015.

M. Lewis and M. Steedman. Combined Distributional and Logical Semantics. *Transactions of the Association of Computational Linguistics*, 1:179–192, 2013.

M. P. Lewis, editor. *Ethnologue – Languages of the World*. SIL International, 16th edition, 2009.

W. Lewoniewski, K. Wecel, and W. Abramowicz. Relative Quality and Popularity Evaluation of Multilingual Wikipedia Articles. In *Journal of Informatics*, 2017.

J. Li and D. Jurafsky. Do Multi-Sense Embeddings Improve Natural Language Understanding? *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2015.

Z. Li, C. Callison-Burch, C. Dyer, J. Ganitkevitch, S. Khudanpur, L. Schwartz, W. N. Thornton, J. Weese, and O. F. Zaidan. Joshua: An Open Source Toolkit for Parsing-based Machine Translation. In *Proc. of the Fourth Workshop on Statistical Machine Translation*, 2009.

P. Liang. Semi-Supervised Learning for Natural Language. Master's thesis, Massachusetts Institute of Technology, 2005.

D. Lin. Automatic Retrieval and Clustering of Similar Words. In *Proc. of COLING-ACL*, 1998a.

D. Lin. An Information-theoretic Definition of Similarity. In *Proc. of the International Conference on Machine Learning (ICML)*, 1998b.

D. Lin, S. Zhao, L. Qin, and M. Zhou. Identifying Synonyms among Distributionally Similar Words. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, volume 3, pages 1492–1493, 2003.

Y. Lin, X. Pan, A. Deri, H. Ji, and K. Knight. Leveraging Entity Linking and Related Language Projection to Improve Name Transliteration. In *Proc. of the Sixth Named Entity Workshop*, 2016.

X. Ling and D. S. Weld. Fine-Grained Entity Recognition. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2012. URL `http://aiweb.cs.washington.edu/ai/pubs/ling-aaai12.pdf`.

X. Ling, S. Singh, and D. S. Weld. Design Challenges for Entity Linking. *Transactions of the Association for Computational Linguistics*, 2015. URL `https://transacl.org/ojs/index.php/tacl/article/view/528/133`.

P. Liu, X. Qiu, and X. Huang. Learning Context-sensitive Word Embeddings with Neural Tensor Skip-gram Model. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2015a.

Y. Liu, Z. Liu, T.-S. Chua, and M. Sun. Topical Word Embeddings. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2015b.

A. Lopez and M. Post. Beyond bitext: Five open problems in machine translation. In *Proceedings of the EMNLP Workshop on Twenty Years of Bitext*, 2013.

A. Lu, W. Wang, M. Bansal, K. Gimpel, and K. Livescu. Deep Multilingual Correlation for Improved Word Embeddings. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, 2015.

H. P. Luhn. A Statistical Approach to Mechanized Encoding and Searching of Literary Information. *IBM Journal of Research and Development*, 1(4):309–317, 1957.

T. Luong, H. Pham, and C. D. Manning. Effective Approaches to Attention-based Neural Machine Translation. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pages 1412–1421, Lisbon, Portugal, Sept. 2015a. Association for Computational Linguistics.

T. Luong, H. Pham, and C. D. Manning. Bilingual Word Representations with Monolingual Quality in Mind. In *Proc. of the Workshop on Vector Space Modeling for NLP*, 2015b.

P. Makarov, T. Ruzsics, and S. Clematide. Align and Copy: UZH at SIGMORPHON 2017 Shared Task for Morphological Reinflection. In *Proc. of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, Aug. 2017.

M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

S. Martin, J. Liermann, and H. Ney. Algorithms for Bigram and Trigram Word Clustering. *Speech Communication*, 24(1):19–37, 1998.

A. Martins, M. Almeida, and N. A. Smith. Turning on the Turbo: Fast Third-Order Non-Projective Turbo Parsers. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2013.

D. Matthews. Machine Transliteration of Proper Names. *Master's Thesis, University of Edinburgh, Edinburgh, United Kingdom*, 2007.

J. Mayfield, D. Alexander, B. J. Dorr, J. Eisner, T. Elsayed, T. Finin, C. Fink, M. Freedman, N. Garera, P. McNamee, et al. Cross-Document Coreference Resolution: A Key Technology for Learning by Reading. In *AAAI Spring Symposium: Learning by Reading and Learning to Read*, volume 9, pages 65–70, 2009.

R. McDonald, S. Petrov, and K. Hall. Multi-source Transfer of Delexicalized Dependency Parsers. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2011.

R. McDonald, J. Nivre, Y. Quirmbach-Brundage, Y. Goldberg, D. Das, K. Ganchev, K. Hall, S. Petrov, H. Zhang, O. Täckström, C. Bedini, N. Bertomeu Castelló, and J. Lee. Universal Dependency Annotation for Multilingual Parsing. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2013.

P. McNamee, J. Mayfield, D. Lawrie, D. W. Oard, and D. S. Doermann. Cross-Language Entity Linking. In *Proc. of the International Joint Conference on Natural Language Processing (IJCNLP)*, 2011.

Q. McNemar. Note on the Sampling Error of the Difference between Correlated Proportions or Percentages. *Psychometrika*, 12(2):153–157, 1947.

O. Melamud, D. McClosky, S. Patwardhan, and M. Bansal. The Role of Context Types and Dimensionality in Learning Word Embeddings. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, 2016.

Y. Merhav and S. Ash. Design Challenges in Named Entity Transliteration. In *Proc. the International Conference on Computational Linguistics (COLING)*, 2018.

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

T. Mikolov, M. Karafiát, L. Burget, J. Cernocky, and S. Khudanpur. Recurrent neural network based language model. *Proc. of Interspeech*, 2010.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*, 2013a.

T. Mikolov, Q. V. Le, and I. Sutskever. Exploiting Similarities Among Languages for Machine Translation. *arXiv preprint arXiv:1309.4168*, 2013b.

G. A. Miller. WordNet: a Lexical Database for English. *Communications of the ACM*, 1995.

G. A. Miller, C. Leacock, R. Tengi, and R. T. Bunker. A Semantic Concordance. In *Proceedings of the workshop on Human Language Technology*, pages 303–308. Association for Computational Linguistics, 1993.

S. Miller, J. Guinness, and A. Zamanian. Name Tagging with Word Clusters and Discriminative Training. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, 2004.

S. Mirkin, I. Dagan, and M. Geffet. Integrating Pattern-based and Distributional Similarity Methods for Lexical Entailment Acquisition. In *Proceedings of COLING-ACL*, pages 579–586. Association for Computational Linguistics, 2006.

S. Mohammad, B. Dorr, and G. Hirst. Computing Word-Pair Antonymy. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2008.

W. Monroe, S. Green, and C. D. Manning. Word Segmentation of Informal Arabic with Domain Adaptation. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2014.

D. S. Munteanu and D. Marcu. ISI Arabic-English Automatically Extracted Parallel Text LDC2007T08, 2007.

B. Murphy, P. Talukdar, and T. Mitchell. Learning Effective and Interpretable Semantic Models using Non-Negative Sparse Embedding. *Proc. the International Conference on Computational Linguistics (COLING)*, pages 1933–1950, 2012.

V. Nair and G. E. Hinton. Rectified Linear Units improve Restricted Boltzmann Machines. In *Proc. of the International Conference on Machine Learning (ICML)*, pages 807–814, 2010.

M. Nasiruddin. A State of the Art of Word Sense Induction: A Way Towards Word Sense Disambiguation for Under-resourced Languages. *arXiv preprint arXiv:1310.1425*, 2013.

R. Navigli. Word Sense Disambiguation: A Survey. *ACM Computing Surveys (CSUR)*, 2009.

R. Navigli and S. P. Ponzetto. BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. *Artificial Intelligence*, 2012.

A. Neelakantan, J. Shankar, A. Passos, and A. McCallum. Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2014.

M. Negri, A. Marchetti, Y. Mehdad, L. Bentivogli, and D. Giampiccolo. Semeval-2012 Task 8: Cross-lingual Textual Entailment for Content Synchronization. *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, 2012.

M. Negri, A. Marchetti, Y. Mehdad, L. Bentivogli, and D. Giampiccolo. Semeval-2013 Task 8: Cross-lingual Textual Entailment for Content Synchronization. *Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, 2013.

H. T. Ng, B. Wang, and Y. S. Chan. Exploiting Parallel Texts for Word Sense Disambiguation: An Empirical Study. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2003.

D. Nguyen, M. Theobald, and G. Weikum. J-NERD: Joint Named Entity Recognition and Disambiguation with Rich Linguistic Features. *Transactions of the Association for Computational Linguistics (TACL)*, 4:215–229, 2016a.

T. H. Nguyen, N. Fauceglia, M. Rodriguez Muro, O. Hassanzadeh, A. Massimiliano Gliozzo, and M. Sadoghi. Joint Learning of Local and Global Features for Entity Linking via Neural Networks. In *Proc. the International Conference on Computational Linguistics (COLING)*, pages 2310–2320, 2016b.

G. Nicolai, B. Hauer, M. Salameh, A. St Arnaud, Y. Xu, L. Yao, and G. Kondrak. Multiple System Combination for Transliteration. In *Proc. of the Fifth Named Entity Workshop*, July 2015.

I. Niles and A. Pease. Towards a Standard Upper Ontology. In *Proceedings of the International Conference on Formal Ontology in Information Systems-Volume 2001*, pages 2–9. ACM, 2001.

S. Padó and M. Lapata. Constructing semantic space models from parsed corpora. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 128–135, Sapporo, Japan, July 2003. Association for Computational Linguistics.

S. Padó and M. Lapata. Dependency-Based Construction of Semantic Space Models. *Computational Linguistics*, 2007.

S. Pado, M. Galley, D. Jurafsky, and C. D. Manning. Robust machine translation evaluation with entailment features. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 2009.

X. Pan, B. Zhang, J. May, J. Nothman, K. Knight, and H. Ji. Cross-lingual Name Tagging and Linking for 282 Languages. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 1946–1958. Association for Computational Linguistics, 2017.

R. Parker. Chinese Gigaword 5th Edition, LDC2011T13, 2011.

J. Pasternack and D. Roth. Learning Better Transliterations. In *Proc. of the ACM Conference on Information and Knowledge Management (CIKM)*, 11 2009.

E. Pavlick, P. Rastogi, J. Ganitkevitch, B. V. Durme, and C. Callison-Burch. PPDB 2.0: Better Paraphrase Ranking, Fine-grained Entailment Relations, Word Embeddings, and Style Classification. *Proc. of ACL-IJCNLP*, pages 425–430, 2015.

K. Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

J. Pennington, R. Socher, and C. D. Manning. GloVe: Global Vectors for Word Representation. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2014.

M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep Contextualized Word Representations. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, pages 2227–2237. Association for Computational Linguistics, 2018.

S. P. Ponzetto and M. Strube. Exploiting Semantic Role Labeling, WordNet and Wikipedia for Coreference Resolution. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, 2006.

M. Potthast, B. Stein, and M. Anderka. A Wikipedia-based Multilingual Retrieval Model. In *Proc. of the European Conference on Information Retrieval (ECIR)*, pages 522–530, 2008.

J. M. Prager, J. Chu-Carroll, and K. Czuba. Use of WordNet Hypernyms for Answering What-Is Questions. In *TREC*, 2001.

R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, and B. Webber. The Penn Discourse Treebank 2.0. In *Proc. of the International Conference on Language Resources and Evaluation (LREC)*, 2008.

L. Qiu, Y. Cao, Z. Nie, Y. Yu, and Y. Rui. Learning Word Representation Considering Proximity and Ambiguity. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2014.

K. Radinsky, E. Agichtein, E. Gabrilovich, and S. Markovitch. A Word at a Time: Computing Word Relatedness using Temporal Semantic Analysis. In *Proc. of the International World Wide Web Conference (WWW)*, 2011.

A. Rahman and V. Ng. Coreference Resolution with World Knowledge. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 814–824. Association for Computational Linguistics, 2011.

J. Raiman and O. Raiman. DeepType: Multilingual Entity Linking by Neural Type System Evolution. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2018.

W. M. Rand. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

R. Rapp. Identifying Word Translations in Non-Parallel Texts. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 320–322, Cambridge, Massachusetts, USA, June 1995. Association for Computational Linguistics.

R. Rapp. Automatic Identification of Word Translations from Unrelated English and German Corpora. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 519–526. Association for Computational Linguistics, 1999.

M. S. Rasooli and M. Collins. Cross-Lingual Syntactic Transfer with Limited Resources. *Transactions of the Association for Computational Linguistics*, 5:279–293, 2017. ISSN 2307-387X.

M. S. Rasooli and J. R. Tetreault. Yara Parser: A Fast and Accurate Dependency Parser. *CoRR*, abs/1503.06733, 2015.

L. Ratinov and D. Roth. Design Challenges and Misconceptions in Named Entity Recognition. In *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*, 6 2009.

L. Ratinov and D. Roth. Learning-based Multi-Sieve Co-Reference Resolution with Knowledge. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2012.

L. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and Global Algorithms for Disambiguation to Wikipedia. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2011.

S. Ravi and K. Knight. Learning Phoneme Mappings for Transliteration without Parallel Data. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, June 2009.

S. Reddy and S. Waxmonsky. Substring-based Transliteration with Conditional Random Fields. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*, pages 92–95. Association for Computational Linguistics, 2009.

J. Reisinger and R. J. Mooney. Multi-Prototype Vector-Space Models of Word Meaning. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, 2010.

P. Resnik and D. Yarowsky. Distinguishing Systems and Distinguishing Senses: New Evaluation Methods for Word Sense Disambiguation. *Journal of Natural Language Engineering (NLE)*, 1999.

E. Riloff and J. Shepherd. A Corpus-based Approach for Building Semantic Lexicons. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 1997.

L. Rimell. Distributional Lexical Entailment by Topic Coherence. In *Proc. of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, April 2014.

P. M. Roget. *Roget's Thesaurus of English Words and Phrases*. Longman Group Limited, 1852.

S. Roller and K. Erk. Relations such as Hypernymy: Identifying and Exploiting Hearst Patterns in Distributional Vectors for Lexical Entailment. *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2016.

S. Roller, K. Erk, and G. Boleda. Inclusive yet Selective: Supervised Distributional Hypernymy Detection. *Proc. the International Conference on Computational Linguistics (COLING)*, 2014.

D. Roth. Learning to Resolve Natural Language Ambiguities: A Unified Approach. In *Proc. of the Conference on Artificial Intelligence (AAAI)*, pages 806–813, 1998.

H. Rubenstein and J. B. Goodenough. Contextual Correlates of Synonymy. *Commun. ACM*, 8(10):627–633, Oct. 1965. ISSN 0001-0782.

S. Ruder, I. Vulic, and A. Sogaard. A Survey of Cross-lingual Word Embedding Models. *Journal of AI Research (JAIR)*, 2018.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533, 1986.

M. Sahlgren. *The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations between Words in High-Dimensional Vector Spaces*. PhD thesis, Stockholm University, 2006.

G. Salton and C. Buckley. Term-weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 1988.

G. Salton, A. Wong, and C.-S. Yang. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11):613–620, 1975.

M. Sammons, V. Vydiswaran, and D. Roth. Recognizing Textual Entailment. In *Multilingual Natural Language Applications: From Theory to Practice*, pages 209–258. Prentice Hall, 5 2012.

E. Santus, A. Lenci, Q. Lu, and S. S. Im Walde. Chasing Hypernyms in Vector Spaces with Entropy. In *Proc. of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2014.

E. Santus, F. Yung, A. Lenci, and C.-R. Huang. EVALution 1.0: An Evolving Semantic Dataset for Training and Evaluation of Distributional Semantic Models. In *Proc. of the 4th Workshop on Linked Data in Linguistics (LDL-2015)*, pages 64–69, 2015.

T. Schnabel, I. Labutov, D. Mimno, and T. Joachims. Evaluation Methods for Unsupervised Word Embeddings. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2015.

P. Scholl, D. Böhnstedt, R. D. García, C. Rensing, and R. Steinmetz. Extended Explicit Semantic Analysis for Calculating Semantic Relatedness of Web Resources. In *European Conference on Technology Enhanced Learning*, pages 324–339. Springer, 2010.

H. Schütze and J. O. Pedersen. Information Retrieval Based on Word Senses. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, 1995.

R. Sennrich, B. Haddow, and A. Birch. Neural Machine Translation of Rare Words with Subword Units. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2016.

J. Sethuraman. A Constructive Definition of Dirichlet Priors. *Statistica Sinica*, 1994.

W. Shen, J. Wang, and J. Han. Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460, 2015.

H. Shi, C. Li, and J. Hu. Real Multi-Sense or Pseudo Multi-Sense: An Approach to Improve Word Representation. In *Proc. of the Workshop on Computational Linguistics for Linguistic Complexity (CL4LC)*, pages 79–88, Osaka, Japan, Dec. 2016. The COLING 2016 Organizing Committee.

V. Shwartz and I. Dagan. Adding Context to Semantic Data-Driven Paraphrasing. In *Proc. of the Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 108–113, Berlin, Germany, 2015.

V. Shwartz, Y. Goldberg, and I. Dagan. Improving Hypernymy Detection with an Integrated Pattern-based and Distributional Method. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2016.

V. Shwartz, E. Santus, and D. Schlechtweg. Hypernyms under Siege: Linguistically-motivated Artillery for Hypernymy Detection. In *Proc. of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2017.

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

A. Sil, G. Kundu, R. Florian, and W. Hamza. Neural Cross-lingual Entity Linking. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2018.

S. L. Smith, D. H. Turban, S. Hamblin, and N. Y. Hammerla. Offline Bilingual Word Vectors, Orthogonal Transformations and the Inverted Softmax. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2017.

R. Snow, D. Jurafsky, and A. Y. Ng. Learning Syntactic Patterns for Automatic Hypernym Discovery. In *Proc. of the Conference on Advances in Neural Information Processing Systems (NIPS)*, 2004.

R. Snow, D. Jurafsky, and A. Y. Ng. Semantic Taxonomy Induction from Heterogenous Evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 801–808. Association for Computational Linguistics, 2006.

R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2013.

A. Søgaard, v. Agić, H. Martínez Alonso, B. Plank, B. Bohnet, and A. Johannsen. Inverted Indexing for Cross-lingual NLP. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2015.

Y. Song, S. Upadhyay, H. Peng, and D. Roth. Cross-lingual Dataless Classification for Many Languages. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

P. Sorg and P. Cimiano. Exploiting Wikipedia for Cross-lingual and Multilingual Information Retrieval. *Data & Knowledge Engineering*, 74:26–45, 2012.

K. Sparck Jones. A Statistical Interpretation of Term Specificity and its Application in Retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

C. Spearman. The Proof and Measurement of Association between Two Things. *The American Journal of Psychology*, 15(1):72–101, 1904.

R. Speer, J. Chin, and C. Havasi. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, pages 4444–4451, 2017.

V. I. Spitkovsky and A. X. Chang. A Cross-Lingual Dictionary for English Wikipedia Concepts. In *Proc. of the International Conference on Language Resources and Evaluation (LREC)*, 2012.

R. Sproat, T. Tao, and C. Zhai. Named Entity Transliteration with Comparable Corpora. In *Proc. of COLING-ACL*, 2006.

N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A

Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research (JMLR)*, 15(1):1929–1958, 2014.

J. H. Steiger. Tests for Comparing Elements of a Correlation Matrix. *Psychological Bulletin*, 87(2):245, 1980.

K. Stratos, D. kyum Kim, M. Collins, and D. J. Hsu. A Spectral Algorithm for Learning Class-Based n-gram Models of Natural Language. In *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2014.

A. Subramanian, D. Pruthi, H. Jhamtani, T. Berg-Kirkpatrick, and E. Hovy. SPINE: SParse Interpretable Neural Embeddings. *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2018.

H. Sun, H. Ma, W.-t. Yih, C.-T. Tsai, J. Liu, and M.-W. Chang. Open Domain Question Answering via Semantic Enrichment. In *Proc. of the International World Wide Web Conference (WWW)*, pages 1045–1055. International World Wide Web Conferences Steering Committee, 2015.

R. H. Susanto and W. Lu. Neural Architectures for Multilingual Semantic Parsing. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, July 2017.

S. Šuster, I. Titov, and G. van Noord. Bilingual Learning of Multi-sense Embeddings with Discrete Autoencoders. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, San Diego, USA, 2016.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to Sequence Learning with Neural Networks. In *Proc. of the Conference on Advances in Neural Information Processing Systems (NIPS)*, 2014.

O. Täckström, R. McDonald, and J. Uszkoreit. Cross-lingual Word Clusters for Direct Transfer of Linguistic Structure. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, 2012.

J. Tiedemann. Parallel Data, Tools and Interfaces in OPUS. In *Proc. of the International Conference on Language Resources and Evaluation (LREC)*, 2012.

S. Tratz and E. Hovy. A Fast, Accurate, Non-Projective, Semantically-Enriched Parser. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pages 1257–1268. Association for Computational Linguistics, 2011.

C.-T. Tsai and D. Roth. Cross-lingual Wikification Using Multilingual Embeddings. In *Proc. of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 6 2016.

C.-T. Tsai and D. Roth. Learning Better Name Translation for Cross-Lingual Wikification. In *Proc. of the Conference on Artificial Intelligence (AAAI)*, 2 2018.

H. Tseng, P. Chang, G. Andrew, D. Jurafsky, and C. Manning. A Conditional Random Field Word Segmenter for Sighan Bakeoff 2005. In *Proc. of the SIGHAN Workshop on Chinese Language Processing (SIGHAN)*, 2005.

Y. Tsvetkov, M. Faruqui, W. Ling, G. Lample, and C. Dyer. Evaluation of Word Vector Representations by Subspace Alignment. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2015.

J. Turian, L. Ratinov, and Y. Bengio. Word Representations: A Simple and General Method for Semi-supervised Learning. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2010.

P. D. Turney and S. M. Mohammad. Experiments with Three Approaches to Recognizing Lexical Entailment. *Journal of Natural Language Engineering (NLE)*, 2015.

P. D. Turney and P. Pantel. From Frequency to Meaning : Vector Space Models of Semantics. *Journal of AI Research (JAIR)*, pages 141–188, 2010.

S. Upadhyay, M. Faruqui, C. Dyer, and D. Roth. Cross-lingual Models of Word Embeddings: An Empirical Comparison. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2016.

S. Upadhyay, K.-W. Chang, M. Taddy, A. Kalai, and J. Zou. Beyond Bilingual: Multi-sense Word Embeddings using Multilingual Context. In *Workshop on Representation Learning for NLP (RepL4NLP) at ACL*, 2017. **Best Paper Award**.

S. Upadhyay, N. Gupta, and D. Roth. Joint Multilingual Supervision for Cross-lingual Entity Linking. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2018a.

S. Upadhyay, J. Kodner, and D. Roth. Bootstrapping Transliteration with Constrained Discovery for Low-Resource Languages. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2018b.

S. Upadhyay*, Y. Vyas*, M. Carpaut, and D. Roth. Robust Cross-lingual Hypernymy Detection using Dependency Context. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, 2018.

A. Ushioda. Hierarchical Clustering of Words. In *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*, pages 1159–1162. Association for Computational Linguistics, 1996.

J. Uszkoreit and T. Brants. Distributed Word Clustering for Large Scale Class-based Language Modeling in Machine Translation. *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2008.

P. Virga and S. Khudanpur. Transliteration of Proper Names in Cross-lingual Information

Retrieval. In *Proc. of the Workshop on Multilingual and Mixed-Language Named Entity Recognition*, 2003.

P. Vossen, E. Laparra, I. Aldabe, and G. Rigau. Interoperability of Cross-lingual and Cross-document Event Detection. In *Proc. of the 3rd Workshop on EVENTS at the NAACL-HLT*, 2015.

D. Vrandečić. Wikidata: A New Platform for Collaborative Data Collection. In *Proc. of the International World Wide Web Conference (WWW)*, pages 1063–1064. ACM, 2012.

I. Vulić. Cross-Lingual Syntactically Informed Distributed Word Representations. In *Proc. of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2017.

I. Vulić and M.-F. Moens. Cross-Lingual Semantic Similarity of Words as the Similarity of Their Semantic Word Responses. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, 2013.

I. Vulić and M.-F. Moens. Bilingual Word Embeddings from Non-Parallel Document-Aligned Data Applied to Bilingual Lexicon Induction. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2015.

Y. Vyas and M. Carpuat. Sparse Bilingual Word Representations for Cross-lingual Lexical Entailment. *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, 2016.

Y. Vyas and M. Carpuat. Detecting Asymmetric Semantic Relations in Context: A Case Study on Hypernymy Detection. In *Proc. of the Joint Conference on Lexical and Computational Semantics (*SEM)*, 2017.

H. Wang, J. G. Zheng, X. Ma, P. Fox, and H. Ji. Language and Domain Independent Entity Linking with Quantified Collective Validation. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2015.

W. Weaver. Translation. In W. N. Locke and A. D. Boothe, editors, *Machine Translation of Languages*, pages 15–23. MIT Press, Cambridge, MA, 1955. Reprinted from a memorandum written by Weaver in 1949.

J. Weeds and D. Weir. A General Framework for Distributional Similarity. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2003.

J. Weeds, D. Weir, and D. McCarthy. Characterising Measures of Lexical Distributional Similarity. In *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*, page 1015. Association for Computational Linguistics, 2004.

E. J. Williams. The Comparison of Regression Variables. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 396–399, 1959.

S. M. Wong, W. Ziarko, and P. C. Wong. Generalized Vector Spaces Model in Information Retrieval. In *Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 18–25. ACM, 1985.

Z. Wu and C. L. Giles. Sense-Aware Semantic Analysis : A Multi-Prototype Word Representation Model Using Wikipedia. *Proc. of the National Conference on Artificial Intelligence (AAAI)*, pages 2188–2194, 2015.

N. Xue. Labeling Chinese Predicates with Semantic Roles. *Computational Linguistics*, 34 (2):225–255, 2008.

D. Yarowsky. One Sense per Collocation. In *Proceedings of the Workshop on Human Language Technology*, pages 266–271. Association for Computational Linguistics, 1993.

D. Yarowsky. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 1995.

D. Yarowsky and G. Ngai. Inducing Multilingual POS Taggers and NP Bracketers via Robust Projection across Aligned Corpora. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, 2001.

K. Y. Yeung and W. L. Ruzzo. Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774, 2001.

W.-t. Yih, G. Zweig, and J. C. Platt. Polarity Inducing Latent Semantic Analysis. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2012.

W.-t. Yih, M.-W. Chang, C. Meek, and A. Pastusiak. Question Answering Using Enhanced Lexical Semantic Models. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2013.

W.-t. Yih, M.-W. Chang, X. He, and J. Gao. Semantic Parsing via Staged Query Graph Generation: Question Answering with Knowledge Base. In *Proceedings of ACL-IJCNLP*, Beijing, China, July 2015.

D. Zeman and P. Resnik. Cross-Language Parser Adaptation between Related Languages. In *Proc. of the International Joint Conference on Natural Language Processing (IJCNLP)*, 2008.

S. Zhang, K. Duh, and B. Van Durme. Cross-lingual Semantic Parsing, 2018a.

S. Zhang, X. Ma, R. Rudinger, K. Duh, and B. Van Durme. Cross-lingual Decompositional Semantic Parsing. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, October-November 2018b.

G. K. Zipf. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Addison-Wesley, 1949.

W. Y. Zou, R. Socher, D. Cer, and C. D. Manning. Bilingual Word Embeddings for Phrase-Based Machine Translation. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2013.

Y. Zou and W. Lu. Learning Cross-lingual Distributed Logical Representations for Semantic Parsing. In *Proc. of the Annual Meeting of the Association of Computational Linguistics (ACL)*, July 2018.