Multilingual Word Embeddings and Their Utility In Cross-lingual Learning

...or lack thereof

By

Artur Kulmizev

Advised by

Eneko Agirre and Gertjan van Noord

University of Groningen
University of the Basque Country

A Master's thesis submitted to the University of Groningen and the University of the Basque Country in accordance with the requirements of the degrees of Master's of Arts and Master's of Science at each university, respectively.

Word Count: 21,392

August 2018

Abstract

Word embeddings - dense vector representations of a word's distributional semantics - are an indespensable component of contemporary natural language processing (NLP). Bilingual embeddings, in particular, have attracted much attention in recent years, given their inherent applicability to cross-lingual NLP tasks, such as Part-of-speech tagging and dependency parsing. However, despite recent advancements in bilingual embedding mapping, very little research has been dedicated to aligning embeddings *multilingually*, where word embeddings for a variable amount of languages are oriented to a single vector space. Given a proper alignment, one potential use case for multilingual embeddings is cross-lingual transfer learning, where a machine learning model trained on resource-rich languages (e.g. Finnish and Estonian) can "transfer" its salient features to a related language for which annotated resources are scarce (e.g. North Sami). The effect of the quality of this alignment on downstream cross-lingual NLP tasks has also been left largely unexplored, however.

With this in mind, our work is motivated by two goals. First, we aim to leverage existing supervised and unsupervised methods in bilingual embedding mapping towards inducing high-quality multilingual embeddings. To this end, we propose three algorithms (one supervised, two unsupervised) and evaluate them against a completely supervised bilingual system and a commonly employed baseline approach. Second, we investigate the utility of multilingual embeddings in two common cross-lingual transfer learning scenarios: POS-tagging and dependency parsing. To do so, we train a joint POS-tagger/dependency parser on Universal Dependencies treebanks for a variety of Indo-European languages and evaluate it on other, closely related languages. Although we ultimately observe that, in most settings, multilingual word embeddings themselves do not induce a cross-lingual signal, our experimental framework and results offer many insights for future cross-lingual learning experiments.

ACKNOWLEDGEMENTS

This work would be not be possible without the support and encouragement of a number of individuals. I'd like to reserve this page to thank them accordingly.

Thank you Eneko for helping to shape this project and providing invaluable advice throughout its entirety.

Thank you Gertjan for your help and encouragement, especially on short notice.

Thank you Mikel and Gorka, for imparting vital technical advice and helping me out of some very difficult situations.

Thank you Gosse, for helping to make the entire moving-to-Europe process as easy it as it can be.

Thank you Bobbye, for fielding what seemed like countless funding requests.

Thanks MGAD lads, for the banter.

Thank you Ed, for bearing with me throughout all of this, especially from afar ♥.

Thanks Dad, Mom, Grandma, Diana, and Mark, for everything. I miss you guys always.

AUTHOR'S DECLARATION

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED:.. DATE: 29/8/18 ....

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The practice of representing words as low-dimensional numeric vectors has existed for a long time within the domain of Natural Language Processing (NLP). The idea that governs this approach is that of the *distributional hypothesis* - i.e. that the meanings of individual words can be derived from the contexts in which they appear. For example, given the following three sentences, one can infer that the ambiguous **bardiwac** refers to a probably alcoholic, wine-like beverage (Evert, 2010).

- *He handed her her glass of* ***bardiwac****.*
- *Beef dishes are made to complement the* ***bardiwacs****.*
- *The drinks were delicious: blood-red* ***bardiwac*** *as well as light, sweet Rhenish.*

With the distributional hypothesis in mind, an appropriate algorithm can be applied to a large, unannotated corpus in order to: first, learn the meanings of words given their various contexts and second, encode these meanings into numeric vector representations. A corpus' entire vocabulary can then be assumed to comprise a single semantic vector space, which enables semantic inference by proxy of linear-algebraic operations. For example, given word vectors for *dog*, *cat*, and *horse*, one could calculate the pairwise cosine similarity between $\vec{dog}, \vec{cat}, \vec{horse}$ in order to learn that *dog* and *cat* are likely more similar to each other than they are to *horse*.

A recent topic of interest within the realm of computational distributional semantics has been that of *bilingual vector spaces*, which intend to align the meanings of words across two languages. Though numerous methods for accomplishing this have been proposed, the most common (and arguably most reliable) is to learn a mapping matrix W that aligns a monolingual source vector space X to a target Z via a bilingual dictionary (Artetxe et al., 2016). This is accomplished via the *isomorphism assumption*, which posits that, by nature of the distributional hypothesis, contexts for any given word in one language will be similar to contexts for the same word in another. In other words, the shape of an entire source language vector space X will be approximately similar to a target language vector space Z. Given a proper alignment between the two, then, the aforementioned semantic inference can be extended cross-linguistically - e.g. within aligned English and Spanish spaces EN,ES, the vector with the highest cosine similarity to $\vec{dog} \in EN$ would likely be $\vec{perro} \in ES$.

Though promising, the vast majority of research in this latter domain has been focused on the bilingual case. The extension of these methods to a multilingual scenario has largely been overlooked due to one main assumption: if embeddings for two different languages are mapped to a common pivot language, cross-lingual inference between them is (generally) enabled as a proxy as well. As such, the need for producing multilingual embeddings (where three or more languages are mapped to a single, shared space) has long been addressed by this simple pivot approach. Of course, this is only possible if high-quality dictionaries between all languages to and the pivot language are available. Though recent developments in bilingual embedding mapping have foregone the use of dictionaries entirely (Conneau et al., 2017; Artetxe et al., 2018a), these methods are largely tailored for the bilingual case and remain entirely un-explored in multilingual scenarios.

Generally speaking, though a considerable body of research has been dedicated to computational distributional semantics over the last two decades, it was not until the advent of *deep learning* that word vectors became indispensable for most NLP processes. In broad terms, deep learning (or neural networks) can be characterized as a class of machine learning algorithms that aim to learn hidden feature representations of data via a series of non-linear weight-matrix transformations. In contemporary NLP, which typically favors deep neural networks in place of simple, linear classifiers, word vectors are often employed as part of a necessary initialization step, wherein each word is substituted for its corresponding vector. Doing so enriches a system with relevant lexical information for the task at hand, enabling it to learn from words' *meanings* instead of solely their functional aspects. This technique, in combination with many other design choices, has produced state-of-the-art performance in a multitude of NLP tasks, including, but not limited to language modeling, dependency parsing, question answering, and natural language understanding.

Despite deep learning's largely positive impact on NLP, however, its increasingly widespread adoption has attracted many strands of criticism. Chief among these is the perennial data problem - that, for most tasks, a deep learning model requires a vast amount of annotated samples in order to learn accurate representations. Though this is a general problem within the realm of supervised learning, it nonetheless more pervasive in deep learning. As such, in scenarios wherein a limited amount of resources is available (e.g. dependency parsing for a minority language such as North Sami), simpler, linear models are often employed in favor of deep ones (Ruder and Plank, 2018).

Another common criticism is that deep learning models are often trained with language-agnosticism in mind. In other words, when a novel architecture is typically proposed, it is only trained, evaluated, and fine-tuned on a single language (usually English), with the assumption that interested parties will retrain and optimize the same model for their language of choice. On one hand, this is logical - it is impossible to expect researchers to evaluate their models even on the world's major languages, for lack of time and resources alone. On the other hand, recent research has suggested that, when accounting for multilinguality, models can learn to generalize better across unseen input, learn more universal feature representations, and improve the model's performance overall.

The central aim of this thesis is to address these two latter concerns in the context of part-of-speech tagging and dependency parsing, employing multilingual word embeddings as the primary input representation. As such, we set out with two goals in mind. First, we aim to leverage existing supervised and unsupervised methods in bilingual embedding mapping towards inducing multilingual embeddings that are of higher quality than embeddings produced by the aforementioned pivot language approach. In doing so, we posit that we will enable cross-lingual transfer for the tasks in question. Second, we evaluate the effect of these alignments in various cross-lingual transfer learning settings, employing Danish, Italian and Slovenian as evaluation languages. In particular, we aim to address the following research questions:

1. *Can we produce better-quality multilingual embeddings than the oft-employed common language approach?*

2. *Can unsupervised mapping algorithms produce higher-quality multilingual embeddings than their dictionary-based, supervised counterparts?*

3. *Can we observe a beneficial cross-lingual signal in regards to the POS-tagging and dependency parsing tasks?*

4. *If so, does the cross-lingual signal persist when accounting for increasingly higher resource scenarios?*

This work is thus organized as follows:

- In **Chapter 2**, we provide an overview of prior research related to the topics explored by this work. First, we review past methods for generating vector spaces and aligning them multilingually. Second, we cover past approaches for cross-lingual learning as applied to the domain of NLP, focusing on the POS-tagging and dependency parsing.

- In **Chapter 3**, we provide an overview of our method for aligning monolingual embeddings to a shared multilingual space.

- In **Chapter 4**, we describe the data we employ in our embedding alignment experiments and share the results we obtain, along with an in-depth discussion.
- In **Chapter 5**, we describe our method for cross-lingual POS-tagging and dependency parsing enabled by multilingual embeddings. We provide an overview of our model as well as the experimental conditions under which we perform our evaluation.
- In **Chapter 6**, we report results for our cross-lingual experiments and discuss them in depth, along with their implications.
- In **Chapter 7**, we summarize our work and offer suggestions for extending our experiments in the future.

Chapter 2

Related Work

2.1 Distributional Semantic Models

The practice of representing word meanings as numeric vectors has a rich history in NLP. Though there are a multitude of motivations for doing so, perhaps the most intuitive is that vector spaces situate semantics within the realm of geometry, wherein mathematical notions of distance and similarity can exploited as a means of linguistic inference (Clark, 2015). Naturally, one can imagine that the meaning of the word *dog* is closer to *cat* than it is to *airplane*. Provided a proper encoding, one could then verify this computationally by calculating the cosine similarity between the vectors corresponding to all three words. Given two vectors a and b, cosine similarity is computed as follows:

$$\text{sim}(a,b) = \frac{ab}{\parallel a \parallel \parallel b \parallel} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}}$$

...where $\parallel a \parallel$ is the Euclidean norm of the vector.

Distributional Semantic Models (DSMs) intend to capture these semantic relations as they occur in a natural language corpus. When said corpus is large enough (e.g. Wikipedia, Common Crawl, or the concatenation of the two), the resulting DSM can be assumed to represent the distributional semantics of an entire language.

The algorithms employed for generating DSMs can be said to belong to one of two classes: *count-based* and *prediction-based* (a.k.a. word embeddings). The former entails generating a *term-context* matrix of word co-occurrences, weighting these co-occurence frequencies via an association score and applying a dimensionality reduction technique to the resultant matrix. The latter, on the other hand, applies a machine-learning algorithm (typically a neural network) towards the task of *predicting* a word given its context (or vice versa). The n-dimensional feature vector that is learned in this process then serves as an encoded representation of the word itself.

2.1.1 Count-based DSMs

Historically, the meanings of words have been able to be computationally approximated by virtue of the distributional hypothesis: "a word is characterized by the company it keeps" (Firth, 1957). Much of the early work in encoding word meaning in this fashion has been inspired by the process of Latent Semantic Analysis (LSA), which attempts to capture the inherent relationships between documents and the terms they contain (Deerwester et al., 1990). The principal motivation behind LSA is that words that could be considered similar in meaning will occur in similar documents (contexts) as well. LSA is performed by formulating a *term-document* frequency matrix, in which rows are represented by words in vocabulary V and columns are represented by documents. Since raw frequency fails to capture the intuition that some words are more characteristic of a document than others and, as such, weighs every word equally, an alternative weighting schema (typically *term frequency-inverse document frequency* or *tf-idf*) is applied to the matrix's entries.

After the construction of the *term-document* matrix, a dimensionality reduction technique is often applied in order to distill the (often) thousands of representative document columns into a smaller number of latent *topics* or *components*. This is typically accomplished by computing the Singular Value Decomposition (SVD) of *term-document* matrix. Put briefly, SVD factorizes the original matrix M into the form $M = U\Sigma V^T$, where the resulting factorization is employed in finding the rank-k matrix that best approximates M. According to Turney (2008), SVD in the context of LSA accomplishes three things:

1. Clusters words around k latent topics that are present in the data.
2. Reduces noise by distilling the matrix dimensions into the few that represent the true signal within the data.
3. Recognizes words as similar to each other when they appear in similar document contexts.

Though LSA functions more as a *topic modeling* approach, where the content of documents is the focus, the same principle can be applied towards building DSMs. In this case, the *term-document* matrix is reformulated as a *term-context* matrix, where *context* represents words occurring within a given context window (typically between 5 and 10 words) of the target term. Like in LSA, the raw frequencies that comprise the entries of the matrix are represented as an association score or weighting scheme (e.g. Positive Pointwise Mutual Information or Log-Likelihood Ratio (Evert, 2005)). Then, as in LSA, SVD or another dimensionality reduction technique is applied in order to avoid vector sparsity and reduce noise. The resulting vectors can thus be considered a representation of a word's meaning, as determined by the words that surround it. As such, this context-based approach is more of a direct interpretation of the distributional hypothesis than LSA, which captures topical instead of lexical similarity (Clark, 2015). Though the application of dimensionality reduction threatens to make the resulting vector dimensions un-interpretable, the utility of interpretability is entirely dependent on the task at hand.

2.1.2 Word2Vec

In sharp contrast to *count-based* matrix factorization, *prediction-based* approaches employ neural networks for predicting words based on the contexts in which they occur. Most of the early approaches in this direction follow a language-modeling objective, which seeks to maximize the probability of a word w_t given history h:

$$P(w_t|h) = \text{softmax}(\text{score}(w_t, h))$$

where score() computes the compatibility between w_t and h. In this case, language modeling is employed as a proxy task, wherein the features learned for the objective can be understood to capture information about words' distributional semantics. The architecture for this approach is fairly straightforward: the previous N words before the target word are encoded as a 1-of-V vector, where V is the size of the training corpus' vocabulary (this can also include future words $w_{t+1}, w_{t+2}, etc.$). This vector is projected onto a hidden layer P of dimensionality $N \times D$, which is itself projected to an output layer O of dimensionality $D \times V$. $\hat{w_t}$ is then computed via $\arg\max_{w \in V} \text{softmax}(O_w)$. In this case, the hidden layer H is extracted as a learned feature representation of w_t, where distributional information is "embedded" into a D-sized dense numeric vector.

Mikolov et al. (2013a) refer to this approach as the **Continuous Bag of Words** (CBOW) model, as the order of words in h is not relevant to the task of encoding context. An alternative to this approach is to invert the typical language modeling objective and predict instead the context h for word w_t. For example, for a sample sentence such as "the tiny dog barked loudly" and the target word "dog", the task (with a window size $n = 1$) would be to predict $P(\text{tiny}|\text{dog})$ and $P(\text{barked}|\text{dog})$. This approach is the natural counterpart to CBOW and is referred to as **Skip-gram** by Mikolov et al. (2013a). Canonically, both CBOW and SKIP-GRAM comprise a suite of DSM algorithms that is termed **word2vec**.

Though employing a language modeling objective towards learning low-dimensional dense word vectors is an intuitive approach, both approaches highlighted so far suffer from a dramatically high complexity per sample. Given a corpus as large as the English Wikipedia, the training procedure would be rendered staggeringly inefficient, as $P(w'|h)$ (or vice-versa) would need to be computed for all words in V at context h, during every training step. To address this problem, Mikolov et al. (2013c) propose the technique of *negative sampling*. In broad terms, negative sampling is motivated by the intuition that a full probabilistic model is not essential for the learning of word features. Instead, in the case of CBOW, the objective could be reformulated as training a binary classifier to differentiate between w_t and $\tilde{w}_t$, which is obtained from a set of k "noise" words, or *negative samples*. Formally, the goal of employing negative sampling is to maximize

$$J_{NEG} = \log O_\theta(D = 1|w_t, h) + k \underset{\tilde{w} \sim P_{noise}}{\mathbb{E}} [\log O_\theta(D = 0|\tilde{w}_t, h)]$$

where $\log O_\theta(D = 1|w_t, h)$ is the probability of word w_t in context h as observed in dataset D, calculated by means of the learned word embeddings θ, and $\log O_\theta(D = 0|\tilde{w}_t, h)$ is the probability of $\tilde{w}$ in k negative samples drawn from a noise distribution $\mathbb{E}$.

The combination of the CBOW and SKIP-GRAM models with negative sampling proved to be a turning point in the creation of DSMs. The significant reduction in computational complexity meant that DSMs could be trained on corpora that were several orders of magnitude larger than previously reported models, in a fraction of the time. As a result, the learned vector representations reported in Mikolov et al. (2013c) proved to be highly robust, given their performance on the analogical reasoning task introduced by Mikolov et al. (2013a). Furthermore, they also discovered that the encodings learned in the training process possessed a compositional quality that could be exploited via vector-arithmetic means. For example, the pairwise addition $\vec{russia}$ and $\vec{river}$ would yield a vector that is very close to the corresponding $\vec{volga}$ in the n-dimensional space.

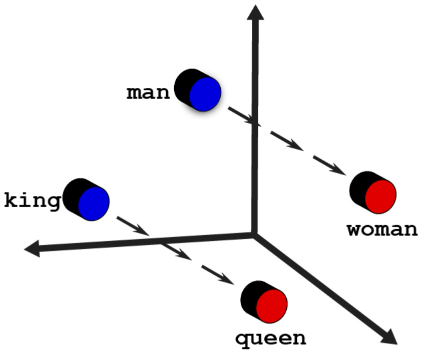


Figure 2.1: An example of analogical reasoning in latent semantic vector space.

The inherent quality of word2vec-produced DSMs is corroborated by Baroni et al. (2014), who conduct a comprehensive evaluation of count and prediction-based methods over a variety of tasks ranging from relatedness, synonymy, categorization, and analogy. Their results reveal that, in almost every case, WORD2VEC defeats the previous state of the art set by count-based models by a significant margin. As distributional semanticists, they concede to the apparent superiority of prediction-based models over count-based matrix factorization, with the caveat that their performance is highly contingent on proper parametrization (context window, negative samples, frequency subsampling treshold, vector dimension, etc.).

2.1.3 Morphologically-enriched DSMs

Given all of the successes of the WORD2VEC and prediction-based approaches to constructing DSMs, one of their perennial drawbacks is the inability to account for the inherent *structure* of words. Historically, most research concerning distributional semantics - much like all of NLP - has focused on English, which follows comparatively simple inflectional patterns. However, other languages, such as Basque or Finnish, feature highly complex agglutinative morphologies, making it possible for thousands of inflections and derivations of a root word form to exist. Since many such forms may not occur in the corpora on which DSMs are trained, this makes it difficult for token-level predictive approaches to learn proper representations of words. Ideally, then, a DSM should strive to capture both lexical and functional aspects of words in order to deem themselves truly language-universal.

There have been numerous attempts to incorporate morphology into distributional semantics in the recent past. For example, Cotterell and Schütze (2015) aimed to induce morphologically-aware word embeddings by extending the log-bilinear model via a joint objective of context and morphological tag prediction. Luong et al. (2013) followed a two-step process of obtaining morphological encodings of words by passing their segments through a Recursive Neural Network and then composing them with word-level learned via language modeling. In a much different approach, Schütze (1993) factorized a matrix of character 4-gram co-occurences via SVD, ultimately representing words as the addition of their constituent character 4-gram vectors. Wieting et al. (2016) sought to learn character n-gram based representations of words and sentences, training on a suite of paraphrase datasets. Using only this information, they reported then-new state-of-the-art results on the Sim-Lex999 dataset. However, though a promising approach, the fact that it was trained on a labeled dataset made it generally less versatile than language-modeling based methods.

Perhaps the most widely-adopted morphologically-enriched extension of WORD2VEC is FASTTEXT (Bojanowski et al., 2016). At its core, FASTTEXT is built upon the SKIP-GRAM with negative sampling algorithm described above, which employs the score function $s(w_t, w_c) = p_{w_t}^{\top} v_{w_c}$ where p_{w_t} and v_{w_c} correspond to the current word and context word vectors, respectively. In this case, however, words are not represented solely as their token vector but also as a bag of character n-grams. For instance, a word such as *water* could be decomposed into the following set of 3-grams:

```
<wa, wat, ate, ter, er>
```

...as well as the original token `<water>`. Here, the `<`,`>` brackets are added to signal the beginning and end of a word, respectively. It is important to note here that the 3-gram `ate` is distinct from the word token `<ate>`. With a range of character n-grams such as 3-to-6 (which the authors employ in their work), it is thus possible to approximate a word's inherent morphology - e.g. prefixes `<un`,

`<pre, <anti` or suffixes `ly>, ism>, able>`. If one imagines a dictionary $\mathscr{G}_{w_t} \subset 1...G$ where word w is decomposed into $G\ n-grams$, the score function becomes the following:

$$s(w_t, w_c) = \sum_{g \in \mathscr{G}_{w_t}} z_g^\top v_{w_c}$$

In their experiments, Bojanowski et al. (2016) report promising results on a suite of similarity and analogy tasks across different languages. When tested against the baselines of Wikipedia-trained CBOW and SKIP-GRAM embeddings, the subword-enriched SKIP-GRAM embeddings perform comparatively better. This is especially clear in the cases of Arabic, German, and Russian, the former of which follows a *root-and-pattern* concatenative morphology and the latter two decline to four and six noun cases, respectively. Bojanowski et al. (2016) also report improved scores on the English Rare Words dataset (Luong et al., 2013) as compared to the `word2vec` baselines, indicating that the similarity at the character level captured by FASTTEXT aids in learning representations for less-frequent words.

2.2 Bilingual Word Embeddings

Since the reported successes of WORD2VEC and its subsequent adoption as the (arguable) DSM-of-choice, much research has been devoted towards making word embeddings multilingual. Certainly, much work in this direction has been motivated by the increasing need for *cross-lingual* models, where knowledge could be learned, shared, and exploited across any variable number of languages. A natural application for such models is the low-resource scenario, where a model trained on a well-resourced language, such as Finnish, could transfer its features to a considerably lesser-resourced, related language, such as North Sami. For such models to exist, a shared representation of meaning, where words are aligned along a single semantic axis, is necessary. In terms of words embeddings, this would mean that representations of equivalent words or concepts in two different languages must themselves be roughly equivalent, thereby enabling cross-lingual inference across the two languages (e.g. $\vec{dog}_{EN} \approx \vec{perro}_{ES}$).

In recent years, research in cross-lingual word embeddings has (in very broad terms) followed one of two strands: sentence or document level training or word-level mapping. The first of these types of approaches is largely governed by the intuition that bilingual parallel corpora (aligned at either the sentence or document level) contain a cross-lingual signal that can be exploited when learning word embeddings. An example of such an approach is Gouws et al. (2015), who propose a Bilingual Bag-of-Words model (BilBOWA) that, for every word vector w_s in a source language sentence S, learns a transformation from w_t to the mean of the word representations in target language sentence T, with the objective of minimizing the distance between the two. Similarly, Luong et al. (2015) propose an extension of SKIP-GRAM that, given a sentence-aligned parallel corpus, attempts to predict not only the respective contexts of words in S and T, but also the contexts in T given words in S, and vice versa. Though these approaches are certainly theoretically sound and produce reasonable quality bilingual word representations in their own right, they are hindered by two crucial factors: A.) they require good-quality parallel corpora, which are expensive to produce and only exist for the world's major languages and B.) the sizes of such corpora are, in almost all cases, orders of magnitude smaller than the typical corpora on which word embeddings are trained, thus leading to less robust and accurate representations. For these reasons, we do not further consider such approaches and instead focus the remainder of this section on mapping methods.

2.2.1 Bilingual Embedding Mappings

The bilingual embedding mapping approach was first proposed by Mikolov et al. (2013b), who observe that the geometric orientation of words within vector spaces is similar across languages. In other words, numeral words in an English space share approximately the same distribution of distances from each other as do numeral words in Spanish. Given this observation, Mikolov et al. (2013b) posit that it is possible to obtain a transformation matrix $W^{s \to t}$ that projects embeddings in a source language S to a target language T. Effectively, this would align S to T, thus enabling cross-lingual inference via vector arithmetic between the two spaces. They accomplish this by creating a *seed dictionary* of $n = 5,000$ most frequent words in S and their corresponding translations in T. Using this dictionary, they employ Stochastic Gradient Descent (SGD) to learn $W^{s \to t}$ by minimizing the Mean Squared Error (MSE) between the transformed entries x_s in S and their translations z_t in T:

$$\text{MSE} = \sum_{i=1}^{n} \parallel W x_i^s - z_i^t \parallel^2$$

After finding the optimal $W^{s \to t}$ for a variety of language pairs, Mikolov et al. (2013b) experiment with basic word-level translation by using cosine distance to to find the closest vector $z \in T$ for a variety of candidates $x \in S$. Though they report encouraging results for English-Spanish (and vice versa), their experiments on English-Czech suggest that there is much room for improvement. This task is now referred to as bilingual dictionary indunction (BDI) and serves as the primary method of evaluation for bilingual embedding mappings.

Xing et al. (2015) note an important flaw in the aforementioned approach. Namely, they point out that the objective function for learning embeddings (maximum-likelihood based on the inner product of hidden layer and the output layer), the distance measurement for relating words in vector space (cosine distance), and the metric employed for learning the mapping matrix $W^{s \to t}$ (Euclidean distance for MSE) is, by and large, inconsistent. As such, they posit that the estimation employed for monolingual vectors, as well as the procedure for learning a bilingual mapping is inherently flawed. To correct the former in terms of the training objective, they propose to normalize each word vector to unit length, which they accomplish by dividing a vector by its $l2$ norm. This has the effect of constraining each vector to a hyperplane and an added benefit

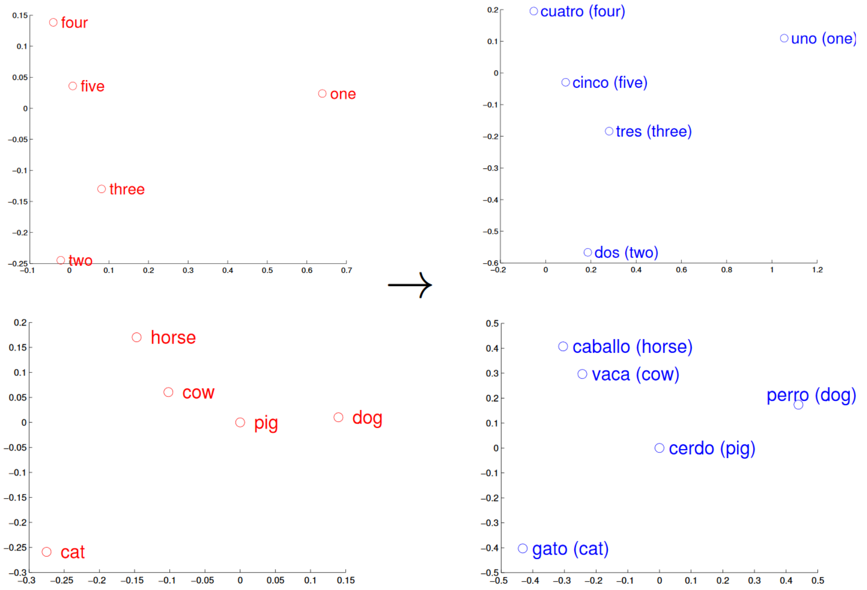


Figure 2.2: Word vector representations of numbers and animals as they occur in the respective English and Spanish vector spaces. The word vectors in each language were projected to two-dimensional space using Principal Component Analysis (PCA) and manually rotated to accentuate the distributional similarity. Graphic taken from Mikolov et al. (2013b).

of equating the inner product with the cosine distance. Furthermore, in order to address the inconsistency of the Euclidean distance used in learning $W^{s\to t}$ and the cosine distance used in retrieval, Xing et al. (2015) reframe the training objective as:

$$\max_{W} \sum_{i=1}^{n} (Wx_i^s)^\top z_i^t$$

In addition, they constrain $W^{s\to t}$ to be orthogonal ($W^\top W = I$), which preserves unit length after mapping.

Artetxe et al. (2016) attempt to generalize the work of Mikolov et al. (2013b) and Xing et al. (2015) by proposing a framework of linear transformations. First, they note that, given the orthogonality constraint, the problem of learning $W^{s\to t}$ can be efficiently solved by $W^{s\to t} = VU^\top$, where $T^\top S = U\Sigma V^\top$ is the SVD factorization of $T^\top S$. Furthermore, they point out that constraining $W^{s\to t}$ to be orthogonal effectively equates the MSE-based optimization of Mikolov et al. (2013b) and the cosine-based optimization of Xing et al. (2015), length normalization non-withstanding. In fact, their experiments reveal that the orthogonality constraint is more relevant to learning $W^{s\to t}$ than length normalization (for which preserving the latter serves as the primary motivation for orthogonality in Xing et al. (2015)), as it leads to better performance and preserves monolingual invariance of both S and T. In addition to this, they also propose dimension-wise mean centering as a means of pre-mapping normalization in order to capture the intuition that two randomly selected words are likely semantically unrelated and that the cosine difference between them is zero.

Artetxe et al. (2018b) later expand on this framework by surveying other linear transformations. Among these are *whitening*, which applies a sphering transformation to S and T, ensuring that the embedding dimension are uncorrelated among themselves, *re-weighting*, which re-weights each dimension according to its cross-correlation after mapping, and *dimensionality-reduction*, which retains n components after mapping and discards the rest. In total, they report the following insights:

- Whitening can help in dictionary induction, but only if the mapped embeddings are appropriately de-whitened.
- Re-weighting is highly beneficial, but should be performed in the target language in order for length normalization to be effective in nearest-neighbor retreival.
- Dimensionality reduction is an extreme form of re-weighting and should be foregone in place of it.

2.2.2 Mapping With Minimal or No Supervision

Though the aforementioned methods, among others, have helped to generalize and refine the mapping procedure introduced in Mikolov et al. (2013b), they have nonetheless relied on a typically high-quality seed lexicon that serves as a "guide" in learning $W^{s\to t}$. Vulic and Korhonen (2016) investigate the effect of the quality and size of such lexicons, corroborating that $2000 > n < 5000$ entries of largely monosemous translation-pairs tends to (generally) yield the best results on the dictionary induction task. This implies that, in order for any bilingual embedding mapping method to learn an accurate transformation, a well-curated lexicon of considerable size must exist for any language pair in question. In the low-resource language scenario, this quickly becomes an issue.

In an attempt to alleviate the need for large bilingual dictionaries, Artetxe et al. (2017) propose an iterative self-learning method that achieves competitive results with considerably smaller lexicons than prior approaches. Their approach is grounded in the intuition that an initial, weak mapping is enough to infer a cross-lingual signal, which can then be leveraged to add more entries to the initial lexicon and refine it. This "refined" lexicon is then passed as input to the same mapping procedure until the average dot-product for the induced lexicon falls below a given threshold from one iteration to the next. They find that an initial dictionary of 25 word-pairs is typically enough to learn a competitive mapping to other state-of-the-art approaches. In the absence of a dictionary, they suggest that the same procedure can be performed given a dictionary of shared numerals between the languages. The same intuition is corroborated by Smith et al. (2017), who report good results on a "pseudo-dictionary" composed of identical strings occurring between two languages.

Artetxe et al. (2018a) extend the approach of Artetxe et al. (2017) to exclude initial seed dictionaries entirely. Instead of relying on shared numerals or strings between languages, they posit that the distribution of similarities of any given word is likely to be similar for any two languages and can thus serve as an initial cross-lingual signal. This is based on the assumption that any two sets of embeddings are isometric, meaning that similarity matrices $M_S = SS^\top$ and $M_T = TT^\top$ are identical up to a permutation of their rows. Though Søgaard et al. (2018) point out that word embeddings for any two languages - especially very distantly related ones - are hardly isometric, the assumption in the case of Artetxe et al. (2018a) is still strong enough in order to generate a weak signal as a proxy of a lexicon. They accomplish this by sorting the values of each row in M_S and M_T in order to obtain sorted(M_S) and sorted(M_T), which enables efficient nearest-neighbor retrieval and alleviates the need to compute the permutation such that $M_S \approx M_T$. Doing so yields S' and T', which are passed as a seed to the iterative process of Artetxe et al. (2017). Artetxe et al. (2018a) find that this approach, along with some refinements to the iterative algorithm (such as introducing stochasticism to the dictionary induction process, a frequency-based vocabulary cut-off, among others) produces state-of-the-art dictionary induction results for the `EN-IT`, `EN-DE`, and `EN-ES` language pair and is competitive for `EN-FI`.

In contrast to the approach of Artetxe et al. (2018a), another line of research in unsupervised bilingual embedding mapping has relied on generative adversarial learning (Goodfellow et al., 2014). One such approach was attempted by Zhang et al. (2017), who learned $W^{s \to t}$ via a classic adversarial setup: a generator G transforms S such that it is indistinguishable from T and a binary classifier discriminator D decides whether S' is, in fact, T or not. They experiment with this architecture in a number of ways, including learning a unidirectional transformation, a bidirectional transformation, and including an auto-encoding objective that simultaneously attempts to re-create the original embedding $x \in S$ while learning $W^{s \to t}$. Conneau et al. (2017) employ a similar adversarial approach as Zhang et al. (2017). However, their method is more similar to Artetxe et al. (2017) in the sense that it removes the reconstruction procedure of Zhang et al. (2017), constrains $W^{s \to t}$ to be orthogonal, and implements a dictionary refinement procedure (albeit via Procrustes, which Artetxe et al. (2017) do not employ). However, as both Artetxe et al. (2018a) and Søgaard et al. (2018) demonstrate, both aforementioned methods fail to perform in more challenging settings than reported in the original experiments, such as learning mappings for less frequent words and between morphologically-dissimilar languages. By and large, Artetxe et al. (2018b) show that their method outperforms the latter approach by a significant margin, as well as that the former approach completely fails in less favorable testing scenarios.

2.2.3 Hubness and Retrieval Methods

Traditionally, the BLI task introduced in Mikolov et al. (2013b) has been performed via conventional nearest neighbor retrieval: given a query term $x \in S$ and a transformation matrix $W^{s \to t}$, map S to T and return $z \in T$, such that z is the closest entry to x' in terms of cosine distance. The problem with this approach, however, is that high-dimensional spaces are asymmetric, meaning that a particular z being a nearest neighbor of x' does not imply that x would be a nearest neighbor of z'. This gives rise to an issue called *hubness*, in which particular words in T are "universal" neighbors to a large number of mapped words in S (Radovanović et al., 2010).

In recent years, there have been numerous efforts to alleviate the hubness problem in relation to nearest neighbor retrieval. For example, Dinu et al. (2014) attempt to address the issue by reversing the query direction from $x \rightarrow z$ to $z \rightarrow x$ and introduce a ranking mechanism for words in S. Likewise, Smith et al. (2017) reformulate the typical retrieval method in terms of confidence as returned by the Softmax and propose instead an Inverted Softmax:

$$P_{z\rightarrow x} = \frac{e^{\beta S_{xz}}}{\alpha_z \sum_n e^{\beta S_{xn}}}$$

where n is the number of randomly sampled words in the vocabulary, α is a normalization vector, and β is a temperature parameter learned to maximize the log probability over the training dictionary:

$$\max_{\beta} = \sum_{x,z} \ln P(z \rightarrow x)$$

However, though Smith et al. (2017) report significant improvement in using the Inverted Softmax over standard cosine difference on the `EN-IT` dataset of Dinu et al. (2014), Artetxe et al. (2018b) do not observe a significant difference on the same dataset. Furthermore, Conneau et al. (2017) deem the temperature parameter of the Inverted Softmax as problematic, as it is tuned in an unsupervised setting. Instead, they propose a non-parametric measure called Cross-Domain Similarity Local Scaling (CSLS). Given a mapped source vector $x' \in S$ and a target vector $z \in T$, CSLS first computes $\mathrm{r}_T x'$ and $\mathrm{r}_S z$, which are the average cosine similarities of x' and z to their k nearest neighborhoods in T and S, respectively. CSLS is then calculated via the following:

$$\mathrm{CSLS}(x', z) = 2\cos(x', y) - \mathrm{r}_T x' - \mathrm{r}_S z$$

This has the effect of simultaneously increasing similarity associated with isolated word vectors as well as decreasing the similarity with vectors in hubs. Conneau et al. (2017) report drastic improvement in their BLI experiments when using CSLS over any other surveyed method, including Dinu et al. (2014) and Smith et al. (2017). Furthermore, CSLS is the retrieval employed in the unsupervised approach of Artetxe et al. (2018a) when inducing translation candidates.

2.3 From Bilingual to Multilingual Spaces

The aforementioned mapping approaches have all had one trait in common: they map monolingual embeddings into a shared *bilingual* space. However, as Ruder et al. (2017) point out, there is clear utility in extending this process to the multilingual scenario, where embeddings for more than two languages are all aligned to each other. For example, in the case of Finnish and Estonian - both agglunitative languages of the Finno-Ugric family - mapping both languages to the resource-rich English would enable inference between the two, which otherwise may not be possible without a good `FI-ET` lexicon (Duong et al., 2017; Ruder et al., 2017). Furthermore, multilingual embeddings enable multi-source learning and transfer, which has been shown to be beneficial in much previous work, albeit not with embeddings directly (Ruder et al., 2017; Guo et al., 2016; Agić et al., 2016).

In general, the most common approach for generating multilingual embeddings is, given monolingually trained embeddings S in n languages, to determine a pivot language T_p (typically English, given the wealth of resources between English and the rest of the world's languages) and learn an orthogonal transformation from each language to the pivot $W^{S_1 \to T_p}...W^{S_n \to T_p}$. However, though indeed a strong baseline, the quality across non-pivot alignments is nonetheless directly dependent on the quality of each language's alignment to the pivot space. In other words, the inter-linguistic features that may be shared by any two non-pivot languages can only be exploited if the same features are aligned equivalently to features in the pivot space. In many situations, especially for morphologically unrelated languages (e.g. Basque and Russian), an alignment to a pivot language (e.g. English) would likely fail to approximate the alignment between the two languages that a direct bilingual mapping would otherwise provide.

Duong et al. (2017) attempt to refine this procedure by learning mappings from various already aligned subspaces into to a single space. Concretely, for a set of embeddings S in n languages aligned to a single set of shared target language embeddings T, a pivot subspace $C_p = \{(S_i, T)\}$ is chosen. The goal of the algorithm is then to learn a mapping $W^{T_i \to T_p}$ from the T subspace in C_i to the T subspace in C_p. This is trival, since T is the same language in every case and a dictionary of equivalent string-pairs can easily be produced for the mapping. The learned transformations $W^{T_i \to T_p}$ are then applied to the S_i subspaces in C_i in order to unify the alignment. Though Duong et al. (2017) report improved results with this method, it is nonetheless reliant on a strong initial mapping from all languages to T and thus does not exploit any inter-language similarities explicity.

Though many more methods for generating multilingual word spaces exist, many require simultaneous training and/or parallel corpora, so we will not cover them here.

2.4 Cross-lingual Learning

The de-facto method for evaluating any of the aforementioned mapping approaches has been bilingual dictionary induction. Though it is arguable whether BDI belongs to the instrinsic or extrinsic class of evaluation methods, it is nonetheless evident that most literature in multilingual embeddings treats it as a measure of inherent quality of the resultant aligned spaces. Shifting the focus from BDI to other, more downstream tasks, can shed insight not only about the inherent quality of the embeddings but also how well this translates to practical scenarios.

Furthermore, doing so can pose many implications for cross-lingual learning, which is concerned with developing machine learning models that are not language agnostic (i.e. trainable on all languages independently) but rather *multilingual* - taking advantage of cross-lingual transfer between languages instead of discounting it. The motivations for cross-lingual learning are twofold. Primarily, given that manually-annotated linguistic resources exist for only a fraction of the world's approximately 6,900 languages, cross-lingual learning makes possible to leverage these to improve models for the remainder of languages for which resources are non-existent or scarce. Furthermore, exploiting the cross-lingual signal between languages can produce models that are less prone to overfitting, more robust to noise, and, in many cases, perform better as a whole - resources nonwithstanding (Ruder et al., 2017). For these reasons, multilingual embeddings are a natural gateway to cross-lingual learning, ensuring that the token-level signal for a number of languages is aligned from the start.

2.4.1 Universal Dependencies

The Universal Dependencies (UD) project (Nivre et al., 2016) has long been a popular testing ground for cross-lingual learning methods. The chief goal of the UD project is to develop a universal syntactic annotation schema that is capable of capturing similarities across a vast number of languages, as well as the idiosyncracies that set them apart from one another. This approach is governed by two guiding principles: *lexicalism* and *dependency*. The former is based on the notion that words are the basic units of grammatical annotation, being composed of an inherent morphological structure and then entering into syntactic relationships. *Dependency* is a complement to this idea in the sense that that grammatical annotation should aim to represent the syntactic structure of a sentence at the word/token level and relate words to each other via a set of directed binary relation arcs (Jurafsky and Martin, 2014). With this in mind, UD provides a universal set of morphological features, part-of-speech tags, and syntactic relation labels that can be applied cross-lingustically at the sub-word [1], token, and token-to-token level, respectively. In addition to universal features, UD annotation also captures language-specific features, POS-tags, and relations that exist outside of the universal set (e.g. the prospective verb aspect that exists in Basque and a small number of other languages). Altogether, UD treebanks provide token-level annotation for collections of sentences per individual language, in compliance with the CoNLL-U format:

```
1    They    they    PRON    PRP    Case=Nom|Number=Plur                   2    nsubj    2:nsubj|4:nsubj
2    buy     buy     VERB    VBP    Number=Plur|Person=3|Tense=Pres        0    root     0:root
3    and     and     CONJ    CC     _                                      4    cc       4:cc
4    sell    sell    VERB    VBP    Number=Plur|Person=3|Tense=Pres        2    conj     0:root|2:conj
```

[1] Universal Dependencies treebanks do not label individual morphemes, but rather the combination of morphological features that adequately describe the inherent structure of words

```
5    books    book    NOUN    NNS    Number=Plur                        2    obj      2:obj|4:obj
6    .        .       PUNCT   .      _                                  2    punct    2:punct
```

2.4.2 Monolingual Part-of-Speech Tagging

The existence of UD has re-ignited an interest in two vital NLP tasks: Part-of-Speech (POS) tagging and dependency parsing. The former is a simple sequence prediction task in which a machine learning model must predict an optimal sequence of part-of-speech tags $\hat{Y}$ that corresponds to an input token sequence X. For example, for a sample input $X =$ (the, cat, meowed), the best corresponding sequence of POS-tags would likely be $\hat{Y} = (\texttt{DET}, \texttt{NOUN}, \texttt{VERB})$. Historically, this has been accomplished via Hidden Markov Models (HMMs), where the probability of observing a sequence of POS-tags $Y = y_1, y_2, ..., y_L$ of length L is given by $P(Y) = \sum_X P(Y|X)P(Y)$, using transition probabilities $P(Y) = \prod_{i=1}^{L+1} P(y_i|y_{i-1})$ and emission probabilities $P(X|Y) = \prod_{i=1}^{L+1} P(x_i|y_i)$. The optimal solution $\hat{Y}$ can then be computed via the Viterbi algorithm.

In recent times, however, traditional statistical approaches such as HMMs have been abandoned in favor of Recurrent Neural Network (RNN) based deep learning models. An RNN (Elman, 1990) is a function that reads an n-sized sequence of fixed-length vectors and produces a output vector h_t for each timestep (or token) t in the sentence. Ultimately, the final hidden state h_n can be interpreted as a composed representation of the sentence as a whole. h_n can then be passed as input to a softmax classifier for the task at hand or another RNN layer for learning higher-order representations. However, though attractive in theory, basic RNN models suffer from the vanishing gradient problem, wherein features learned earlier in a sequence are effectively forgotten and elements occurring towards the tail end are prioritized. Long Short-term Memory Networks (LSTMs) (Hochreiter and Schmidhuber, 1997) are a type of RNN that aim to mitigate this issue by introducing gating mechanisms that control what information is retained and forgotten between timesteps.

In the context of POS-tagging, a bidirectional LSTM (BiLSTM) is often preferred in favor of a basic LSTM. Unlike basic LSTMs, BiLSTMs complete both a forward and backward pass over the input sequence. The output vector h_n is then computed by concatenating the respective forward and backward passes: $h_n = \text{LSTM}_f(x_{1:n}) \oplus \text{LSTM}_b(x_{n:1})$. For syntactic tasks such as POS-tagging, this has the effect of further mitigating the vanishing gradient for especially long sentences and accounting for right-branching nodes in a syntactic tree. A neural POS-tagging pipeline typically passes a sequence of vectors $D \in \mathbb{R}^{n \times d}$ through a BiLSTM, where n is the length of the sequence and d is the dimensionality of the word vectors corresponding to the sequence's tokens. At each timestep t, the output of the LSTM h_t is passed to a softmax output layer, which produces a probability distribution over the tag vocabulary M. This allows for the optimal tag to be selected via $\hat{y}_t = \arg\max_{y \in M} P(y|w_t)$.

Wang et al. (2015) employ this exact structure and experiment with a variety of pre-trained word embeddings and hidden layer sizes. They report state-of-the-art performance across all of their experiments, demonstrating the effectiveness of this approach over previous statistical methods. Plank et al. (2016) expand on this work in numerous dimensions. Primarily, they evaluate their tagger on 22 languages in addition to English. Furthermore, they incorporate an additional input representation in the form of *character embeddings*. Character embeddings are learned by passing a BiLSTM over the characters that comprise a word, yielding the character representation $\vec{c}$. In the case of Plank et al. (2016), character embeddings are concatenated with pretrained word vectors in order to yield a shared token + sub-token representation $\vec{w} \oplus \vec{c}$. Lastly, they incorporate an auxiliary loss for predicting the log-frequency of a token as it is observed in the training data $a = int(log(\text{freq}_{\text{train}}(w)))$, yielding a joint-training objective $L = L(y', y) + L(a', a)$. They find that a BiLSTM model with character and word-level representations in addition to the auxiliary loss produces state-of-the-art performance on almost all surveyed languages, with the sub-word features proving especially beneficial for non-Indo-European and Slavic languages.

2.4.3 Cross-lingual Part-of-Speech Tagging

Though Plank et al. (2016) claim their model to be multilingual, they train on each surveyed language individually and thus do not take advantage of any cross-lingual signal that may exist between languages. There have, however, been numerous efforts to extend POS-tagging multilingually. A common pre-deep learning approach pioneered by (Yarowsky et al., 2001) took advantage of automatically aligned parallel corpora in order to project POS tags from an annotated language onto another, unannotated language. They accomplish this via a combination of techniques, including downsampling statistically "noisy" alignments and training the emission and transition HMM parameters separately. Ultimately, they find that training on 500K automatically aligned English-French sentences can yield a comparable POS-tagging accuracy as would training on 100K "gold" aligned sentences.

Zhang et al. (2016) employ a feature-based HMM enhanced with coarsely mapped word embeddings towards the task of POS-tagging an unannotated target language based on supervision from a well-resourced source language. Their method consists of initially training the feature-based HMM on a source language for which resources can be assumed to be plentiful. Following the training, they learn a coarse orthogonal mapping from the low-resource target language to the source language using a dictionary of 10 translation-pairs. The mapped target embeddings are used to initialize the emission parameters of the target model while the transition parameters are transferred from the previously-trained source HMM. In training the target model, they employ a series of refinement techniques in order to improve the quality of the initial coarse mapping and utilize the Expectation-Maximization algorithm to regularize the learned emission parameters to remain as close as possible to the source model. They find that this latter step significantly improves performance over simply transferring the source HMM to the target language.

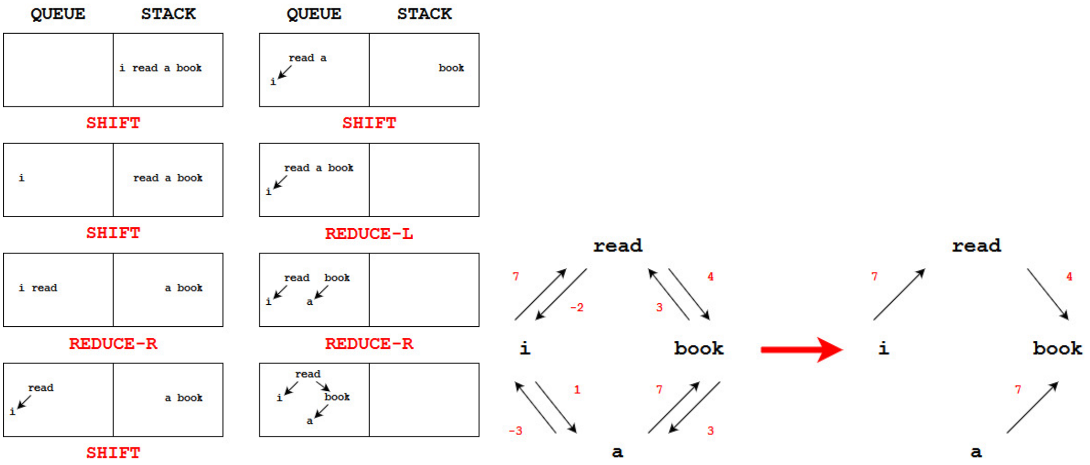
Like Zhang et al. (2016), Fang and Cohn (2017) also focus on POS-tagging in low resource scenarios. However, they situate their experiments in the context of deep learning and, in particular, multi-task learning. In their experiments, a POS-tagger is trained on a well-resourced target language (e.g. English). This model is then employed for tagging a held-out portion of unannotated target language data as a form of distant supervision. The "noisy" tagged data is then passed to a BiLSTM layer that is shared with input from gold-standard target language data. Ultimately, each input stream diverges into its respective task-specific layers, with the gold-standard data passing through a dense layer before being output via a task-specific softmax. In all cases, they map pre-trained monolingual embeddings for the source and target languages to a shared space via the CCA and Clustering methods described in Ammar et al. (2016). They report encouraging findings, showing that the multi-task approach with distant supervision fares better than any of the surveyed methods. However, the individual contribution of either embedding mapping strategy varies greatly per language.

2.4.4 Monolingual Dependency Parsing

Although POS-tagging can shed some light on the syntactic characteristics of a sentence, it nonetheless fails to capture the inherent tree-like structure of which a sentence is (typically) composed. Dependency parsing is a higher-order solution to this problem, framing the notion of grammatical relations as a directed graph $G = (V, A)$. Here, V are vertices corresponding to the set of words in a sentence, and A are a set of ordered pairs of vertices, or arcs, linking **head** words to their **dependents** (Jurafsky and Martin, 2014). In the Universal Dependencies annotation schema, every arc is also provided with a label, describing the syntactic relationship between a head word and its dependent (e.g. direct object, adverbial modifier, etc.). The task of parsing Universal Dependencies treebanks is thus to produce an optimal $\hat{A}$ for a provided sentence S. This is typically represented via two metrics: **Unlabeled Attachment Score** (UAS), which measures the correct assignment of heads and their attachment to the correct dependents; and **Labeled Attachment Score** (LAS), which measures the same, including the correct labeling of the arc.

Dependency parsers can be generally attributed as belonging to one of two classes of algorithms: transition-based parsing or graph-based parsing. Transition-based parsing is performed by reading tokens in a sentence from left to right, where a "buffer" manages yet-unparsed words and a "stack" collects words whose head has not been seen or whose dependents have not all been fully parsed. The shift-reduce parser is arguably the simplest variant of transition-based parsers, comprising only of three operations: SHIFT, REDUCE-R, and REDUCE-L. For every token in the sentence, the shift-reduce parser must decide either to SHIFT the token from the buffer to the stack, REDUCE-Rightward, making the top word of the stack as the head of the second word, or REDUCE-Leftward, making the second word of the stack the head of the top word. Given an annotated corpus of parses, a classifier can be trained to choose an action at each timestep. Figure 2.3(a) depicts a step-by-step shift-reduce parse of a sample sentence.

However, though deep-learning-oriented transition-based parsers have been applied suc-



(a) Shift-reduce parse of a sample sentence.

(b) Graph-based parse of a sample sentence

Figure 2.3: Approaches for parsing a sample sentence *"I read a book."*

cessfully to the task of dependency-parsing (Dyer et al., 2016), they have recently fallen out of favor due to the relative simplicity and comparable performance of graph-based parsers. Such approaches employ machine learning in order to assign weights to every edge $A \in G$. A maximum spanning tree (MST) is then constructed by retrieving the tree with the highest combination of edges. Figure 2.3(b) depicts a graph-based parse of a sample sentence[2].

Kiperwasser and Goldberg (2016) present a neural graph-based parser that takes sequences of word embeddings as input and passes them through BiLSTMs to learn context-aware feature representations of head and dependent tokens as they relate to the parsing task. Their approach is inspired by arc-factored parsing, which decomposes the score of a tree to the sum of the score of its head-dependent arcs (h,d):

$$\text{parse(s)} = \arg\max_{y \in Y(s)} \sum_{(h,m) \in y} \text{MLP}(\phi(s,h,d))$$

Here, the feature extracting function $\phi(s,h,d)$ is simply the concatenation of the BiLSTM encoding of the head and dependent word, respectively:

$$\phi(s,h,d) = \text{BiLSTM}(x_{1:n},h) \oplus \text{BiLSTM}(x_{1:n},d)$$

Furthermore, the scoring function is a tanh-activated multilayer perceptron (MLP) of dimensionality L, where L is the number of possible (h,m) combinations. As such, each dimension $l \in L$ corresponds to an individual (h,d) pair. While this only accounts for scoring parses, the features $\phi(s,h,d)$ are passed to a separate MLP that is tasked with predicting relation types.

[2] Both transition and graph-based figures are inspired by Graham Neubig's dependency parsing tutorial, which can be found at: `http://www.phontron.com/slides/nlp-programming-en-11-depend.pdf`.

Dozat and Manning propose an important extension to this approach. As in Kiperwasser and Goldberg (2016), their parser accepts sequences of pretrained word embeddings added element-wise to randomly initialized "holistic" embeddings, wherein each word vector w_i is encoded via a BiLSTM, resulting a hidden vector h_i. The distinction in their approach is that, instead of relying on the BiLSTM to serve as the standalone featurizer, each hidden vector h_i is further passed to four separate MLPs, each of which serve as separate featurizers that seek to characterize a word via four different representations:

- a head seeking its dependents: $h^{arc-head} = \mathrm{MLP}^{(arc-head)}(h_i)$
- a dependent seeking its head: $h^{arc-dep} = \mathrm{MLP}^{(arc-dep)}(h_i)$
- a dependent deciding on its label: $h^{rel-dep} = \mathrm{MLP}^{(rel-dep)}(h_i)$
- a head deciding on the labels of its dependents: $h^{rel-head} = \mathrm{MLP}^{(rel-head)}(h_i)$

Furthermore, instead of relying on an MLP for arc/relation prediction, Dozat and Manning (2016) consider a bilinear transformation (i.e. a biaffine classifier). Unlike a linear transformation $Wx + b$, a bilinear transformation seeks to transform two matrices via a single weight and bias: $x_1 W x_2 + b$. Thus, in order to predict an arc for a token i in a sentence, a bilinear transformation between the matrix of all head vectors $H^{(arc-head)}$ and the current dependent vector $h^{(arc-dep)}$ is applied:

$$\begin{aligned} s_i^{(arc)} = H^{(arc-head)} W^{(arc)} h_i^{(arc-dep)} \\ + H^{(arc-head)} b^{T(arc)} \\ y_i'^{(arc)} = \arg\max_j s_{ij}^{arc} \end{aligned}$$

In effect, the bilinear transformation results in an $n \times n$ matrix, where n is the number of words in a sentence. In this case, the rows are likened to head words, columns to their dependents, and the matrix entries to probabilities of a token being the head of another token.

After predicting the arc $y_i^{(\hat{arc})}$, another bilinear transformation is applied in order to predict its relation label:

$$\begin{aligned} s_i^{(rel)} = h_{y_i'^{(arc)}}^{T(rel-head)} U^{(rel)} h_i^{(rel-dep)} \\ + W^{(rel)} (h_i^{(rel-dep)} \oplus h_{y_i'^{(arc)}}^{(rel-head)}) + b^{(rel)} \end{aligned}$$

Here, the first term corresponds to the probability of observing a label given the information encoded in both $h^{(rel-head)}$ and $h^{(rel-dep)}$ vectors (e.g. the probability of the label *det* given word i is *the* with the head *cat*). The second term relates to the probability of observing a label given

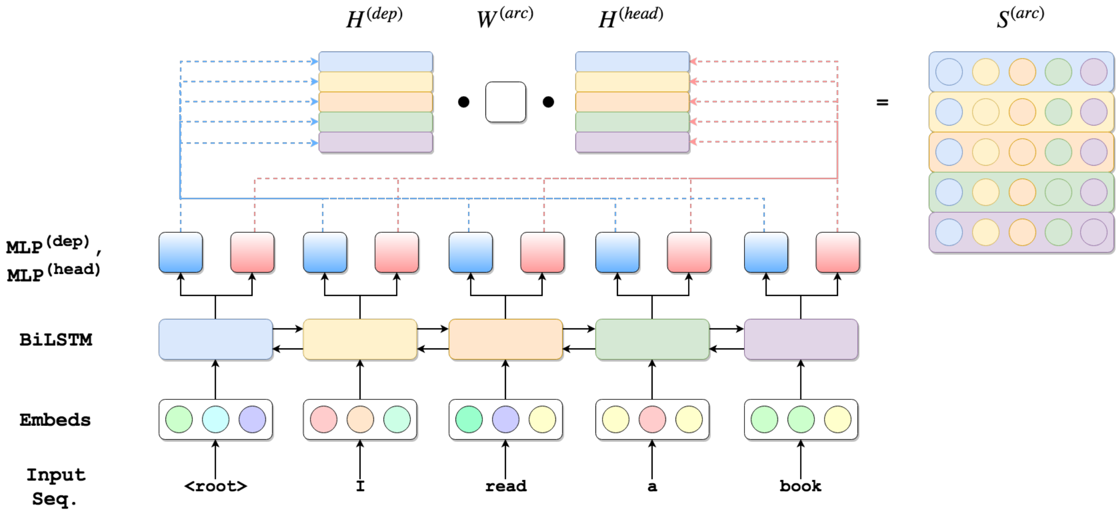


Figure 2.4: Dozat and Manning (2016)'s parser applied a sample sentence.

either $h^{(rel-head)}$ or $h^{(rel-dep)}$ vector (e.g. the probability of the label *det* given word i is *the* or word j is *cat*). The last term simply relates to the probability of a label (Dozat and Manning, 2016). As before, the relation for the arc $y_i^{(\hat{arc})}$ is predicted via:

$$y_i^{(\hat{arc})} = \arg\max_j s_{ij}^{arc}$$

This approach is later refined by Dozat et al. (2017), who include character and POS-tag embeddings as additional input representations in order to account for languages with rich morphology. Ultimately, they report new state-of-the-art results on the majority of languages represented in Universal Dependencies.

2.4.5 Cross-lingual Dependency Parsing

As with POS-tagging, there has been considerable research in the domain of extending dependency parsers cross-linguistically. Most of the early approaches in this regard have been concerned with de-lexicalizing the feature representations learned for a source language in order to transfer them to an unseen target language. This is largely motivated by the fact that POS-tags and dependency relation labels are, in general, universal across languages, whereas word-level lexical features are not. One of the first successful approaches in transferring a de-lexicalized parser to a target language was reported in McDonald et al. (2011), who experimented with a variety of training and evaluation languages. The features they included were the POS-tags of words on the buffer and stack, the word identities of words on the buffer and stack (i.e. the numbered location of a word as it appears in a sentence), and the word identity of the syntactic head of the top word on the stack. In addition to a direct transfer approach, wherein a model trained on one language is evaluated directly on another, McDonald et al. (2011) also considered a projected approach, where the training language parser is employed for parsing an initial set of evaluation language sentences as a form of distant supervision (similar in spirit to Fang and Cohn (2017)).

However, though McDonald et al. (2011) showed that de-lexicalization is a viable approach for cross-lingual dependency parsing, it is nonetheless apparent that the success of monolingual parsers is largely owed to their advantage in accounting for lexical features. Recent advancements in multilingual embedding alignment have inspired a trend towards lexicalization in parsing, as the distributional information carried by word embeddings for different languages can be encoded into vectors that reside in a single space. One of the first embedding-based lexicalized parsing approaches was carried out by Guo et al. (2015), whose method includes a transition-based neural dependency parser that is trained on a de-lexicalized English features. These include word, POS-tag, and dependency relation features that are projected to an embedding layer which the network estimates throughout training. In addition to this, they include lexical features in the form of monolingual embeddings projected to multilingual space via an extension of Faruqui and Dyer (2014)'s CCA alignment method. In their experiments, they find that lexicalizing the parser via multilingual embeddings improves the de-lexicalized parser by an average error rage of 10.9% when evaluating on an unseen language.

Similarly, Ammar et al. (2016) propose a transition-based parsing architecture that can be trained on a variable number of languages simultaneously while taking a variety of feature representations into account. Though they enable cross-lingual learning by experimenting with a variety of multilingual embedding algorithms, they also include numerous other input representations for their parser. These include multilingual Brown clusters, which are projected from Brown clusters of English words to other languages via word alignment, word type embeddings, fine-grained POS-tag embeddings, and "language embeddings" which are learned to predict the identity of the training language. In addition to this, they propose an additional architecture that incorporates a BiLSTM-based POS-tagger in order to provide POS-tags in situations where gold POS-tags are not available. Ultimately, they find that the combination of lexical embeddings (Brown clusters and word types), language-id embeddings, and fine-grained POS embeddings yields impressive performance that, in most cases, outperforms monolingual parsing when trained multilingually with a variety of languages serving as input.

CHAPTER 3

IMPROVING MULTILINGUAL WORD EMBEDDINGS

In this Chapter, we introduce three algorithms for generating multilingual word embeddings. The first is an iterative algorithm that employs ground-truth dictionaries, which we call the **Supervised Iterative Multilingual System** or ITERSUP. The second is an unsupervised algorithm that induces dictionaries at the first step and then follows the procedure of ITERSUP. We call this algorithm the **Fixed Unsupervised Multilingual System** or UNSUPFIXED. The last algorithm induces dictionaries at the first step and continues to induce refined dictionaries after each iteration. We call this the **Generative Unsupervised Multilingual System** or UNSUPGEN. The principal difference between the latter two systems is that, while UNSUPFIXED induces a fixed set of dictionaries between all languages that is employed throughout the iterative process, UNSUPGEN induces an initial set of dictionaries and continues to induce "refined" dictionaries after every iteration.

In addition to these systems, we consider a bilingual upper bound, which we call the **Pairwise Bilingual System** or PAIRWISE. We refer to this as an upper bound due to the fact that it employs ground truth dictionaries in all language directions, reflecting the most favorable conditions for mapping and making it completely supervised. In addition to this, we employ a supervised multilingual baseline, which we call **Baseline Multilingual System** or BASELINE.

3.1 Supervised Iterative Multilingual System

For this approach, we aim to leverage existing ground-truth dictionaries between a variable amount of languages in order to map embeddings for each language into a single, multilingual vector space. Algorithm 1 describes this process in detail. We begin by considering a set of languages $S = \{l_1, l_2, \ldots, l_n\}$. The first step of the algorithm is to designate one language $p \in S$ as a pivot (line 2). Our working set then becomes $M = S \setminus p$ (3). Using a bilingual dictionary $l_i : p$, we learn an orthogonal transformation $W^{l_i \to p}$ and map pretrained embeddings E_{l_i} to the space of the pivot language embeddings: $E'_{l_i} = \text{map}(E_{l_i}, E_p, l_i : p)$ for $i, \ldots, |M|$ (4:5). We then declare $E'_p = E_p$ (6).

This begins an iterative process wherein a new pivot language q is chosen and original p is returned to the working set M, thereby making a new working set $O = S \setminus q$ (8:9). We then concatenate row-wise the embeddings $A = \bigoplus_{j \in O} E'_{l_j}$ and dictionaries $d = \bigoplus_{j \in O} l_j : q$ (10:14). This ensures that mappings from the previous iteration are preserved in accordance to the previous pivot language and are not overridden in respect to the new pivot. Following this, we use d to learn $W^{A \to E_q}$ and map the concatenated embeddings to the pivot space: $A' = \text{map}(A, E_q, d)$ (15). A' is then decomposed into its constituent individual language E_{l_j} for $j \in O$ (16:17). The process continues switching pivots until there are no more unmapped pivots in M. Every language in M is then mapped to the original pivot p in the same fashion and the iteration stops (20:25).

Algorithm 1 Supervised Iterative Multilingual System

```
 1: function IterSup($S, E, d$)
 2:     $p \leftarrow l \in S$                                  ▷ Choose pivot language
 3:     $M \leftarrow S \setminus p$
 4:     for $l \in M$ do
 5:         $E'_l \leftarrow \text{map}(E_l, E_p, l : p)$            ▷ Map every language to pivot
 6:     $E'_p \leftarrow E_p$
 7:     for $l \in M$ do                                    ▷ Iterate over remaining languages
 8:         $q \leftarrow l \in M$                               ▷ Choose new pivot
 9:         $O \leftarrow p \cup M \setminus q$
10:         $A \leftarrow \emptyset$
11:         $d \leftarrow \emptyset$
12:         for $l \in O$ do
13:             $A \leftarrow \text{concat}(A, E_l)$           ▷ Concatenate embedding and dictionaries
14:             $d \leftarrow \text{concat}(d, l : q)$
15:         $A' \leftarrow \text{map}(A, E_p, d)$          ▷ Map conc. embeddings to pivot using conc. dicts
16:         for $l \in O$ do
17:             $E'_l \leftarrow \text{deconcat}(A', E_l)$      ▷ Deconcatenate embeddings for each language
18:     $A \leftarrow \emptyset$
19:     $d \leftarrow \emptyset$                                      ▷ End iteration
20:     for $l \in M$ do                                    ▷ Reconcatenate and map to original pivot
21:         $A \leftarrow \text{concat}(A, E_l)$
22:         $d \leftarrow \text{concat}(d, E_l)$
23:     $A' \leftarrow \text{map}(A, E_p, d)$
24:     for $l \in M$ do
25:         $E'_l \leftarrow \text{deconcat}(A', E_l)$
    return $E'$
```

3.2 Fixed Unsupervised Multilingual System

An important hindrance of ITERSUP is that it assumes the availability of ground-truth dictionaries for all directions in the language set S. In the vast majority of cases, access to such dictionaries will not be possible. As such, we employ the system of Artetxe et al. (2018a) in order to induce dictionaries in an unsupervised manner. These are then fed to ITERSUP to serve as an alternative to the ground-truth dictionaries employed therein. We deem this algorithm as *fixed* due to the fact that the dictionaries induced at the initial step are reused throughout the iterative process.

Algorithm 2 describes the UNSUPFIXED algorithm in detail. First, S is assumed to be the same as in ITERSUP. For every language pair $(l_i, l_j) \in S$, we map $E'_{l_i} = E_{l_i} W^{l_i \to l_j}$ (lines 3:4). We then employ E'_{l_i} in order to induce the dictionary $l_i : l_j = \text{induce}(E'_{l_i}, E_{l_j})$ via CSLS retrieval [1] (Conneau et al., 2017) (5:6). The dictionaries yielded during this process are then fed to the ITERSUP algorithm as a proxy for ground-truth dictionaries (7).

Algorithm 2 Fixed Unsupervised Multilingual System

```
1: function INDUCEUNSUP($S$, $E$)
2:     $d = \emptyset$
3:     for $(l_i, l_j) \in S$ do
4:         $E'_{l_i} \leftarrow \text{map\_unsupervised}(E_{l_i}, E_{l_j})$
5:         $l_i : l_j \leftarrow \text{induce}(E'_{l_i}, E_{l_j})$
6:         $d \leftarrow \text{append}(d,\ l_i : l_j)$
7:     $E' \leftarrow$ ITERSUP($S$, $E$, $d$)
   return $E'$
```

3.3 Generative Unsupervised Multilingual System

Though UNSUPFIXED alleviates the need for ground-truth dictionaries, it does not leverage the possible information gain yielded by the resultant mappings. In other words, after learning a mapping $E'_{l_i} = E_{l_i} W^{l_i \to l_j}$, it is possible that the alignment between E'_{l_i} and E_{l_j} could induce a better dictionary $l_i : l'_j$ than the initial, fixed $l_i : l_j$. As such, we introduce an induction step after every map() operation in ITERSUP in order to iteratively produce “refined” dictionaries d. Algorithm 3 expresses this procedure in detail. It is important to note here that lines 2:6 refer to the initial dictionary induction step, which is identical to Algorithm 2. Lines 12:15 and lines 27:30 thus refer to the "refined" dictionary induction steps following the initial mapping and subsequent iterative mappings, respectively.

[1] see Chapter 2.2.2

Algorithm 3 Unsupervised Generative Multilingual System

```
function UnsupGen($S$, $E$)
    $d = \emptyset$
    for $(l_i, l_j) \in S$ do
        $E'_{l_i} \leftarrow$ map_unsupervised($E_{l_i}$, $E_{l_j}$)
        $l_i : l_j \leftarrow$ induce($E'_{l_i}$, $E_{l_j}$)
        $d \leftarrow$ append($d$, $l_i : l_j$)
    $p \leftarrow l \in S$
    $M \leftarrow S \setminus p$
    for $l \in M$ do
        $E'_l \leftarrow$ map($E_l$, $E_p$, $l : p$)
    $E'_p \leftarrow E_p$
    $d = \emptyset$
    for $(l_i, l_j) \in S$ do
        $l_i : l_j \leftarrow$ induce($E'_{l_i}$, $E_{l_j}$)
        $d \leftarrow$ append($d$, $l_i : l_j$)
    for $l \in M$ do
        $q \leftarrow l \in M$
        $O \leftarrow p \cup M \setminus q$
        $A \leftarrow \emptyset$
        $d \leftarrow \emptyset$
        for $l \in O$ do
            $A \leftarrow$ concat($A$, $E_l$)
            $d \leftarrow$ concat($d$, $l : q$)
        $A' \leftarrow$ map($A$, $E_p$, $d$)
        for $l \in O$ do
            $E'_l \leftarrow$ deconcat($A'$, $E_l$)
        $d = \emptyset$
        for $(l_i, l_j) \in S$ do
            $l_i : l_j \leftarrow$ induce($E'_{l_i}$, $E_{l_j}$)
            $d \leftarrow$ append($d$, $l_i : l_j$)
    $A \leftarrow \emptyset$
    $d \leftarrow \emptyset$
    for $l \in M$ do
        $A \leftarrow$ concat($A$, $E_l$)
        $d \leftarrow$ concat($d$, $E_l$)
    $A' \leftarrow$ map($A$, $E_p$, $d$)
    for $l \in M$ do
        $E'_l \leftarrow$ deconcat($A'$, $E_l$)
return $E'$
```

3.4 Baseline and Upper Bound

As a point of comparison, we experiment with a baseline multilingual system (Baseline) that is often employed in generating multilingual embeddings. This is analogous to the first step in IterSup, which designates a single language as a pivot $p \in S$ and maps embeddings for all remaining languages $l \in S \setminus p$ to the pivot embedding space E_p. Embeddings for all languages $l \in S$ can then be said to be aligned to each other as a byproduct of this operation. Algorithm 4 describes Baseline in detail.

Algorithm 4 Baseline Multilingual System

```
function Baseline($S$, $E$, $d$)
    $p \leftarrow l \in S$                                  ▷ Choose pivot language
    $M \leftarrow S \setminus p$
    for $l \in M$ do
        $E'_l \leftarrow \text{map}(E_l, E_p, l : p)$       ▷ Map every language to pivot
    return $E'$
```

In addition to Baseline, we also consider a **Pairwise Bilingual System** (or Pairwise), which serves as an upper bound. The motivation in doing so is that the optimal alignment for any language pair $(l_i, l_j) \in S$ can theoretically be achieved by leveraging an existing ground-truth dictionary $d_i : d_j$ in learning the mapping $W^{l_i \rightarrow l_j}$. As such, we do so for every language language pair $(l_i, l_j) \in S$ in order to evaluate how all of the proposed approaches compare to a completely supervised upper-bound.

CHAPTER 4

MULTILINGUAL EMBEDDING MAPPING EXPERIMENTS

This chapter is concerned with delivering the results of our experiments in generating multilingual embedding spaces. First, we outline the data and resources that we employ throughout our experimentation. Then, we provide a brief description of each experiment that we conduct. Following this, we post the results of each experiment and supplement them with a detailed discussion.

4.1 Data

We employ FASTTEXT embeddings provided by Facebook (Bojanowski et al., 2016) for our multilingual embedding mapping experiments. These embeddings were pre-trained on the Wikipedia dumps for 294 languages with the following hyperparameters:

- **Dimension:** $d = 300$
- **Negative Samples:** $k = 5$
- **Context Window:** c uniformly sampled between 1 and 5
- **Character N-gram Range:** $3 \geq n \leq 6$

It is important to note here that, though Wikipedia is a comparable corpus across all languages, its size varies dramatically between them. Thus, since FASTTEXT is based on SKIP-GRAM, which, in turn, learns better word representations with more training data, the quality of FASTTEXT embeddings is largely dependent on the size of a language's Wikipedia.

In addition to this, we make use of the dictionaries provided by Facebook's project MUSE (Conneau et al., 2017) in order to generate mappings for the ITERSUP, BASELINE, and PAIRWISE systems. Though MUSE provides dictionaries from 41 different languages to English and vice versa, we largely base our experiments on a different set, in which dictionaries in every direction are provided for six different languages: English, Spanish, German, French, Italian, and Portuguese. These are composed of unique, frequent terms in a source language's vocabulary as well as their corresponding translations in the target language (as determined by an Facebook-internal translation tool). Furthermore, each dictionary is split into a 5,000-entry TRAIN set and a 1,500-entry TEST set. Though we only employ the former for training the ITERSUP, BASELINE, and PAIRWISE systems, we make use of the latter for evaluating proposed systems. It is also important to note that there is notable overlap between words in a $source : target$ training dictionary and a $target : source$ test dictionary. As such, we only employ dictionaries in one direction and use their reverse for the complement, which effectively eliminates any dictionary overlap that may otherwise exist.

Our experiments make use of the vecmap software [1], which allows for fast and efficient linear mappings with GPUs.

4.2 Results

We conduct three main experiments in which we employ ITERSUP, UNSUPFIXED, and UNSUPGEN towards the goal of generating a multilingual embedding space for the language set $S = \{en, es, de, fr, it, pt\}$. Prior to mapping, we truncate the embeddings for each language to the top-200,000 most frequent items in its respective vocabulary. Following this, we normalize each set of embeddings to be unit length and mean-centered and then re-normalize for unit length. For all algorithms outside of PAIRWISE, we choose English as the initial pivot language p and follow the order (es, de, fr, it, pt) for the subsequent mappings.

Table 4.1 shows the complete results for all of the evaluated algorithms. Ultimately, the completely supervised ITERSUP produces the best BDI results. However, UNSUPGEN proves to be very competitive with ITERSUP, falling just 0.17 percentage points short in terms of accuracy. Furthermore, a glance at the individual dictionary performance reveals that ITERSUP produces the best accuracy in only half (15 out of 30) of the cases, where UNSUPGEN fares better for 13 of the remaining dictionaries, with the exception of $it : es$ and $it : de$. In all cases, all of the proposed algorithms perform better than BASELINE, while ITERSUP and UNSUPGEN also outperform PAIRWISE. We consider this a very positive result, as PAIRWISE is a completely supervised system and thus represents the theoretical optimal mapping between two languages.

In order to shed light onto the behavior of each algorithm, Figure 4.1 presents of all algorithms for each iteration. It is important to note here that the first step of ITERSUP involves mapping each language in S to English, which also serves as the entire BASELINE metric. Likewise, the first step of both UNSUPFIXED and UNSUPGEN involves mapping each language in S to English

[1] https://github.com/artetxem/vecmap

	PAIRWISE	BASELINE	ITERSUP	UNSUPFIXED	UNSUPGEN
en:es	78.30	80.00	79.93	79.93	**80.20**
en:de	70.73	72.13	**72.87**	71.27	70.80
en:fr	77.27	79.27	**79.80**	78.47	78.80
en:it	73.00	74.53	**75.20**	74.40	75.07
en:pt	73.67	74.87	75.00	75.53	**75.33**
es:en	78.20	78.20	78.40	79.27	**79.53**
es:de	63.33	60.40	**66.07**	61.67	63.67
es:fr	82.07	78.67	82.07	80.87	**82.20**
es:it	79.33	76.00	**80.87**	79.87	80.40
es:pt	83.13	81.40	**84.40**	83.33	83.60
de:en	69.20	69.20	70.20	70.00	**70.87**
de:es	61.67	57.73	**62.80**	59.47	61.33
de:fr	67.40	62.40	68.60	63.13	**69.00**
de:it	63.34	56.87	**64.20**	60.60	64.13
de:pt	53.20	48.53	**55.60**	52.07	54.40
fr:en	77.40	77.40	77.87	77.33	**78.00**
fr:es	78.93	74.93	78.80	78.53	**80.27**
fr:de	67.53	60.73	**67.60**	60.73	66.13
fr:it	78.67	74.60	**79.20**	75.60	78.00
fr:pt	72.47	69.73	**73.93**	72.20	73.27
it:en	73.60	73.60	73.87	74.44	**74.87**
it:es	**85.07**	80.87	**85.07**	83.87	85.00
it:de	**63.60**	56.53	62.80	59.00	63.00
it:fr	83.27	79.67	83.53	81.53	**84.00**
it:pt	77.00	71.67	77.00	76.60	**77.53**
pt:en	74.87	74.87	74.53	75.07	**75.53**
pt:es	89.13	87.60	**89.93**	87.93	89.07
pt:de	58.20	53.47	**58.60**	56.13	57.93
pt:fr	79.00	73.93	78.13	77.47	**79.60**
pt:it	76.60	72.67	77.13	76.53	**77.60**
avg	73.64	71.08	**74.47**	72.67	74.30

Table 4.1: The general results for each evaluation dictionary direction, where table entries represent accuracy on the dictionary induction task, using standard nearest-neighbor retrieval. Bold entries in each row represent the best performing algorithm in that direction. The last row represents the mean of each column.

via the dictionaries induced in an unsupervised manner. While UNSUPFIXED continues to employ the rest of the induced dictionaries throughout the iterative process, UNSUPGEN induces new dictionaries at each iteration step. It is apparent that both ITERSUP and UNSUPGEN benefit from the iterative process, with each successive iteration improving the overall alignment of the space. Perhaps expectedly, a closer look reveals that, in the case of ITERSUP and UNSUPGEN, the language direction only improves the mappings that relate to the direction, while all other mappings remain unaltered: e.g. mapping to es improves $es : en$ and $de : es$ but does not affect $pt : en$ or $fr : it$. This is intuitive, since we jointly map the set of languages $O = p \cup M \setminus q$ to E_q, where q is the target pivot language designated for the iteration. Doing so has the effect of preserving the multilingual invariance between the shared source language embedding spaces (e.g. the quality of the already completed mappings) when mapping to a new language. Each iteration can thus be thought of a language-specific *refinement step* that is performed until convergence. In our experiments, we observe that continuing the iterative process after the last mapping step is typically not beneficial and, on occasion, worsens the overall mapping. As such, we recommend halting the mapping process after one iteration of the entire algorithm, as convergence is typically reached then. Furthermore, though we do not experiment with separately mapping the languages in O to E_q for each iteration, we posit that this would have the effect of redundantly re-mapping every language per step and discarding the (beneficial) alignments produced by previous iterations.

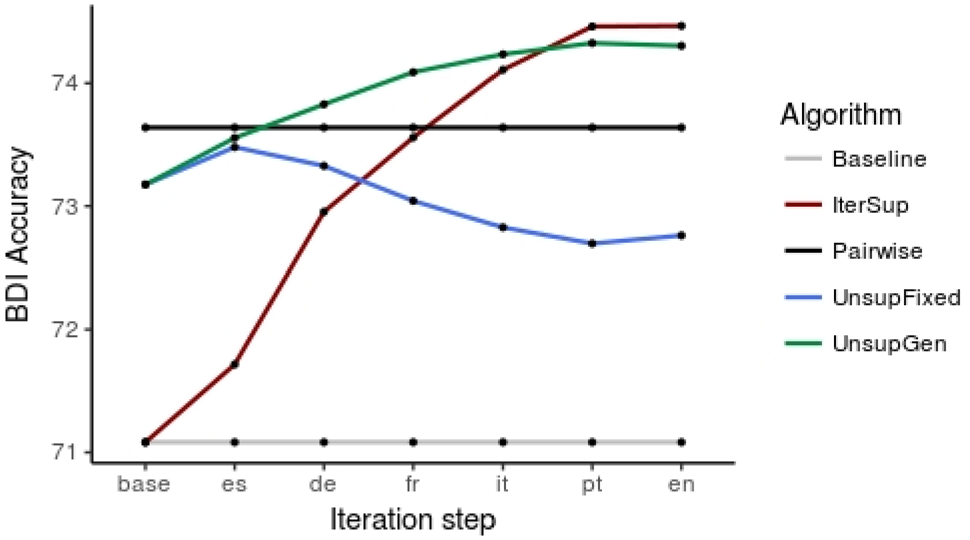


Figure 4.1: Dictionary induction results per iteration step, where *base* corresponds to the initial mapping to English.

We are also interested in the effect that training dictionary quality may have on the quality of mappings. Figure 4.1 demonstrates that, while UNSUPFIXED performs considerably well for the base mapping step, the quality of the mapping degrades after each consecutive iteration (with the slight exception of the *es* step). In contrast, both ITERSUP and UNSUPGEN continue to refine the mapping after each iteration, with ITERSUP improving at a faster rate. One possible explanation for this could be that the algorithm employed in Artetxe et al. (2018a) (upon which ITERSUP is based) succeeds at capturing domain similarities between embeddings, which can be leveraged to learn a domain-specific mapping in addition to a looser language-specific mapping. In our case, the dictionaries induced by this unsupervised process might contain a lot of Wikipedia-related content that is uniform across all languages, thereby resulting in a stagnation of or a degradation in BDI performance. In other words, the dictionaries induced in the first step of UNSUPFIXED are of poor general utility. In contrast, employing language-specific ground-truth dictionaries (as in ITERSUP) or inducing a new dictionary from two already-aligned embeddings (as in UNSUPGEN) might help to alleviate this effect.

4.2.1 Effects of Pivot Language

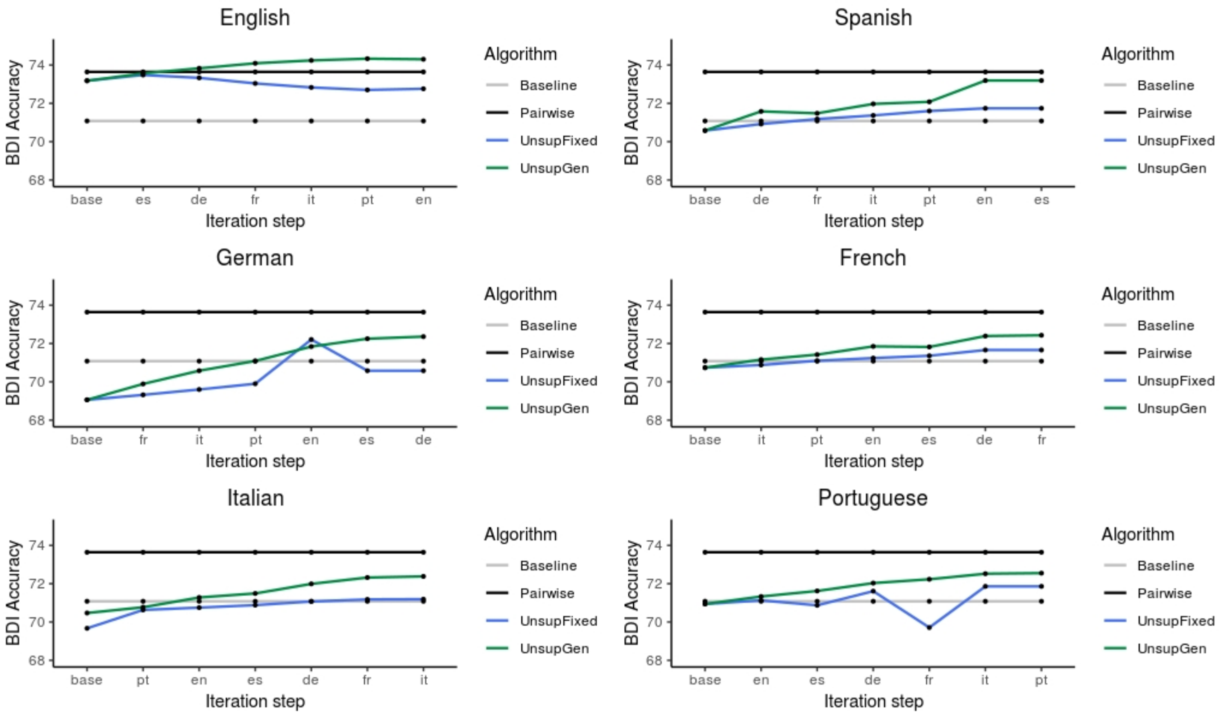


Figure 4.2: BDI results for UNSUPFIXED and UNSUPGEN for all possible pivots in our language set.

Due to the degradation in performance of UNSUPGEN, we decide to conduct an experiment where we choose a pivot language other than English in order to investigate the effect that pivot language choice has on the performance on either unsupervised algorithm. Our results are depicted in Figure 4.2. It is important to note here that we simply rotate the iteration order when choosing a new pivot (e.g. for Spanish, the iterations are $(es, de, fr, it, pt, en, es)$). We do so in order to investigate whether or not we can observe if the same accuracy curves reoccur or if entirely patterns emerge. The results that we report are quite striking. Perhaps the most noticeable observation is that UNSUPGEN does not outperfrom the PAIRWISE upper-bound for any pivot language other than English (though Spanish comes close). Furthermore, UNSUPFIXED performs erratically under the German and Portuguese settings, failing to beat BASELINE for the former [2]. There is no simple answer for this behavior, other than that the overall mapping procedure is clearly contingent on the choice of pivot language. Given that German is likely the most dissimilar language from all other languages in the set from a typological and, by extension, an isomorphic perspective, the initial "poor" mapping to its space likely gives it a disadvantage compared to other pivots. Though UNSUPGEN clearly learns to refine this intermediate mapping over the span of the iterative procedure via an initial "seed" dictionary, UNSUPFIXED employs a fixed set of dictionaries that may of poor quality throughout the procedure. As such, it appears that UNSUPGEN is a more reliable alignment algorithm than its counterpart. However, this is largely conjecture and a more thorough investigation into the quality of the induced dictionaries is needed.

[2] We continue to employ the English-pivot BASELINE here for fair comparison

CHAPTER 5

Multilingual Embeddings for Cross-lingual Transfer

Our cross-lingual experiments are concerned with measuring the utility of the cross-lingual signal enabled by multilingual embeddings as it relates to two common NLP tasks: POS-tagging and dependency parsing. We aim to do so in a straightforward fashion: training a model on one language and evaluating on another. Since the model that we employ is deep-learning-oriented, it accepts sequences of word embeddings as input. According to the literature covered in section 2.4, monolingually-trained word embeddings mapped to a single shared space are expected to facilitate a cross-lingual transfer, wherein a model can effectively treat embeddings for semantically equivalent words (cat::gato) as approximately the same (instead of just random noise as in the unaligned case). As such, it is reasonable for one to assume that a model (e.g. POS-tagger) trained on language like Norwegian would yield reasonable performance when evaluated on a closely-related language such as Danish. As such, we attempt to evaluate this principle under four different experimental conditions:

- the quality of embedding alignment.
- the training language.
- training bilingually or multilingually.
- the amount of in-language training data included.

Our training procedure is thus as follows:

1. Align embeddings for training and evaluation language to a single space.
2. Train model on training language with sequences of word embeddings as input.
3. Evaluate on evaluation language.

5.1 Model

Our model (shown in 5.1 is a joint POS-tagger/dependency parser that simultaneously predicts POS-tags and employs them as an additional input presentation to the parser. This is loosely inspired by Hashimoto et al. (2016), who propose a shared model for a multitude of NLP tasks, POS-tagging and depedency parsing included. Though we experiment with a variety of architectures, languages, and data-inclusion settings - standalone POS-tagging and dependency parsing among them - we find that the joint model yields the best results in terms of POS-tagging accuracy and UAS/LAS.

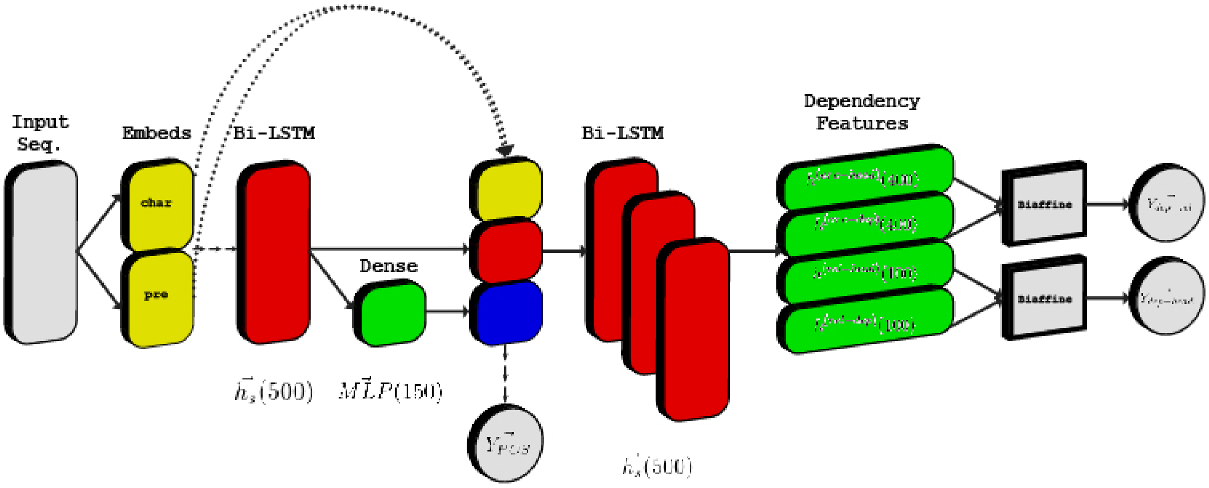


Figure 5.1: Our joint POS-tagger/dependency parser architecture.

The POS-tagging component of our model takes in sequences of pretrained word embeddings as input. Every word embedding is concatenated row-wise with the character embedding corresponding to the word. These representations are then passed to a BiLSTM and, following this, an MLP in order to extract higher-order features. The output of the MLP is passed to a softmax classifier which predicts the POS-tag for every word. It is important to make note of two things at this stage: first, we keep the word embedding layer frozen, which means that its gradient is not updated throughout the training procedure and the vectors are left as-is throughout. Though we experiment with unfreezing this layer, as well as including randomly initialized embeddings alongside it, we do not observe any improvement in results from doing so. We also posit that this is the most controlled manner for evaluating any observed transfer, since the word vectors are kept in-tact throughout training. Second, we include the character embeddings as a measure of the benefit that non-lexical features may have for tagger/parser performance as a whole. However, we consider their presence as an additional experimental setting and remove them when controlling for the cross-lingual signal at the word embedding level.

Our subsequent parser is a model that is based on Dozat and Manning's (2016) deep biaffine attention dependency parser. The main difference between our model and theirs is that A.) we employ a joint-learning objective alongside POS-tagging and B.) our input representations differ from theirs. Whereas Dozat and Manning (2016) learn POS embeddings from gold-standard tags and employ them as an additional input representation, we repurpose the vectors from the POS-tag softmax prediction layer as a proxy for this. Furthermore, they also include "holistic" word embeddings, which are in effect randomly initialized embeddings, albeit with a few modifications. In place of this, we employ the BiLSTM hidden layer that corresponds to each word from the POS-tagger. These two representations are then concatenated with the pretrained embeddings in order to serve as a modified input representation which is then passed to a stacked 3-layer BiLSTM. The output of the last BiLSTM is then used as input for four dense layers with a ReLU activation, producing four vector representations: a word as a dependent seeking its head; a word as a head seeking all its dependents; a word as a dependent deciding on its label; a word as head deciding on the labels of its dependents. These representations are then passed to biaffine softmax classifiers to produce a fully-connected labeled probabilistic dependency graph (Dozat and Manning, 2016). Finally, a non-projective maximum spanning tree parsing algorithm (Chu, 1965; Edmonds, 1967) is used to obtain a well-formed dependency tree.

5.2 Experimental Conditions

In order to fully assess the extent of the cross-lingual signal enabled by multilingual embeddings, we evaluate our models under a variety of conditions. We hope that a thorough investigation will provide insight for future research into cross-lingual learning via multilingual embeddings, and where they may fail.

5.2.1 Embedding Alignment

We experiment with three different mapping methods in our crosslingual experiments. The first is the BASELINE method from Chapter 3, where embeddings for a set of languages S are mapped to a single pivot language's embeddings E_p using a bilingual dictionary. In this case, we employ English as a pivot language, given the wealth of resources that exist between English and the rest of the world's languages. Our second method is UNSUPGEN, which we choose due to its comparable performance to the upper-bound PAIRWISE and fully-supervised ITERSUP, despite being completely unsupervised. For this case, we choose the evaluation language as the initial pivot and schedule the iterations alphabetically in regards to the name of the training languages. Lastly, we consider a baseline measure, where embeddings for all languages are left unaligned, which we call UNALIGN. We posit that this will reveal the extent to which the multilingual alignment aids the model transfer, or whether or not the cross-lingual signal is present at all.

5.2.2 Language Choice

In order to avoid biasing our results on a single language pair, we exploit the breadth of Universal Dependencies' coverage in order to experiment with a variety of languages. We settle on three evaluation languages and designate three different, but related languages as training data for each one. Unfortunately, though Universal Dependencies has broad coverage, many of its treebanks are very small and poorly annotated, thus making proper experimentation under full settings difficult. As such, our experiments are confined to the much-explored Indo-European macro-family, from which we extract three different families: Germanic, Romance, and Slavic [1]. Our criterion for choosing evaluation languages is simply to sort each treebank by number of tokens and choose the 4th-best resourced one. The top-3 treebanks in terms of number of tokens are then selected as the individual training languages for the evaluation language in question. We find tokens to be a better representation of the treebank's resources, as sorting by number of sentences produces a very different grouping that favors treebanks with very short, poorly annotated entries (e.g. Slovakian). Table 5.1 shows our final selection of languages.

family	**train languages**	**eval language**
Germanic	Dutch, German, Norwegian	Danish
Romance	Catalan, French, Spanish	Italian
Slavic	Bulgarian, Croatian, Czech	Slovenian

Table 5.1: Our selection of languages for the cross-lingual experiments.

5.2.3 Bilingual and Multilingual Training

Our basic scenario in the cross-lingual experiments is training on a single designated training language and evaluating on a related evaluation language. We refer to this as the Bilingual case. However, in order to take advantage of the multilingual embeddings, we add a condition wherein we train on the combination of all training languages per family. We refer to this as the Multilingual case. This is accomplished by simply concatenating the training treebanks together and evaluating on the evaluation language as we typically would. Doing this is trivial, since our embeddings are multilingual from the start. We posit that this is a form of dataset expansion and expect the transfer to be more noticeable in the multilingual case.

5.2.4 In-Language Dataset Inclusion

Though out experiments are mainly concerned with the zero-shot scenario, where models are trained on one language and evaluated on another, we are also interested in cases where portions

[1] These are the only families in Universal Dependencies that match our criteria of housing 4 or more languages (3 train, 1 evaluation language).

of (unseen) evaluation language training data are included during training. We posit that gradually including more evaluation language data will nudge the model towards learning better representations of the evaluation language, which would (ideally) yield better performance for both POS-tagging and dependency parsing. In effect, this condition is meant to simulate varying degrees of resource-availability, which could, in turn, yield insights in regards to when the cross-lingual signal is useful and when it is not. In all cases, we validate the dataset inclusion by considering a baseline wherein a model is trained on the same amount of evaluation language data without the support of any of the training data.

CHAPTER 6

CROSS-LINGUAL LEARNING EXPERIMENTS

This chapter is concerned with delivering the results for our cross-lingual POS-tagging and dependency parsing experiments. First, we outline the data we employ for our experiments and describe the procedure we follow in order to partition it. Next, we provide the hyperparameters for our model and briefly describe its implementation. Lastly, we present the results for a variety of different experimental conditions and supplement them with a detailed discussion.

6.1 Data

6.1.1 UD Treebanks

We employ treebanks from Universal Dependencies release 2.1 (Nivre et al., 2017), which collects 102 different treebanks. The specific languages that we experiment with are outlined in Table 5.1, as well as the justification for choosing them. It is important to note that, in cases where two or more treebanks exist for a single language, we choose the canonical name - e.g. `UD_Slovenian` instead of `UD_Slovenian-SST`. In order to account for the vast disparity in number of training samples among treebanks, we choose a cutoff of 7,500 sentences for all training languages. This number was determined by sorting all training treebanks by number of total sentences and selecting the size of lowest-ranked treebank (`UD_Croatian`) in order to account for as much data as uniformly possible [1]. We follow the same strategy with our evaluation languages and set a sentence cutoff of 4,000 according to Danish - the lowest-ranked evaluation treebank. Furthermore, for our data inclusion experiments, we decide on windows of 0, 75, 250, 1000, and 4000 sentences to gradually include during training. We posit that this represents an adequate spread of data that can provide insights about when cross-lingual learning is most useful, if at

[1] We round down to the nearest 500 sentences.

layer	**dimension**
word embeddings	300
char embeddings	150
POS BiLSTM	200
POS MLP	400
parse BiLSTM	500
parse MLP arc-head	400
parse MLP arc-dep	400
parse MLP rel-head	150
parse MLP rel-dep	150

Table 6.1: Hyperparameters for our joint POS-tagger and dependency parser.

all. In all cases, we use the designated TRAIN, DEVELOPMENT, and TEST splits as provided by the UD release.

6.1.2 Embeddings and Dictionaries

We employ the same pretrained FASTTEXT embeddings provided by Facebook (Bojanowski et al., 2016) for our cross-lingual experiments that we used in the multilingual experiments. We also normalize each set of embeddings to be unit length and mean-centered and then re-normalize for unit length, but do not truncate as in the multilingual mapping experiments. For the BASELINE alignment, once again employ the dictionaries provided by Facebook's MUSE project (Conneau et al., 2017). In this case, however, embeddings for every language are mapped to English, though English is not included among any of the language family groups. We do this to induce a baseline alignment, against which we evaluate a theoretically better alignment in the form of UNSUPGEN.

6.2 Model

Our model is implemented in the PyTorch library for Python (Paszke et al., 2017) based on the description in Dozat et al. (2017). Though we experiment with various hyperparameters, we ultimately settle on the ones outlined in Table 6.1 and employ them for all language families. In addition this, we employ Cross-Entropy as a loss function and the Adam algorithm for optimization, for which we set the initial learning rate at 0.001.

6.3 Word Embedding Experiments and Results

Table 6.2 shows the results for our cross-lingual experiments in the zero-shot transfer setting. It is important to note that both tagger and parser architectures here are entirely lexicalized, where the sole input representation is word embeddings. Though we do not expect performance

UAS			
	Unalign	Baseline	UnsupGen
de	0.09438	**0.16562**	0.13080
nl	0.09219	**0.11065**	0.08620
no	0.05390	0.03272	**0.08091**
germanic	0.04180	0.09827	0.04580
ca	**0.13833**	0.05597	0.12960
es	0.07853	**0.10694**	0.10147
fr	0.02966	**0.10857**	0.05981
romance	0.04848	**0.12643**	0.12640
bg	0.04731	0.06930	**0.09896**
cs	0.15614	**0.17300**	0.13298
hr	0.11494	0.11238	**0.12339**
slavic	**0.15479**	0.13341	0.13340
LAS			
de	0.01417	**0.02155**	0.01477
nl	0.01517	**0.02305**	0.01566
no	0.01100	**0.03272**	0.01786
germanic	0.00938	0.00748	**0.00980**
ca	0.01315	0.01277	**0.01421**
es	0.01181	**0.01786**	0.01286
fr	0.01046	**0.01546**	0.01498
romance	0.01334	**0.01853**	0.0185
bg	**0.01158**	0.00940	0.01002
cs	**0.01819**	0.01560	0.01328
hr	0.00923	**0.01257**	0.00945
slavic	**0.01513**	0.01172	0.0117
POS			
de	0.10665	0.05896	**0.13788**
nl	**0.14028**	0.10735	0.11075
no	**0.06280**	0.05787	0.05827
germanic	0.10885	**0.11324**	0.11170
ca	0.07603	0.09984	**0.10195**
es	0.09465	0.09580	**0.09878**
fr	0.09628	**0.12182**	0.10099
romance	**0.09830**	0.09158	0.09160
bg	0.03552	0.0338	**0.13085**
cs	0.08674	**0.13660**	0.12673
hr	0.06770	**0.07523**	0.02785
slavic	**0.09114**	0.03474	0.03470

Table 6.2: Results for the cross-lingual POS-tagging and dependency experiments under the zero-shot data setting. Input representations include only frozen word-embeddings.

that is close to the monolingual state-of-the-art, we nonetheless anticipate a noticeable transfer from the training language to the evaluation language via embedding alignment. Our results, however, demonstrate that *no* transfer is observed in the zero-shot setting, regardless of alignment quality. For example, in regards to POS-tagging, there is no discernible pattern between the accuracies yielded by the unaligned embeddings and the embeddings aligned via the Baseline or UnsupGen algorithms. The same can be said for the UAS results, though the majority of scores seems to fall in favor of the embeddings aligned via either algorithm. LAS scores are understandably low given the reliance of the metric on good arc-prediction and POS-information. Since the parser relies only on supposedly transferred lexical features, it will often fail at predicting proper arcs. Furthermore, since arc labels are often directly related to POS-information (which is effectively absent in this setting), it is understandable that the LAS performance is so poor.

The choice of training language is likewise uninformative. For example, though Dutch yields the highest POS-accuracy for Danish, German yields the highest UAS score. This is unexpected, as Norwegian - a Scandinavian language closely related to Danish in syntax and morphology- seems to consistently produce the lowest scores out of the three Germanic languages. Furthermore, it appears that training multilingually yields *no* benefit to training bilingually. This is despite the tripled training set applied in all instances. In the multilingual situations, it appears that adding more languages to the training set effectively adds noise that the model struggles to cope with. Indeed, when neither the tagger nor the parser succeeds in learning from just one related language, it cannot be expected to learn from a combination of them.

It is difficult to isolate the reason as to why embedding alignment bears no discernible positive effect in the zero-shot setting. Indeed, though lexical features are less salient than non-lexical features for cross-lingual transfer (Guo et al., 2015), we still expect to observe a signal across languages. Our results, however, fail to corroborate the findings reported in other work. One possible explanation for this could be that, since the size of the Wikipedia corpus on which our FastText embeddings are trained varies highly by language, the quality of the embeddings themselves is likely to be poor for languages with small Wikipedias. Indeed, while the German Wikipedia is the 4th largest of all languages, comprising 2,214,257 articles, Bulgarian is the 33rd largest, with only 245,289 articles[2]. However, we observe no benefit even when large corpora are concerned: e.g. French and Spanish embeddings (4th and 8th largest Wikipedias, 2,214,257 and 1,464,375 articles, respectively) transferring to Italian (9th largest Wikipedia, 1,457,656 articles). This suggests that the size of the training corpus is not to blame. Another possible factor could be the FastText embedding algorithm itself. Indeed, most literature in the cross-lingual learning domain (e.g. Fang and Cohn (2017); Guo et al. (2015)) makes use of Word2Vec, training a separate set of embeddings for each language per experiment. Given that our language set comprises 12 different languages, we regrettably do not possess the computational resources to

[2] Wikipedia size statistics taken from: `https://meta.wikimedia.org/wiki/List`$_{o}f_{W}ikipedias$

train 12 separate WORD2VEC models in order to investigate the possible superiority of its word representations over those learned by FASTTEXT. However, given the competitive performance of FASTTEXT on a variety of intrinsic and extrinsic tasks (Bojanowski et al., 2016) in comparison to WORD2VEC - as well as the results of our own embedding alignment experiments in Section 4 - we surmise that the choice of embedding algorithm also cannot be blamed for these poor results.

6.3.1 In-language Training

Regarding the benefit of training alongside in-language data, Figures 6.1, 6.2, and 6.3, show our results for the UNALIGN, BASELINE, and UNSUPGEN embeddings, respectively. We also include a monolingual evaluation language baseline as a point of comparison. Here, it is apparent that, for the 75, 250, and 1000 sentence settings, training alongside other languages induces a (albeit weak) cross-lingual that is transferred to the evaluation language. This suggests that "seeding" the model with evaluation language data allows it to learn some sense of "shared" representations among the training languages that is relevant to the task at hand. This is contrast to the zero shot setting, where no transfer is observed. More interestingly, it is evident that, once again, there is *no* observed benefit to aligning embeddings to a single space. Though performance varies per language, there is nonetheless no clear trend that suggests the transfer is owed to alignment. Instead, one can surmise that the observed cross-lingual signal may be inherited from higher-order features that are inferred by the LSTM/MLP and are not directly interpretable. Likewise, it is possible that, in training alongside the evaluation language, the model might learn low-level positional (i.e. word identity) features that persist among similar languages and are not inherently lexical in nature. However, this is mostly conjecture and a more detailed analysis is required.

6.3.2 Character Embedding Experiments and Results

In attempt to see what benefit other, non-lexical input representations may provide for cross-lingual transfer, we add randomly-initialized character embeddings to our model by concatenating them to the pre-trained word embeddings. In comparison with the word embedding-only models, the addition of character embeddings dramatically improves performance. Furthermore, we observe no benefit in regards to the embedding alignment strategy, confirming our findings in the previous section. However, the inclusion of character information helps shed light on other phenomena that were otherwise unobserved throughout the standalone word embedding experiments. As such, figure 6.4 shows these results only for the UNALIGN embeddings. Based on Figure 6.4, it is clear that the same trend is observed throughout the standalone word embedding experiments persists for the character embedding experiments as well. Namely, it is evident that the model infers a cross-lingual signal that aids performance in all data settings except the full-resource scenario. However, while the signal is minimal - if not non-existent - in the zero-shot setting when trained on standalone word-embeddings, it is much more pronounced when character

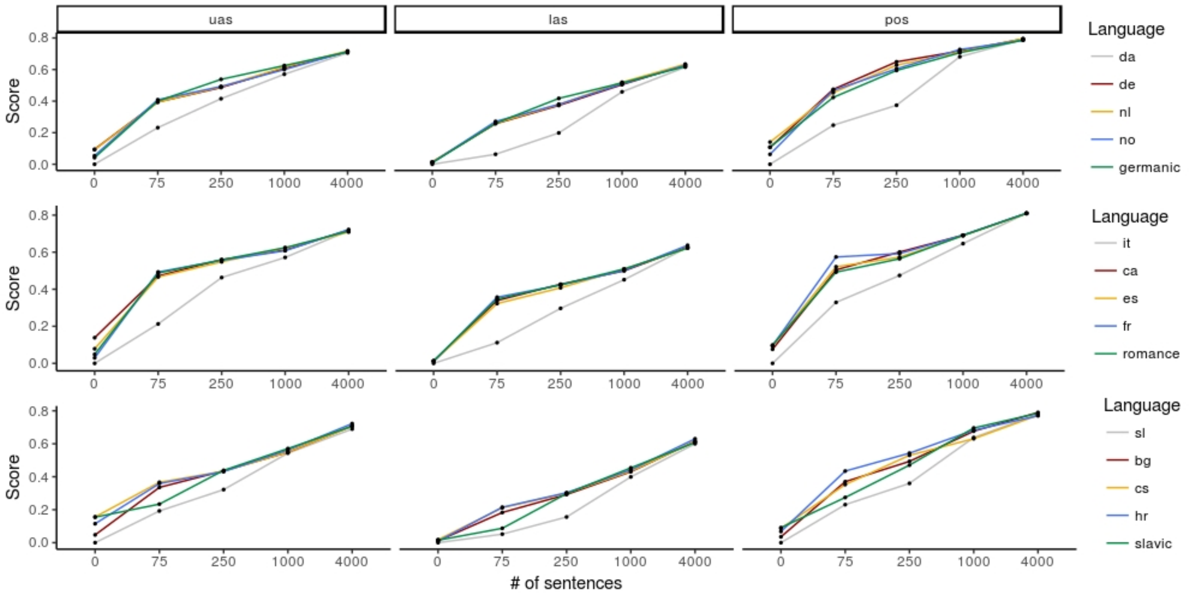


Figure 6.1: Transfer performance for each task per language group with **unaligned** word embeddings only. Columns represent the UAS, LAS, and POS metrics, respectively, while rows represent the language family.

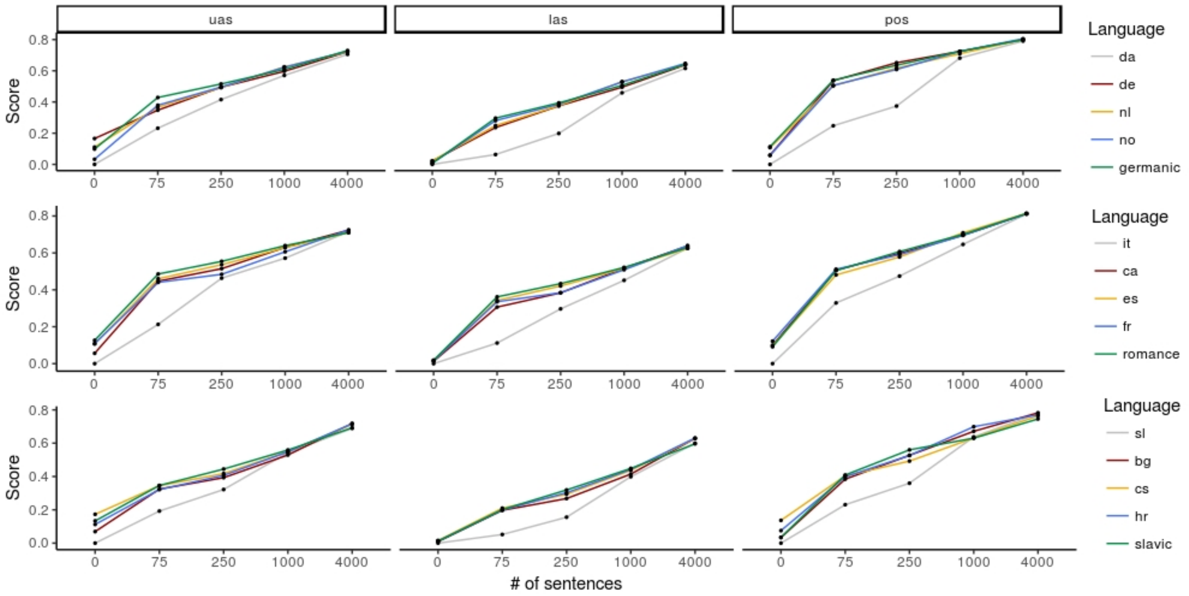


Figure 6.2: Transfer performance for each task per language group with BASELINE word embeddings only. Columns represent the UAS, LAS, and POS metrics, respectively, while rows represent the language family.

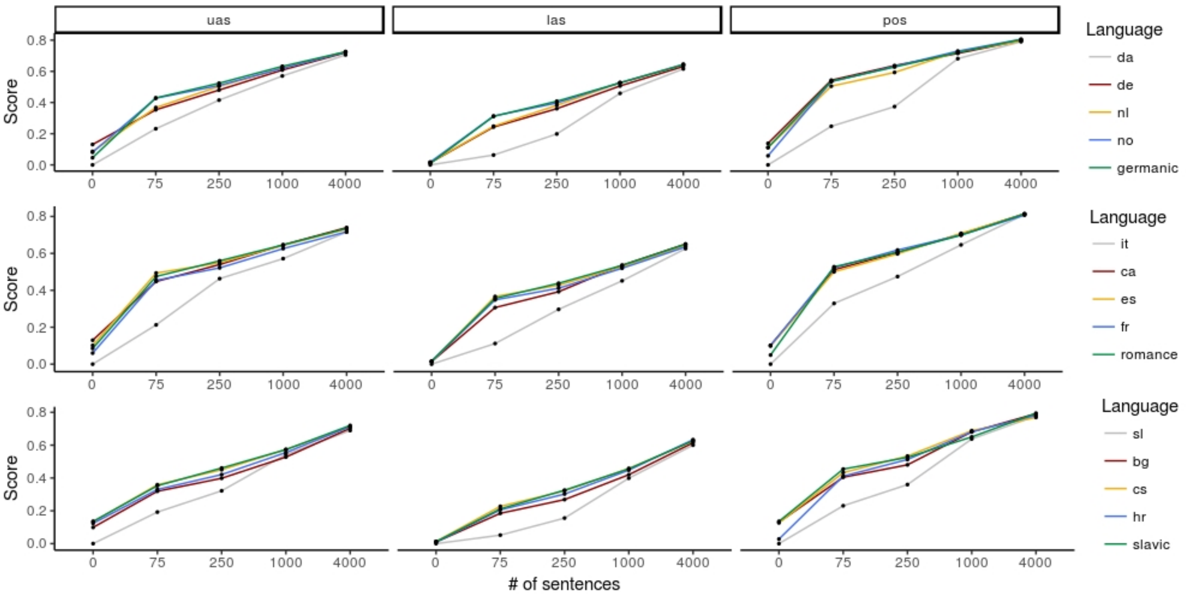


Figure 6.3: Transfer performance for each task per language group with UNSUPGEN word embeddings only. Columns represent the UAS, LAS, and POS metrics, respectively, while rows represent the language family.

embeddings are added. This speaks to the general utility of de-lexicalized representations of words for both POS-tagging and dependency parsing tasks.

The inclusion of character embeddings also helps us to answer questions relating to two of our experimental settings: the effect of the training language and the effect of training bilingually or multilingually. In regards to the first setting, it is clear that, for the Germanic group, training alongside Norwegian yields a clear performance boost for Danish when compared to more distantly-related German and Dutch. A similar effect can be observed in the Slavic group, for Croatian in regards to Slovenian. This is much less pronounced the Romance group, except for the zero-shot setting, where French appears to be the most beneficial. Furthermore, it is apparent that, in almost all data settings, training on the concatenation of all languages within a language family is almost always better than training on standalone languages. This persists across all language families and tasks and is most explicit in the zero-shot setting. This is expected, since the training data is effectively tripled for the multilingual settings.

6.3.3 Comparison to Previous Work

Given the relative uniformity of our results (i.e. that there is no clear benefit in aligning word embeddings for POS-tagging and dependency parsing, under any setting), we do not surmise that evaluating our approach on the language pairs presented in other work (which are different than ours) will yield drastically different results. However, it is nonetheless important to note that, though the papers surveyed in Sections 2.4.3 and 2.4.5 report a significant benefit to the

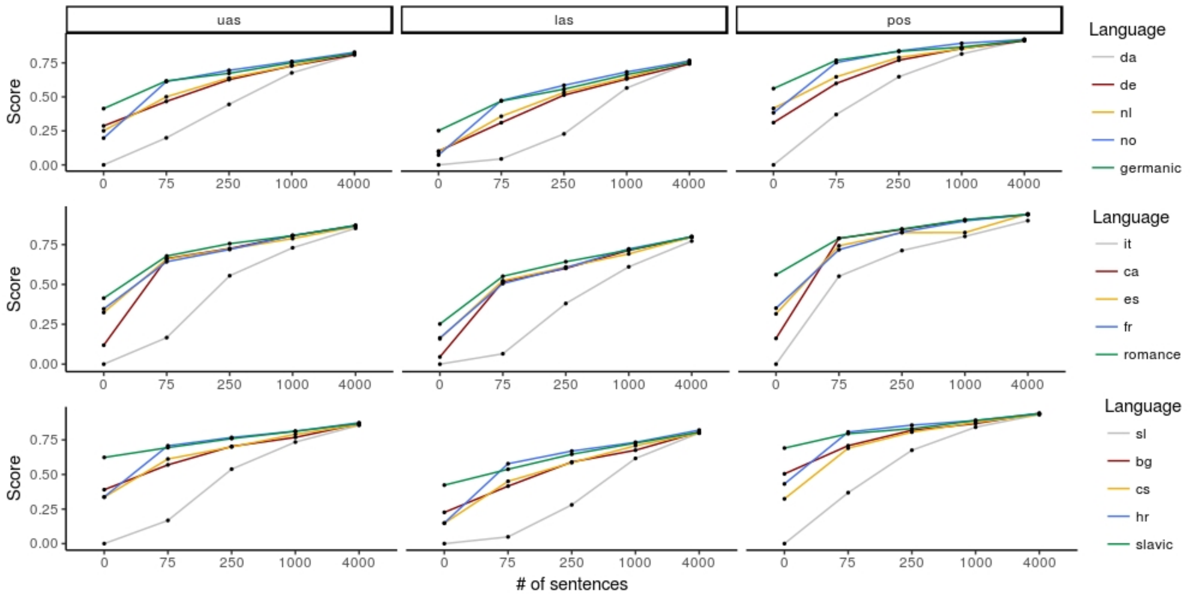


Figure 6.4: Transfer performance for each task per language group with **unaligned** embeddings and **character embeddings**. Columns represent the UAS, LAS, and POS metrics, respectively, while rows represent the language family.

use of multilingual word embeddings in cross-lingual POS-tagging and dependency parsing, our experiments do not corroborate such claims. Indeed, while much of the existing literature in cross-lingual learning places multilingual word embeddings at the forefront of research - citing embedding alignment between two languages as the “key” to unlocking cross-lingual transfer (Ruder et al., 2017) - it does so amidst other, much more beneficial feature representations and complex model architectures. As such, we feel that is important to contextualize our experiments among this work.

For example, in the case of POS-tagging, Fang and Cohn (2017) base their experiments around embedding alignment, citing multilingual embeddings as the principal source of the positive cross-lingual transfer that they observe. However, though they do include a random POS-tag baseline, they do not experiment with unaligned, monolingual embeddings. Furthermore, the model that they employ is a multi-task model that effectively obscures the source of cross-lingual transfer. Indeed, as shown in our in-language data inclusion experiments, a model may infer a source of transfer when it is presented with disparate sources of input (e.g. a seed of Danish data when training alongside Norwegian), irrespective of embedding alignment. As such, since the model employed by Fang and Cohn (2017) is designed to leverage various input sources (noisy POS-tags and gold POS-tags), it may learn to leverage these features much better than our simple model, even with monolingual embeddings.

In regards to cross-lingual dependency-parsing, Guo et al. (2015) report positive results with de-lexicalized features in the zero-shot setting. In their experiments, the inclusion of multilingual word embeddings via a novel alignment method (multi-CCA) improves the parser by an average of ~ 4% UAS/LAS. Like Fang and Cohn (2017), however, they fail to contextualize the benefit of embedding alignment by performing experiments with multilingual embeddings only. The same can be said for Ammar et al. (2016), who employ a variety of input representations for their parser - multilingual word embeddings among them. However, embedding alignment is not the focus of their work, unlike in Guo et al. (2015), where it is the chief motivation. In the context of these papers, our experiments show that embedding alignment yields no cross-lingual signal for dependency parsing whatsoever and should not be trusted as the source of improved parsing performance that is reported. Instead, our results suggest that whatever benefit is observed is likely owed to other, higher-order features that are otherwise inferred by a model, regardless of alignment. Indeed, in the case of Guo et al. (2015), whatever benefit word embeddings may (or may not) have in cross-lingual learning is small in comparison to non-lexical features. This is corroborated by our own character embedding experiments, where a strong cross-lingual is indeed observed in all cases.

Chapter 7

Conclusion

In this work, we set out with two goals in mind. Our first goal was to venture beyond bilingual word embedding mappings and investigate methods of aligning word embeddings multilingually. We proposed three novel algorithms for doing so - one of them supervised (ITERSUP) and two unsupervised (UNSUPFIXED and UNSUPGEN). Comparing each algorithm to a dictionary-based bilingual upper-bound and widely employed baseline for six languages (English, Spanish, German, French, Italian, Portuguese), we found that ITERSUP produced the best results while UNSUPGEN was close behind it in performance - despite being competely unsupervised. These two algorithms comfortably outperformed the baseline measure as well as the upper-bound, showcasing the reliability of our overall approach. In addition to this, we investigated the effect of the chosen pivot language on both unsupervised systems. These experiments revealed that the choice of pivot language directly impacted the performance of the algorithm as a whole, with English producing the best results and German the worst. We posit that, since the latter as comparatively the most distant language in the set and, as such, comparatively least isomorphic, the resultant multi-lingual space was likewise of the worst quality. Ultimately, we find that, despite the pivot language, UNSUPGEN refines the mapping across each iteration of the algorithm and produces results that are reliable above the baseline, UNSUPFIXED performs erratically and should be avoided in favor of the former algorithm.

Our second goal was to investigate the utility of multilingual embeddings in two common cross-lingual transfer learning scenarios: POS-tagging and dependency parsing. To do so, we trained a joint POS-tagger/dependency parser on Universal Dependencies treebanks for a variety of Indo-European languages and evaluated it on other, closely related languages. When experimenting with pretrained word embeddings as the sole input representation, we did not observe any benefit from their alignment to a common space - quality of alignment nonwithstanding. This led us to conclude that - under our experimental settings - multilingual embeddings *do not* yield a cross-lingual signal that can be leveraged for syntactic tasks. Though we did observe a benefit to training jointly with held-out evaluation language training data, we surmised that, since this was observed for unaligned embeddings in addition to the aligned case, our network was learning higher-order or positional features that were not necessarily lexical. In an attempt to experiment with other, non-lexical features, we added randomly-initialized character embeddings to our input representation. We observed a clear benefit in doing so, highlighting the effect of morphological/delexicalized features for the cross-lingual syntactic tasks. Overall, our experiments lead us to cast the general utility of multilingual embeddings for syntactic tasks into doubt, urging future work in the cross-lingual domain to investigate precisely where the cross-lingual signal comes from.

In the opening chapter of this thesis, we posed four research questions to guide our experiments. Having concluded our project, we can now answer them:

Can we produce better-quality embeddings than the oft-employed common language approach?

- Yes, all of our proposed algorithms beat this baseline in terms of accuracy on the bilingual dictionary induction (BDI) task. In addition to this, two of our algorithms (ITERSUP and UNSUPGEN) perform better than an entirely supervised bilingual system, showing the reliability of our approach.

Can unsupervised mapping algorithms produce higher-quality multilingual embeddings than their dictionary-based, supervised counterparts?

- In our experiments, the fully supervised algorithm ITERSUP performs slightly better than our best unsupervised system UNSUPGEN. However, the difference in performance is very slight.

Can we observe a beneficial cross-lingual signal in regards to the POS-tagging and dependency parsing tasks?

- In our experiments, we **do not** observe any cross-lingual signal when training exclusively on word embeddings in the zero-shot scenario. In general, the unaligned embeddings perform just as well as the multilingual embeddings aligned by either BASELINE or UNSUPGEN approaches.

If so, does the cross-lingual signal persist when accounting for increasingly higher resource scenarios?

- Though we do not observe a cross-lingual signal in the zero-shot scenario, we nonetheless make note of an additional transfer effect that occurs when we add held-out evaluation language data to the training set. We attribute this to the network learning higher-order or positional features that may persist regardless of alignment.

7.1 Future Work

In regards to multilingual embeddings, we would like to continue the work described in this thesis in several directions. Namely, we would like to investigate larger multilingual spaces and different combinations of languages in order to yield stronger insights about our proposed methods. Indeed, our experiments so far have focused on a set of very much related Indo-European languages, whose similarity has the potential to influence the overall quality of the multilingual space. As such, we think that it would be worthwhile to experiment with languages that are inherent differently in terms of typology (e.g. Basque, Turkish, Chinese). In addition to this, would like to investigate methods for incorporating the re-weighting approach of Artetxe et al. (2018b), which has proven to improve mappings in the bilingual case.

Our cross-lingual experiments have admittedly raised more questions than they have answers. Before discounting the utility of multilingual embeddings for cross-lingual syntactic tasks entirely, we would like to experiment with more sophisticated network architectures - e.g. multi-task learning as in Fang and Cohn (2017). Furthermore, given that we do observe a minimal source of transfer when adding held-out evaluation language data to the training set, it would be interesting to conduct a thorough investigation into the source of the transfer. Also, given that the quality of pretrained FASTTEXT embeddings varies considerably across languages, we would like to experiment with other training domains and embedding algorithms in order to investigate the effect of the quality of embeddings on cross-lingual tagger/parser performance as a whole. Lastly, since both of our tasks were syntactic in nature, further experimentation with semantically-oriented tasks (e.g. semantic tagging, document classification) would provide a proper overview for when multilingual embeddings have a proven benefit and when they do not.

BIBLIOGRAPHY

Željko Agić, Anders Johannsen, Barbara Plank, Héctor Martínez Alonso, Natalie Schluter, and Anders Søgaard. Multilingual projection for parsing truly low-resource languages. *Transactions of the Association for Computational Linguistics*, 4:301–312, 2016.

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A Smith. Many languages, one parser. *arXiv preprint arXiv:1602.01595*, 2016.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, 2016.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 451–462, 2017.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia, July 2018a. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, 2018b.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 238–247, 2014.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

Yoeng-Jin Chu. On the shortest arborescence of a directed graph. *Science Sinica*, 14:1396–1400, 1965.

Stephen Clark. Vector space models of lexical meaning. *The Handbook of Contemporary semantic theory*, pages 493–522, 2015.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017.

Ryan Cotterell and Hinrich Schütze. Morphological word-embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1287–1292, 2015.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. Improving zero-shot learning by mitigating the hubness problem. *arXiv preprint arXiv:1412.6568*, 2014.

Timothy Dozat and Christopher D Manning. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*, 2016.

Timothy Dozat, Peng Qi, and Christopher D Manning. Stanford's graph-based neural dependency parser at the conll 2017 shared task. *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, 2017.

Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. Multilingual training of crosslingual word embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 894–904, 2017.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A Smith. Recurrent neural network grammars. *arXiv preprint arXiv:1602.07776*, 2016.

Jack Edmonds. Optimum branchings. *Journal of Research of the national Bureau of Standards B*, 71(4):233–240, 1967.

Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

Stefan Evert. Distributional semantic models. In *NAACL HLT 2010 Tutorial Abstracts*, pages 15–18. Association for Computational Linguistics, 2010. URL `http://www.aclweb.org/anthology/N10-4006`.

Stefan Evert. The statistics of word cooccurrences: word pairs and collocations. 2005.

Meng Fang and Trevor Cohn. Model transfer for tagging low-resource languages using a bilingual dictionary. *arXiv preprint arXiv:1705.00424*, 2017.

Manaal Faruqui and Chris Dyer. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471, 2014.

John R Firth. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, 1957.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. Bilbowa: Fast bilingual distributed representations without word alignments. In *International Conference on Machine Learning*, pages 748–756, 2015.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. Cross-lingual dependency parsing based on distributed representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1234–1244, 2015.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. A representation learning framework for multi-source transfer parsing. In *AAAI*, pages 2734–2740, 2016.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. A joint many-task model: Growing a neural network for multiple nlp tasks. *arXiv preprint arXiv:1611.01587*, 2016.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Dan Jurafsky and James H Martin. *Speech and language processing*, volume 3. Pearson London, 2014.

Eliyahu Kiperwasser and Yoav Goldberg. Simple and accurate dependency parsing using bidirectional lstm feature representations. *arXiv preprint arXiv:1603.04351*, 2016.

Thang Luong, Richard Socher, and Christopher Manning. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113, 2013.

Thang Luong, Hieu Pham, and Christopher D Manning. Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159, 2015.

Ryan McDonald, Slav Petrov, and Keith Hall. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the conference on empirical methods in natural language processing*, pages 62–72. Association for Computational Linguistics, 2011.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013b.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013c.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, and others. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, 2016. URL `http://www.lrec-conf.org/proceedings/lrec2016/pdf/348_Paper.pdf`.

Joakim Nivre, Lars Ahrenberg ˇZeljko Agic, et al. Universal dependencies 2.0-conll 2017 shared task development and test data. *LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University*, 2017.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. *arXiv preprint arXiv:1604.05529*, 2016.

Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(Sep): 2487–2531, 2010.

Sebastian Ruder and Barbara Plank. Strong baselines for neural semi-supervised learning under domain shift. *arXiv preprint arXiv:1804.09530*, 2018.

Sebastian Ruder, Ivan Vulic, and Anders Søgaard. A survey of cross-lingual embedding models. *CoRR, abs/1706.04902*, 2017.

Hinrich Schütze. Word space. In *Advances in neural information processing systems*, pages 895–902, 1993.

Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *arXiv preprint arXiv:1702.03859*, 2017.

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. On the limitations of unsupervised bilingual dictionary induction. *arXiv preprint arXiv:1805.03620*, 2018.

Peter D Turney. A uniform approach to analogies, synonyms, antonyms, and associations. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 905–912. Association for Computational Linguistics, 2008.

Ivan Vulic and Anna-Leena Korhonen. On the role of seed lexicons in learning bilingual word embeddings. 2016.

Peilu Wang, Yao Qian, Frank K Soong, Lei He, and Hai Zhao. Part-of-speech tagging with bidirectional long short-term memory recurrent neural network. *arXiv preprint arXiv:1510.06168*, 2015.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Charagram: Embedding words and sentences via character n-grams. *arXiv preprint arXiv:1607.02789*, 2016.

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1006–1011, 2015.

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

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. Adversarial training for unsupervised bilingual lexicon induction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1959–1970, 2017.

Yuan Zhang, David Gaddy, Regina Barzilay, and Tommi Jaakkola. Ten pairs to tag-multilingual pos tagging via coarse mapping between embeddings. Association for Computational Linguistics, 2016.