Natural Language Processing for Resource-Poor Languages

A thesis presented
by

Long Thanh Duong

to
The Department of Computing and Information System
in total fulfillment of the requirements
for the degree of
PhD

The University of Melbourne
Melbourne, Australia
Feb 2017

Declaration

This is to certify that:

(i) the thesis comprises only my original work towards the PhD except where indicated in the Preface;

(ii) due acknowledgement has been made in the text to all other material used;

(iii) the thesis is fewer than 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.

Signed: _____ Date: _____

Thesis advisor(s)
A/Prof. Steven Bird
Dr. Trevor Cohn

Author
Long Thanh Duong

Natural Language Processing for Resource-Poor Languages

Abstract

Contents

Title Page i
Abstract iii
Table of Contents iv
List of Figures vi
List of Tables vii
Citations to Previously Published Work viii
Acknowledgments ix
Dedication x

1 Introduction **1**
1.1 Motivation 1
1.2 Research Questions 4
1.3 Scope 5
1.4 Contributions 6
1.5 Thesis Overview 10

2 Background **11**
2.1 Low-resource Natural Language Processing 11
2.1.1 What is a low-resource language? 11
2.1.2 What resources can we expect? 13
2.1.3 Transfer learning 18
2.1.4 Notable work 20
2.2 POS tagging 20
2.2.1 Typologically related information 23
2.2.2 Projected information 24
2.2.3 Dictionary Information 25
2.2.4 Small Annotated Data Information 26
2.2.5 Universal tagset 28
2.3 Dependency Parsing 29
2.3.1 Supervised dependency parsing 31
2.3.2 Low-resource Dependency Parsing 35
2.3.3 Summary of Approaches 37

2.4 Crosslingual Word Embeddings . . . 38
2.4.1 Monolingual Word Embeddings . . . 39
2.4.2 Building Crosslingual Word Embeddings . . . 41
2.4.3 Evaluation . . . 45
2.5 Unwritten Language Processing . . . 46
2.5.1 Unsupervised segmentation and lexical discovery . . . 47
2.5.2 Speech recognition for low-resource language . . . 50
2.5.3 Low-resource Speech Data Collection . . . 53
2.5.4 The Propose Task . . . 54

3 Research Summary **56**
3.1 Publications . . . 57
3.1.1 EMNLP 2014 . . . 57
3.1.2 ACL 2015 . . . 71
3.1.3 EMNLP 2015 . . . 78
3.1.4 CoNLL 2015 . . . 89
3.1.5 EMNLP 2016 . . . 100
3.1.6 EACL 2017 - if accept . . . 112
3.1.7 NAACL 2016 . . . 112
3.2 Evaluation of Contribution . . . 124
3.2.1 Research question revisited . . . 124
3.3 Future Work . . . 124
3.4 Conclusion . . . 124

A Other Papers **143**

List of Figures

1.1 Fraction of world population (percentage) by number of native speaker in 2007. This diagram should be viewed in color. (source Wikipedia) 2

2.1 Examples of part-of-speech projection from English to German using parallel text. No tag is given to the German word *klitzkie* that is not aligned. 18
2.2 Phrase structure tree (Left) and Dependency tree (Right) of the same sentence. 29

List of Tables

2.1 Number of languages having more than minimum number of expression in Wiktionary and Panlex. The number for Panlex is from Kamholz *et al.* (2014). 17
2.2 Notable related work on low-resource natural language processing. . 21
2.3 Previously published token-level POS tagging accuracy 25
2.4 The size of annotated data, number of tags included and missing for all considered languages 27
2.5 Unlabelled attachment score (UAS) of different models across seven languages. 37
2.6 Summary of crosslingual word embeddings papers 44

Citations to Previously Published Work

Large portions of chapter **??** have appeared in the following paper:

> **Long Duong**, Paul Cook, Steven Bird, and Pavel Pecina. 2013. Simpler unsupervised POS tagging with bilingual projections. In *Proceedings of the main conference 51th Annual Meeting of the Association for Computational Linguistics (ACL 2013)*. Association for Computational Linguistics, Sofia, Bulgaria

Acknowledgments

I would like to thanks my supervisors, family and friends....

Dedicated to youse all.

Chapter 1

Introduction

1.1 Motivation

Natural Language Processing (NLP) is an active field of research aim at teaching computer to understand human language. Achieving that goal is not easy as computer has to understand many aspect of the languages such as syntactic, semantic with respect to different input formats such as raw text, image and speech. Most NLP algorithms employ some form of machine learning techniques. Recently, many advancements in NLP are realized thanks to more computing resource, better understanding of the algorithm and most importantly, more annotated data. Solving a NLP task is usually involve annotating alot of data and then apply supervised machine learning approach. For example, if we interested in the part-of-speech (POS) tagging, we would imagine annotate each word in the sentence with the correct POS tag such as Noun, Verb, Adjective and then train a statistical classifier. In this approach, annotated data is crucial as it provides the only guidance for the model. Seeing the importance of annotated data, annotated resources have always been a part of most NLP conferences. Language Resources and Evaluation Conference (LREC) is the a major conference dedicated mainly for languages resources stressing their significance.

Clearly, clean annotated data is gold, however, it is expensive and slow to build since typically requires the careful design, testing, and subsequent refinement of

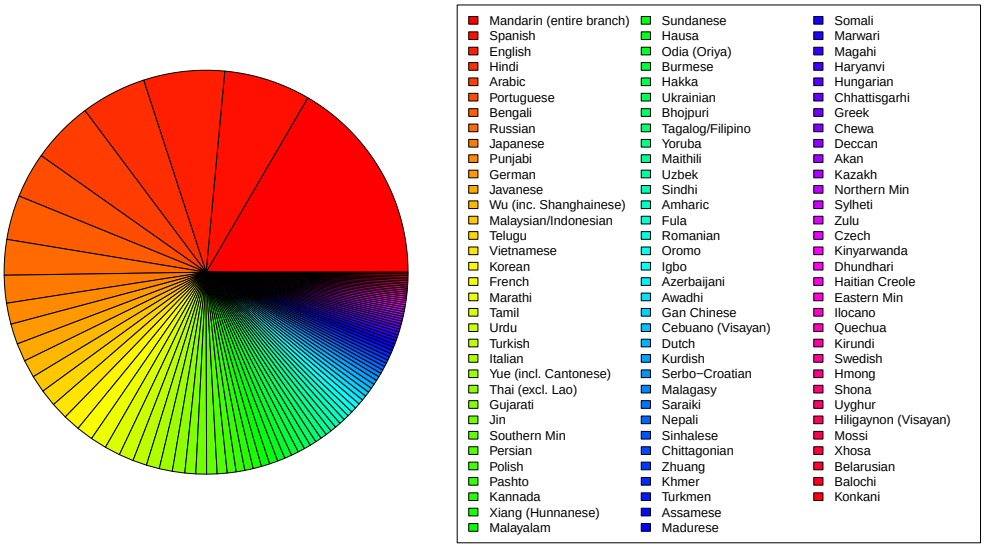


Figure 1.1: Fraction of world population (percentage) by number of native speaker in 2007. This diagram should be viewed in color. (source Wikipedia)

annotator guidelines, as well as assessment and management of annotator quality. For example, in the case of the Prague Dependency Treebank (PDT), it took a year to annotate the first 1000 sentences and 8 years to finish version 1 (Böhmová *et al.* 2001). Moreover, annotated data is usually task specific, meaning that it can not be reuse for different purposes. In this fast changing world, realize solely on annotated data is risky and not a very good strategy, remedy for this is one of the focus of this thesis.

Since it is expensive and hard to get, most annotated data is in resource-rich languages such as English, Mandarin and Portuguese. Doing NLP is much more challenging for so-called "resource-poor" languages for which there are limited available resources, particularly annotated data such as treebanks, wordnets, and the like. Standard supervised learning techniques require significant amounts of annotated data which is not suitable for resource-poor languages. There are approximately 7,000 languages in the world but of these only a small fraction (20 languages) are considered resource-rich (Baumann and Pierrehumbert 2014). Resource-poor languages are in dire need of a method to overcome the resource barrier, such that advances in NLP can be realised much more widely. Figure 1.1

shows proportion of the world language by native speaker. Despite the dearth of data, many languages are not uncommon and widely spoken such as Bengali, Punjabi, Javanese, Wu, Telugu, Vietnamese. Together the resource-poor languages show in Figure 1.1 are spoken by almost 2 billion people, roughly a third of the world's population.

Despite lack of annotated data, even for low-resource languages there are some unannotated data which we can exploit to learn more accurate model. With the growing quantity of text available online, and in particular, multilingual parallel texts from sources such as multilingual websites, government documents and large archives of human translations of books, news, and so forth, unannotated parallel data is becoming more widely available. This parallel data can be exploited to bridge languages, and in particular, transfer information from a resource-rich language to a resource-poor language. Knowledge bases such as dictionaries, wordnets and other lexical resources are another possible source of information, and exist in some form for many of the world's low-resource languages. The argument is that the the manually annotated data is hard to get yet dictionary is more widely available as part of the lexicon study. For instance, the Wiktionary project[1] uses crowd-sourcing to build dictionaries in many languages using the collaborative efforts of volunteers. In this way the dictionary grows in both size and language coverage over time. However, this resource is limited to only lexical items, and no deeper annotation. Panlex (Kamholz *et al.* 2014) is another example of multilingual dictionary that covers thousands of languages. Clues from related languages can also compensate for the lack of annotated data, as we expect there to be information shared between closely related languages in terms of the lexical items, morphology and syntactic structure. In this thesis, we investigate the method for effectively harness these additional resources aside from annotated data aiming for complementary effect.

Out of the 7,000 languages, most of them do not even have the writing system and many are dying. It is estimated that by the end of this century, half of the world

[1] wiktionary.org

languages will come to extinction as there are no speaker for that language (Crystal 2002). Since language captures the knowledge and wisdom. More attention is given for preserving the language before it lost forever. Bird *et al.* (2014b) pioneers on using speech technology to preserve the language using their android application called Aikuma which records the speech in the low-resource language and the translation in the higher resource language. They use speech translation as a way to preserve the language. However, it is unclear how to automatically process and learn from this collected data which also motivate a part of this thesis.

1.2 Research Questions

Extending existing NLP methods to cater for resource-poor languages is a highly active research area. For these languages, the conventional approach using supervised machine learning is inappropriate due to the lack of annotated data. Unsupervised approaches appear to be a better fit, however, despite considerable efforts, their performance lags well behind supervised approaches, and is rarely adequate. A more pragmatic and fruitful research direction which is attracting much attention, is to exploit different source of information aside from simple annotated text. Moreover, in the extreme case, it is also unclear how to process unwritten languages which are surprisingly common for the world's languages. Thus our research questions are:

- How can we achieve more accurate model for low-resource languages using less annotated data ? The assumption here is that annotated data is hard to get but other resources such as parallel data, monolingual text, bilingual dictionary are more widely available.

- What can we learn from unwritten languages? This question aim at solving the extreme case where we do not even have the writing system.

1.3 Scope

We aim at building the NLP framework for processing resource-poor languages. However, due to the complexity of a language, it is hard to say when a language is sufficiently processed. Even for high-resource language such as English, there is currently no framework to truly and completely understand English. However, there are well-established NLP tasks for processing a languages such as text summarization, part-of-speech tagging, correference resolution, machine translation, named entities recognition, optical character recognition, natural language understanding, parsing, sentiment analysis, speech recognition, speech segmentation, text-to-speech, word segmentation, word sense disambiguation, question answering, natural language generation. Each task has different objective and tackle very different problem however, they can be related to *syntax* and *semantics*. Some tasks such as part-of-speech tagging or parsing is purely about syntax, while word sense disambiguation is mainly about semantic. Nevertheless, most of tasks such as machine translation or sentiment analysis need some knowledge of both syntax and semantics. It appears that to process a language at least we need some basic tools to analyse the syntactic and semantic aspect of that language. The other distinguishing feature between NLP tasks is the *input format*. While many tasks process raw text, optical character recognition and speech processing take images and speech as the input. It is highly desirable that a framework to process a language must be able to handle multiple input formats. Moreover, as mentioned before, many languages do not even have the writing system, processing directly with the speech is the only way. Given the time limitation for the thesis with respect to the syntactic and semantics aspect, and the requirement for multi-modal inputs, we are going to focus on four most essential NLP tasks.

1. Part-of-speech (POS) tagging which tells us about the syntax categories of lexical item, classifying words into POS categories such as noun, verb, adjective.

2. Dependency parsing which shows the dependency relationship between words

in the sentence such as head/modifier, subject/verb.

3. Cross-lingual word embeddings which represent lexical items from multiple languages to the same dense vector space, preserving the monolingual and bilingual property of the language. These embeddings would the the bridge between resource rich and resource-poor languages allowing for transfer learning.

4. Speech to text translation which learn the alignment and translation between speech in a low-resource language and the translated text in the higher-resource language. This will be useful for task such as keyword spotting and also relevant for unwritten languages.

These tasks are very related and normally the latter are built based on the former. The reason for choosing these tasks is mainly because it appears in most of NLP pipelines and an advancement in NLP can not be realized without recourse to these tasks. We cover both syntactic (task 1 and 2) and semantic (task 3 and 4), and also attempt both text (task 1,2 and 3) and speech (task 4) representation of a language.

1.4 Contributions

The main contribution is the algorithm it self. We propose several algorithms motivated by machine learning approaches to effectively incorporate additional information to the model to further improve the performance.

POS tagging In the chapter about POS tagging, we proposed a semi-supervised method which effectively incorporate the noisy information from parallel data to the model as a prior. In this way, we demonstrate that only small amount of annotated data is sufficient for large improvements in performance. Compared with the state of the art, who also take advantage of parallel data, we make more realistic assumptions and use less parallel data, yet achieve a better overall result.

The second contribution is the novel tagset mapping algorithm. The corpora we employ make use of mappings from language-specific POS tag inventories to a common universal tagset (Petrov *et al.* 2012). However, such a mapping might not be good or even available for resource-poor languages. Therefore, we also propose a variant of our method capable of handling arbitrary tagsets based on a two-layer maximum entropy model. Evaluating on the resource-poor language Malagasy, we exceed the state-of-the-art by a large margin. This part is published as the long paper at EMNLP 2014 (§3.1.1).

Dependency Parsing In the chapter about dependency parsing, first we propose a semi-supervised learning based on parameter sharing in a neural network parser. The additional information we incorporate to the model is the language relatedness. We showed that we can achieve more accurate parser using the same training data by taking reference from related model in different languages in the cascade style. This work is published as a short paper to ACL 2015 (§3.1.2).

Latter we realize that we can do it better by jointly train the model instead of cascade approach. Our approach works by jointly training a neural network dependency parser to model the syntax in both a source and target language. In this way, the information can flow back and forth between languages, allowing for the learning of a compatible cross-lingual syntactic representation, while also allowing for the parsers to mutually correct one another's errors. Our experiments show that this outperforms a purely supervised setting, on both small and large data conditions, with a gain as high as 10% for small training sets. Our proposed joint training method also out-performs the cascade approach mentioned earlier. The other contribution concerns the learned word embeddings. We demonstrate that these encode meaningful syntactic phenomena, both in terms of the observable clusters and through a verb classification task. This part is published as a long paper at EMNLP 2015 (§3.1.3).

In the extreme case where there are not any available annotated data, we also propose an unsupervised dependency parser taking advantage of a novel syntactic word embeddings. Words from both source and target language are mapped to a

shared low-dimensional space based on their syntactic context, without recourse to parallel data. While prior work has struggled to efficiently incorporate word embedding information into the parsing model (Bansal *et al.* 2014; Andreas and Klein 2014; Chen *et al.* 2014a), we present a method for doing so using a neural network parser. When applied to the target language, we show consistent gains across all studied languages. Moreover, when multiple source languages are available, we can attempt to boost performance by choosing the best source language, or combining information from several source languages. To the best of our knowledge, no prior work has proposed a means for selecting the best source language given a target language. To address this, we introduce two metrics which outperform the baseline of always picking English as the source language. We also propose a method for combining all available source languages which leads to substantial improvement. This work has been published as long paper at ConLL 2015 (§3.1.4).

Crosslingual Word Embeddings Crosslingual word embeddings represent lexical items from different languages in the same vector space, enabling transfer of NLP tools. However, previous attempts had expensive resource requirements, difficulty incorporating monolingual data or were unable to handle polysemy. We address these drawbacks in our method which takes advantage of a high coverage dictionary in an Expectation-Maximization style training algorithm over monolingual corpora in two languages. Our model achieves state-of-the-art performance on bilingual lexicon induction task exceeding models using large bilingual corpora, and competitive results on the monolingual word similarity and crosslingual document classification task. We also evaluate several methods for combining embeddings which help in both crosslingual and monolingual evaluations. This part has been published as a long paper at EMNLP 2016 (§3.1.5).

We extend our work to cover more than two languages since most prior work on building crosslingual word embeddings focuses on a pair of languages. English is usually on one side, thanks to the wealth of available English resources. However, it is highly desirable to have a crosslingual word embeddings for many languages so that different relations can be exploited. We proposed a novel al-

gorithms for post-hoc combination of multiple bilingual word embeddings, applicable to any pre-trained bilingual model. We also extend our prior work to jointly learn multilingual word embeddings over monolingual corpora in several languages achieving uniformly excellent performance across various tasks. The last contribution is the investigation of the effectiveness of different source languages in transfer learning for cross-lingual document classification, plus incorporation of a new multilingual analogy dataset. This work has been submitted to EACL 2017 (§3.1.6).update if get accepted.

Unwritten language processing For many low-resource languages, spoken language resources are more likely to be annotated with translations than transcriptions. This bilingual speech data can be used for word-spotting, spoken document retrieval, and even for documentation of endangered languages. We experiment with the neural, attentional model applied to this data. On phone-to-word alignment and translation re-ranking tasks, we achieve large improvements relative to several baselines. On the more challenging speech-to-word alignment task, our model nearly matches GIZA++'s performance on gold transcriptions, but without recourse to transcriptions or to a lexicon. Our main contributions are: (i) proposing a new task, alignment of speech with text translations, including a dataset extending the Spanish Fisher and CALLHOME datasets; (ii) extending the neural, attentional model to outperform existing models at both alignment and translation reranking when working on source-language phones; and (iii) demonstrating the feasibility of alignment directly on source-language speech. This part has been published as a long paper at NAACL 2016 (§3.1.7).

All in all, our contributions are:

1. Show how to effectively incorporate different information (such as parallel data, or language relatedness) to the model aside from annotated data. In many case, this help not only resource-poor language but also resource-rich languages.

2. Analyse and tackle many real-world low-resource scenario such as tagset

mapping and limited resource.

3. Show the feasibility of learning meaningful relations directly from speech data.

4. Propose a new task of speech to text translation and several new datasets such as English-Serbian bilingual lexicon induction dataset, speech to text alignment corpus.

1.5 Thesis Overview

The backbone of the thesis is the set of publications through out the PhD candidature, attempt at answering all the research questions. Chapter 2 listed the background needed to understand the thesis. Chapter 3 summaries the research outcome through set of publications concerning with all four tasks including POS tagging, dependency parsing, crosslingual word embeddings and speech translation.

We will give the retrospective view for each publication and the analysis of the strong and weak points of each paper. Chapter 4 is conclusion and future work which revisit the list of research questions. In the appendix we will discuss other papers related to the thesis that I contributed to but shouldn't be counted toward my PhD.

Chapter 2

Background

In this chapter, we give the overview of low-resource natural language processing including definition, dataset, common techniques and a high level review of what people have done in this topic. We then give the background for four tasks that we will discuss on this thesis focusing on related work that we will compare with in the published papers.

2.1 Low-resource Natural Language Processing

2.1.1 What is a low-resource language?

Low-resource languages recently attracts much of attention, but we haven't given any concrete definition for low-resource languages. According to LORELEI, [1] low-resource language can be defined as a language that no automated human language technology exist. However, the term human language technology is vague. We base on this definition but develop further by picking an essential NLP task such as syntactic parsing as the yardstick. We instead can define a low-resource language as the language that does not have any syntactically annotated corpus which is essential to train the syntactic parser. Dependency treebank is popular

[1] Low Resource Language for Emergent Incidents (LORELEI) is a US government funded project aiming at developing human language technology for low-resource languages

among syntactically annotated corpora. Universal dependency treebank (Nivre *et al.* 2016) is the largest collection of dependency treebank in multiple languages currently covers 40 languages. Thus we can consider languages outside of those 40 languages, low- resource languages. However, with this definition, it is arguable that can we use one syntactic task such as dependency parsing to represent language technology. Moreover, some languages (such as Buryat, Coptic, Kazakh, Sanskrit or Tamil) in those 40 languages have very modest size (less than 1000 annotated sentences). Dependency parsers trained on those treebanks would, expectedly, achieve modest performance. On other end, Berment (2004) proposed a long list of basic language resource kit for measuring language resources taking into consideration the minimum set of corpora, tools and human resource. They define a language is low-resource if the weighted score is less than 10 out of 20 points. However, as expected, this definition is also heuristic as criticized by Prys () by lack of consideration for raw material such as newspapers. We take the middel ground approach by simplify the definition of Berment (2004). We instead defining low-resource language taken into account the task.

> *A language is considered low-resource for a given task if there is no algorithm using currently available data to automatically solve the it.*

This definition implies that a language is consider low-resource based on the task specific. For example, Spanish is not low resource language with respect to the part-of-speech tagging task with a decent [2] performance. However, it is resource poor language for sentiment analysis task since there are not any annotated data for this task in Spanish. Different domain or genre inside a language can also be considered low-resource language. Take POS tagging task as an example, the annotated corpus is mainly constructed for news wire domain, the accuracy of English tagger on this domain can be as high as 97% (Toutanova *et al.* 2003). However, for historical English domain we achieve much lower accuracy (Yang and Eisenstein 2016). In this way, historical English domain becomes low- resource language given POS tagging task. With this definition, for a given task requirement, many

[2] more than 90% accuracy

languages becomes low- resource regardless of the number of speakers and the popularity of that language, which will be the subject of this thesis.

Moreover, it should be noted that a language or domain is low-resource today but might not be in the future. English Twitter text is an example. It was resource-poor language 5 years ago without any tool to process. However, with the high demand on social data analysis, alot of research are poured into building resources and models to effectively normalized the text, POS tagging, dependency parsing and sentiment analysis (Han and Baldwin 2011; Gimpel *et al.* 2011; Kong *et al.* 2014; Agarwal *et al.* 2011) which makes English Twitter text no longer low-resource languages for those tasks. Our thesis is all about improving the performance to meet the expectation, leveraging the low-resource scenario. However, instead of looking at each domain (e.g. Twitter) or language, we want to investigate on the algorithm part such that it can widely be applied to many low-resource languages.

2.1.2 What resources can we expect?

For a low-resource language, we can not expect large annotated corpus. However, we might be able to get some small annotated data, monolingual corpus, parallel corpus or dictionary. This section speculate the type of resource we can reasonably expect in the real world low-resource scenario.

Field linguist annotation

Half of the world 7000 languages are unwritten languages (Lewis 2009). It is the extreme case where the resource for those languages are usually come from field linguist annotation under language preservation and documentation projects. There can be many outputs from field linguist analysing some aspect of the language such that lexicon, morphology, phonology, dictionary. However, it is usually unsuitable for automated natural language processing method because of the tiny size. Recently, Bird *et al.* (2014b) proposed an mobile application called Aikuma enabling much faster and easier way for collaborative language documentation. The output from Aikuma is the parallel speech between the source low-resource

and the target higher-resource languages with the options of re-speaking for higher quality record and some transcription in the target language. For the initial experiment, Bird *et al.* (2014a) managed to collect around 10 hours of speech from indigenous communities in Brazil and Nepal. Blachon *et al.* (2016) used an extended version of Aikuma to collect more than 80 hours of speech from Congo-Brazzaville. Thus, for the unwritten language, using Aikuma probably we can expect order of 100 hour of parallel speech.

Monolingual Corpora

For the other half of the 7000 languages, we have some writing system. To cheaply collect the examples of a language (monolingual data), World Wide Web is probably the best option. The Crúbadán project (Scannell 2007) is an attempt to crawl monolingual data for resource-poor languages, to date they managed to support more than 2000 languages[3]. Wikipedia is another major source for monolingual data contributed by volunteers, covering for more than 200 languages. Leipzig Corpora Collection (LCC) (Goldhahn *et al.* 2012) is another project for collecting monolingual data which currently covers more than 200 languages, crawled from the web. However, we can expect very different monolingual data size for each language. For example, from Wikipedia, to date, 58 languages have more than 100k articles and 132 languages have more than 10k articles[4]. Goldhahn *et al.* (2012) shows that in LCC corpus at 2012, around 50 languages have 1 millions sentences and 100 languages have 70k sentences. That is why when working with languages in the top 50, probably we can expect the order of millions sentences or (10 million words) only.

Comparable and Bilingual Corpora

Monolingual data only show the relationship between lexical items inside a language. Comparable or bilingual corpora, on the other hand, can relate languages

[3]crubadan.org (accessed 14/09/2016)
[4]en.wikipedia.org/wiki/List˙of˙Wikipedias (accessed 14/09/2016)

together. As the development of multilingual news, books, subtitles, government websites, it is easier to get the comparable corpora for many low-resource languages. They contains set of documents in multiple languages that is topically "comparable". Wikipedia is a source for comparable data since one topic (e.g. Barack Obama) is usually written in several languages. Multilingual online news service such as BBC[5] is another source for comparable data, available in 32 languages. STRAND (Resnik and Smith 2003) is the system to crawl comparable documents from the web based on the structure of the website. Their proposed system can be used to crawl comparable corpora for any language pair. However, as admitted, they can not find many language pairs and modest in size.

Bilingual corpora is usually extracted from comparable corpora, representing sentence aligned translations. Bilingual corpora is particularly of interest since it is the main input for machine translation and can be used as the bridge between languages. Europarl (Koehn 2005) is a popular bilingual corpus cover many European languages as legal documents and policies are needed to translate to the language of all participating countries. Opus (Tiedemann 2012) is an open-access platform for retrieving bilingual corpora, data is manually collected from many open sources such as movie subtitle, bible, European documents. Opus probably is the largest collection of freely available parallel corpora, covers more than 90 languages. Each pair in the top 100 language pairs in Opus has more than 100 million words. This is a decent size even for resource intensive NLP. However, most of the top 100 language pairs are from well-supported languages with some exceptions such as Romanian-Turkish and Bulgarian-Hungarian. That is why for low-resource languages, we can not expect that much of parallel data which is usually overlook by previous work. Many previous approaches for low-resource NLP rely on parallel data as the bridge to transfer the annotation from source resource-rich language to the target resource-poor language (Das and Petrov 2011; Duong *et al.* 2013b). However, due to lack of the evaluation for real low-resource language, they usually evaluate on well-supported language where annotated data are avail-

[5]bbc.com (accessed 14/09/2016)

able. Consequently, the parallel data is much easier to get for those languages.

How much parallel data can we reasonably expect for a low-resource language? It is hard to estimate and will be different according to language. However, if we are working with top 15 languages (≈ 100 pairs) which mainly are European languages, probably we can get a decent parallel size in order of million sentence pair. Nevertheless, in our EMNLP 2014 (§3.1.1) paper, we experimented with two low-resource languages Malagasy and Kinyawanda. The biggest bilingual corpora we can find are in order of 100k and 10k sentences respectively.

Bilingual Dictionary

Bilingual dictionary is a common resource for a low-resource language. This is usually the output of linguist when one language is studied. Bilingual dictionary contains the word translation of a low-resource language to the more common language such as English. There are some notably corpus that collect translation for multiple languages. Panlex (Kamholz *et al.* 2014), a dictionary which currently covers around 1300 language varieties with about 12 million expressions[6]. This dataset is growing and aims at covering all languages in the world and up to 350 million expressions. The translations in PanLex come from various sources such as glossaries, dictionaries, automatic inference from other languages, etc. Accordingly, Panlex has high language coverage but often noisy translations. Wiktionary [7] is another notable source for bilingual dictionary. A part of Wiktionary is also manually extracted from dictionaries and glossaries but it is also powered by volunteers. Currently, Wiktionary covers over 2500 languages[8] with 4.5 millions expressions. Bilingual dictionary is very useful and can be used as the bridge between low-resource and higher-resource languages through translations. Aside from translations, some entries in Wiktionary and Panlex also contains additional information such as part-of-speech, pronunciation and sample sentences which can be very useful to process the low-resource languages. However, what is the reasonable size

[6] an expression is usually a word in a language.
[7] en.wiktionary.org (accessed 19/09/2016)
[8] However, more than 2000 languages have tiny (less than 100) expressions.

Min # Expressions	Wiktionary	Panlex
2,000	105	369
20,000	34	87
200,000	9	23

Table 2.1: Number of languages having more than minimum number of expression in Wiktionary and Panlex. The number for Panlex is from Kamholz *et al.* (2014).

for a dictionary concerning low-resource languages? Table 2.1 shows the number of languages in Wiktionary and Panlex having minimum number of expressions which gives the idea of what we can expect for low-resource languages.

Small annotated data

Some languages have small annotated data for the task of interest. This is usually the result of the field linguists when studying a language. The size of these corpora is often small and inadequate for a decent supervised learning. However, as shown in our EMNLP 2014 (§3.1.1), ACL 2015 (§3.1.2) and EMNLP 2015 (§3.1.3) papers, with a careful design and training, even a small annotated corpora can help immensely. Again, the question of how much annotated data we can expect varies depend on task and languages. For example, for part-of-speech (POS) tagging task, Garrette and Baldridge (2013) reported a corpus of $\approx$ 10k words for Kinyawanda and Malagasy as the result of 4 hour annotation. As for dependency parsing task, the smallest corpora (in words) in the universal dependency treebanks are Sankrit (1k), Kazakh (4k), Coptic (4k), Buryat (5k), Tamil (8k) which represent the low-resource scenario. Probably, for low-resource languages we can not expect the annotated corpora with more than 10k annotated words.

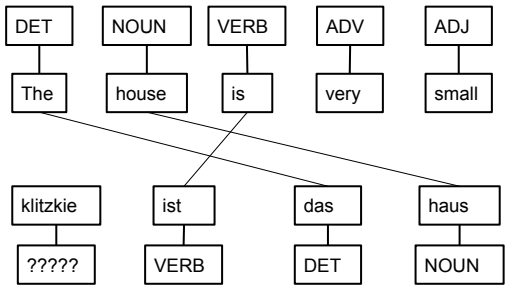


Figure 2.1: Examples of part-of-speech projection from English to German using parallel text. No tag is given to the German word *klitzkie* that is not aligned.

2.1.3 Transfer learning

Transfer learning is a common techniques when working with low-resource languages (Täckström *et al.* 2013; Das and Petrov 2011; Yarowsky and Ngai 2001; Duong *et al.* 2013a; Hwa *et al.* 2005; Ma and Xia 2014). The annotation information is transferred from resource rich-language to the resource-poor language. In fact, most of our work in this thesis motivated by transfer learning, covering almost all our publications except for NAACL 2016 (§3.1.7).

Annotation transfer

There are many transferable things between source and target languages. The most common form is the annotation. Since the annotated data is more common in the source resource-rich language, it is transferred to the target language through bilingual resource such as bitext. Figure 2.1 shows an example of part-of-speech annotation projection from English to German through alignments. There are several successful application of this approach to low-resource part-of-speech tagging and noun-phrase chunking (Yarowsky and Ngai 2001), dependency parsing (Hwa *et al.* 2005), named entity recognition (Wang *et al.* 2013). The challenge to this approach is that (1) the alignment is not always accurate, (2) not all tokens in the target language got the annotation (e.g. German word *klitzkie*) and (3) the projected annotation is not always linguistically correct in the target language. For example, in Malagasy all number are considered *ADJ* (adjective) which is wrongly

assigned as *NUM* (number) if projected from English. This is the reason why the projected annotation is usually post-processed, mostly rule based (Hwa *et al.* 2005) or converted to soft-constrains (Das and Petrov 2011; Täckström *et al.* 2013) before feeding to the machine learning algorithm.

Model transfer

The pipeline for annotation transfer is normally involve (1) the supervised model is trained on the source resource rich language, (2) use this model to provide annotation for the source language from bilingual data (3) project the annotation to the target language (4) use this projected annotation for target language model. Each step might introduce some noise, consequently the final target language model might be very biased. Therefore, instead of transferring the annotation, we can also transfer the model directly from the source resource rich languages to the target resource poor language (Zeman *et al.* 2008; Ma and Xia 2014). This is also the approach we took for many published work from this thesis including EMNLP 2014 (§3.1.1), EMNLP 2015 (§3.1.3), ACL 2015 (§3.1.2) and ConLL 2015 (§3.1.4).

The model can only be transferred when both source and target features are in the same space. That is why the features that can be shared in both languages are desirable. Universal part-of- speech tagset (Petrov *et al.* 2012) is an attempt to map any language specific tagset to the same universal tagset. Universal dependency treebank (Nivre *et al.* 2016) is another attempt to map the annotation from different treebank to the same universal annotation. World Atlas of Language Structures (Dryer and Haspelmath 2013) which indicate structural properties of languages such as English has SVO structure or Japanese has SOV, can also be shared across languages. Naseem *et al.* (2012) and Täckström *et al.* (2013) use these features for transfering dependency parser. Täckström *et al.* (2012) induced crosslingual word cluster where lexical items in both languages are grouped together using parallel data, also applied for transferring dependency parser. In this thesis, we investigate on crosslingual word embeddings where lexicon in sev-

eral languages are represented as dense vector in the same semantic space, enable transfer learning. The crosslingual word embeddings must capture well the monolingual and bilingual relations in the semantic and syntactic space. We have successfully built and apply crosslingual word embeddings for several tasks in our EMNLP 2016 (§3.1.5) and EACL 2017 (§3.1.6) remove if not accepted papers.

The transferred model from the source language is normally inadequate for the target language and usually need refinement (Zeman *et al.* 2008). Ma and Xia (2014) added the constrains from parallel data to the transfered model. McDonald *et al.* (2011) additionally exploit multiple source languages. We, on the other hand, take advantage of a small annotated corpora. We show that we can correct much of the transferred model with the guideline from small annotated data.

2.1.4 Notable work

Table 2.2 list notable published work on low-resource natural language processing covering some tasks related to speech, part-of-speech tagging, dependency parsing and named entity recognition with the data assumption. This is by no mean an exhaustive list but give some idea of what people have done for low-resource natural language processing and their resource assumption. Some prior work use cheap resource such as monolingual data or unlabelled speech. However, many of them exploit parallel corpus which is harder to get for many low-resource languages, limiting their applicability. Most of the paper listed in Table 2.2 will be covered in more detail in subsequent sections.

2.2 POS tagging

We will work with four main NLP tasks for low-resource languages in our thesis. They are POS tagging, dependency parsing, crosslingual word embedding and unwritten language processing. In this section we focus on the first task – POS tagging which is the task of assigning morphological categories i.e. *Noun*, *Verb*, *Adjective* etc to the lexical items. Moreover, POS tagging in useful in itself as

Paper	Topic	Resource
Kamper *et al.* (2015b)	speech lexicon discovery	unlabelled speech
Kamper *et al.* (2016b)	speech lexicon discovery	unlabelled speech
Besacier *et al.* (2014)	speech recognition	speech + transcription
Khanagha *et al.* (2014)	speech segmentation	unlabelled speech
Gelling *et al.* (2012)	dependency parsing and POS tagging	monolingual corpus
Sun *et al.* (2014)	dependency parsing	small annotated corpus
Xia and Lewis (2007)	dependency parsing	interlinear grossed text
Georgi *et al.* (2013)	dependency parsing	small annotated corpus + interlinear grossed text
Zeman *et al.* (2008)	dependency parsing	source language annotation
Täckström *et al.* (2013)	dependency parsing	source language annotations
Zhang and Barzilay (2015)	dependency parsing	source language annotation
Naseem *et al.* (2012)	dependency parsing	source language annotation
McDonald *et al.* (2011)	dependency parsing	parallel corpus
Ganchev *et al.* (2009)	dependency parsing	parallel corpus
Hwa *et al.* (2005)	dependency parsing	parallel corpus
Ma and Xia (2014)	dependency parsing	parallel corpus
Yarowsky and Ngai (2001)	POS tagging	parallel corpus
Duong *et al.* (2013b)	POS tagging	parallel corpus
Das and Petrov (2011)	POS tagging	parallel corpus
Täckström *et al.* (2013)	POS tagging	parallel corpus + POS dictionary
Li *et al.* (2012)	POS tagging	POS dictionary
Garrette and Baldridge (2013)	POS tagging	2 hour annotation
Wang and Manning (2013)	Named Entity Recognition	parallel corpus
Darwish (2013)	Named Entity Recognition	parallel corpus + Wikipedia links
Nothman *et al.* (2013)	Named Entity Recognition	Wikipedia links
Tsai *et al.* (2016)	Named Entity Recognition	Wikipedia links

Table 2.2: Notable related work on low-resource natural language processing.

an important step in many NLP pipelines, informing deeper layers of annotation, helping to understand the syntactic aspect of the language. We now briefly review prior approaches proposed for POS tagging for resource-poor languages, focusing on their supervision requirements. In our EMNLP 2014 (§3.1.1) paper, we present our own semi- supervised learning approach which we argue has more realistic data requirements befitting the resource-poor scenario.

Supervised Learning

The traditional approach to POS tagging builds a separate tagger for each target language, usually based on supervised machine learning algorithms (Brants 2000; Brill 1995; Toutanova *et al.* 2003). Supervised learning needs manually annotated data which is time consuming and costly to construct. If we were to apply supervised learning to a resource-poor language the first question we have to consider is the amount of annotated data needed. This is a hard question to answer in general, due to the lexical and syntactic properties of the language, as well as the cost of manual annotation.

Moreover, corpus annotation is time consuming and costly. For example, for the POS layer of the Penn Treebank (Marcus *et al.* 1993) it took 3 years to annotate 4.5 million tokens. We cannot expect anywhere near as a large annotated corpora for resource-poor languages. However, Garrette *et al.* (2013) show that POS annotations for 1,000 tokens are easy to acquire with around 1 hour of manual effort. This raises the challenge of how we can best make use of such tiny amounts of annotated data.

Unsupervised learning

Unsupervised approach is typically suitable for resource-poor language since it doesn't need any manually annotated data and unlabelled data is relatively easy to acquire. These approaches try to group words having the same morphological/syntactic properties into the same group (cluster) (Christodoulopoulos *et al.* 2010; Biemann 2006b; Biemann 2006a). It is believed that words in the

same cluster are likely to have the same POS tag. One problem with this approach is determining the number of clusters. Defining that number beforehand might not be a good solution (Biemann 2006b). We might force the algorithm to separate coherent clusters or to join unrelated ones. On the other hand, letting the algorithm choose when to stop could result in a too specific or too general clusters. Evaluation is also another major consideration, since we don't have the gold data to compare with. However, the biggest problems introduced by unsupervised approach is the poor performance (Christodoulopoulos *et al.* 2010; Blunsom and Cohn 2011) hinder its usage in real world applications.

Semi-supervised learning

As mentioned above, supervised learning needs large training corpora, which are only available for resource-rich languages. Unsupervised POS tagging, on the other hand, is suitable for resource-poor languages since requires only unannotated text, however their relatively poor performance is not suitable for practical applications. Semi-supervised learning appears to better fit which is also the approach in our EMNLP 2014 (§3.1.1) paper. We show that we can achieve high performance POS tagger exploiting only tiny amount of annotated data and some distance supervision from additional resources. It is important to understand what kind of supervision signal we might have from additional resources which will be reviewed in the following.

2.2.1 Typologically related information

For closely related languages, such dialects of the same language or those in the same language family, the lexicon and syntactic structures of the languages are likely to be highly similar. These kinds of similarities can be exploited when developing tagging models for low-resource languages (Hana *et al.* 2004; Feldman *et al.* 2006; Reddy and Sharoff 2011). They propose tying together the transition probabilities and estimating the emission probability separately either by mimicking the source language lexicon or with supervised learning from a small amount of anno-

tated data. Note that this method does not need parallel data, as no alignments are required, however monolingual annotated data is required for related languages, which is unlikely to be available for many low-resource languages.[9]

2.2.2 Projected information

Yarowsky and Ngai (2001) pioneered the use of parallel data for projecting tag information from a resource-rich language to a resource-poor language. They first tag the source resource-rich language using a supervised POS tagger, and the tagging is then projected to the target resource-poor language through a word alignment. They observed that although this works well in many cases, the projected tags are very noisy. Thus, they apply a heuristic based on sentence alignment score to filter out noisy alignments. Finally, the projected tags are used to build the target language tagger, which can then be applied to other texts. Duong *et al.* (2013b) used a similar method on using sentence alignment scores to rank the goodness of sentences. They trained a seed model from a small part of the projected data, then applied this model to the rest of the data using self-training with revision.

Das and Petrov (2011) also used parallel data but additionally exploited graph-based label propagation to expand the coverage of labelled tokens. Each node in the graph represents a trigram in the target language. Each edge connects two nodes which have similar context. Originally, only some nodes received a label from direct label projection, and then labels were propagated to the rest of the graph. Rather than use the labels directly, Das and Petrov (2011) instead use the labels to extract a tag dictionary which is used as constraints in learning a feature-based HMM (Berg-Kirkpatrick *et al.* 2010). Both Duong *et al.* (2013b) and Das and Petrov (2011) achieved 83.4% accuracy on the test set of 8 European languages (Table 2.3).

[9]Especially for languages only spoken by small communities, in which case the best we might hope for is parallel data between the target language and a mainstream 'contact' language, such as English or a pidgin.

	da	nl	de	el	it	pt	es	sv	Average
Duong *et al.* (2013b)	85.6	84.0	85.4	80.4	81.4	86.3	83.3	81.0	83.4
Das and Petrov (2011)	83.2	79.5	82.8	82.5	86.8	87.9	84.2	80.5	83.4
Li *et al.* (2012)	83.3	86.3	85.4	79.2	86.5	84.5	86.4	86.1	84.8
Täckström *et al.* (2013)	88.2	85.9	90.5	89.5	89.3	91.0	87.1	88.9	88.8

Table 2.3: Previously published token-level POS tagging accuracy for various models across 8 languages: Danish (da), Dutch (nl), German (ge), Greek (el), Italian (it), Portuguese (pt), Spanish (es), Swedish (sv) evaluated on CoNLL data.

2.2.3 Dictionary Information

A tag dictionary specifies the set of allowable tags for a word. Even an incomplete or noisy tag dictionary is sufficient to allow for a POS tagger to be learned using standard unsupervised inference, such as the Expectation Maximization (EM) algorithm, where the entries in the tag dictionary are used to constrain the tags for each word (Kupiec 1992; Merialdo 1994; Banko and Moore 2004; Goldberg *et al.* 2008). The usefulness of tag dictionaries is due to many words having very few possible tags and thus the tag dictionary drastically restricts the search space, while also steering EM away from poor local optima. With a dictionary derived from gold-standard data Das and Petrov (2011) achieved an accuracy of approximately 94% on the same 8 languages. The effectiveness of a gold-standard dictionary is undeniable, however it is costly to build one, especially for resource-poor languages. Cheaper crowd-sourced dictionaries are also valuable, as demonstrated by Li *et al.* (2012) used Wiktionary[10] to achieve 84.8% accuracy on the same 8 languages (see Table 2.3). Note, however, that there are large differences in the performance for words appearing in dictionary and out-of-vocabulary (OOV) words (89% vs 63%), which suggests that their approach will be of much less use for small and incomplete POS dictionaries.

[10] wiktionary.org

Täckström *et al.* (2013) combined both token information from bilingual projection and type constraints from Wiktionary to achieve the current state-of-the-art in low-resource tagging. Their approach first builds a tag lattice, which is then pruned using the token information and type constraints. The remaining paths are used to train a Conditional Random Field (CRF) tagger. They achieved 88.8% accuracy on the same 8 languages (see Table 2.3). In our EMNLP 2014 (§3.1.1) paper, we will mainly compare the results of our approach with Täckström *et al.* (2013). Note that our method and theirs have very different data requirements: we use a small corpus of annotated part-of-speech in the target language, but only limited parallel data and no tag dictionaries, while they use orders of magnitude more parallel data as well as implicit supervision courtesy of their tag dictionary. As argued above, while both approaches have limited supervision, our data requirements are more appropriate to a low-resource scenario.

Table 2.3 summarises the performance of the above models across all 8 languages. Note that these methods vary in their reliance on external resources. The systems listed in Table 2.3 are sorted in the ascending order of resource usage. Duong *et al.* (2013b) use the least, i.e. only the Europarl Corpus (Koehn 2005). Das and Petrov (2011) additionally use the United Nation Parallel Corpus. Li *et al.* (2012) did not use any parallel text but used Wiktionary. Täckström *et al.* (2013) exploited most parallel data by additionally using parallel data crawled from web, as well as using the tag dictionary from Li *et al.* (2012). The pattern of results in Table 2.3 illustrates the common lesson in NLP: when adding additional resources, the models perform better.

2.2.4 Small Annotated Data Information

An alternative approach for tagging resource-poor languages is to assume a small corpus of manually annotated data. Garrette *et al.* (2013) built a POS tagger for two resource-poor languages, Kinyarwanda (`Kin`) and Malagasy (`Mlg`). They used no parallel data, but instead exploited four hours of manual annotation to label 4,000 tokens or 3,000 word-types. These tokens or word-types were used to

Lang	Size (k)	# Tags	Not Matched
da	94	8	DET, PRT, PUNC, NUM
nl	203	11	PRT
de	712	12	
el	70	12	
it	76	11	PRT
pt	207	11	PRT
es	89	11	PRT
sv	191	11	DET
kin	9.3	9	PRT, PRON, NUM
mlg	9.5	11	NUM

Table 2.4: The size of annotated data, number of tags included and missing for the 8 European languages: Danish (da), Dutch (nl), German (de), Greek (el), Italian (it), Portuguese (pt), Spanish (es) Swedish (sv) and 2 resource-poor languages Kinyarwanda (`Kin`) and Malagasy (`Mlg`).

build a tag dictionary. They employed label propagation to expand the coverage of this dictionary, much like Das and Petrov (2011). The dictionary was used to label training examples, from which they learned a tagger. This achieved 81.9% and 81.2% accuracy for `Kin` and `Mlg` respectively.

The method we propose in our EMNLP 2014 (§3.1.1) paper is similar in that we also use a small amount of annotation. However, we directly use the annotated data to train the model rather than indirectly via a tag dictionary. We argue that with a proper "guide", namely parallel projection, we can take advantage of very limited annotated data. Furthermore, our approach is also able to use a dictionary, although even without this form of supervision our method results in high accuracy taggers, well above baseline approaches and in most cases outperforming the previous state-of-the-art.

2.2.5 Universal tagset

Core to many of the projection methods described above is an assumption of a matching tagset between the source and target languages. That way the labels projected from the source have meaning in the target language, and can be used directly as target labels, constraints, etc. It is uncommon for languages to have been annotated with same tag set, for this reason these approaches use the universal POS tagset (Petrov *et al.* 2012). This tagset consists of a list of tags that are said to be shared across languages, as well as mappings into this scheme from native tagsets in several languages. The universal tagset is extremely useful in multilingual applications, enabling joint multi-lingual modelling as well as simpler evaluation of results across languages. In our setting, using the universal tagset can simplify our problem, removing the difficult issue of matching between different tagsets. For low-resource languages without an official tagset, such as Bengali or Lahnda, the universal tagset would be a good starting point for linguistic annotation.

The universal tagset from Petrov *et al.* (2012) consists of 12 common tags: *NOUN, VERB, ADJ* (adjective), *ADV* (adverb), *PRON* (pronoun), *DET* (determiner and article), *ADP* (preposition and post-position), *NUM* (numerical), *CONJ* (conjunctions), *PRT* (particle), *PUNC* (punctuation) and *X* (all other categories including foreign words and abbreviations). Petrov *et al.* (2012) provide the mapping from several language-specific tagsets to the universal tagset.

Nevertheless, using universal tagset looses information, such as tense and case information is often lost in the mapping. For example, the Penn treebank tags verbal tags *VB, VBD, VBG, VBN, VBP, VBZ* are mapped to the generic *VERB* tag in the Universal tagset. Moreover, the mapping is not always straightforward. Table 2.4 shows the size of the annotated data for each language, the number of tags presented in the data, and the list of tags that are not matched. We can see that only 8 tags are presented in the annotated data for Danish, i.e, 4 tags (*DET, PRT, PUNC,* and *NUM*) are missing.[11] Thus, a classifier using all 12 tags will be heavily

[11] Many of these are mistakes in the mapping, however, they are indicative of the kinds of issues expected in low-resource languages.

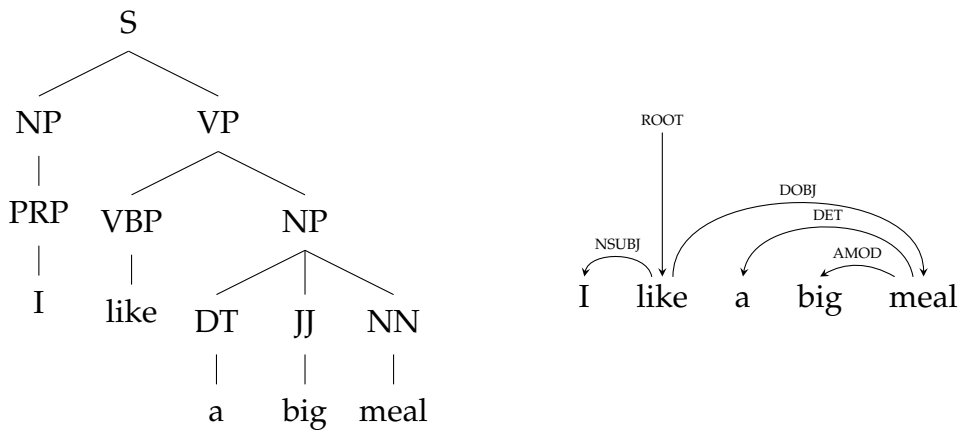


Figure 2.2: Phrase structure tree (Left) and Dependency tree (Right) of the same sentence.

penalized in the evaluation.

Li *et al.* (2012) considered this problem and tried to manually modify the Danish mappings i.e. map tag *AC* and *AO* as *NUM* or match tag *U* to *PRT* etc. Moreover, *PRT* is not really a universal tag since it only appears in 3 out of the 8 languages. Plank *et al.* (2014) state that *PRT* often gets confused with *ADP* even in English. We will later show that the mapping problem causes substantial degradation in the performance of a POS tagger exploiting parallel data. The method we present in our EMNLP 2014 (§3.1.1) paper is more target-language oriented: our model is trained on the target language, in this way, only relevant information from the source language is retained. Thus, we automatically correct the mapping, and other incompatibilities arising from incorrect alignments and syntactic divergence between the source and target languages.

2.3 Dependency Parsing

While part-of-speech tagging operates in the word level, provides information about syntactic category of each separate word in the sentence, we move to the sentence level in the second task which is parsing. Sentence parsing is the task of understanding the underlying structure of sentence. There are two main

structure representations: (1) phrase structure tree (2) dependency tree. Phrase-structure tree represents nesting structure of phrases such as noun phrase, verb phrase, preposition phrase etc. Dependency tree, on the other hand, shows the dependencies between words. For example, the sentence "*I like a big meal*" has two representations as shown in Figure 2.2.

Phrase structure tree is more meaningful for understanding grammatical (syntactic) structure of the sentence. However, for each language, the phrase structure might be very different. For example, English favours *"Subject Verb Object"* structure while Japanese switch the position of *Object* and *Verb* (always puts *Verb* at the end of the sentence). Thus, when copying information from the source language to the target language, phrase structure tree is not particularly suitable. Dependency tree, in contrast, shows the semantic structure i.e. answering question such as *who did what to whom by which means?*. Thus, dependency structure will be more transparent across languages. In addition, dependency tree are better at capturing long distance relations which is desirable in many applications. We are going to use this structure for building parser for target resource-poor languages.

Dependency tree is usually formalized as labelled directed graph G = (V,A) where V is the set of nodes, A is the set of arcs. For example, the dependency tree in Figure 2.2 has

$$V = \{I, like, a, big, meal\} \tag{2.1}$$

$$A = \{(like, nsubj, I), (like, dobj, meal), (meal, det, a), (meal, amod, big)\} \tag{2.2}$$

A is the set of (w_i, r, w_j) which represent the relation r from the head w_i to dependent w_j.Most of the time the roles of head and dependence are very distinguishable. However, sometime it is hard to distinguish, especially when it involves articles, complementizers and auxiliary verbs etc. For example, in the sentence *I give up my thesis*, it's unclear whether $give$ is the head of up or vice versa. Due to all these uncertainty, dependency parsing is a much harder task compared with POS tagging especially in the low-resource scenario. The rest of this section is organized as follow. In section 2.3.1, we are going to review some supervised methods to build a dependency parser. In section 2.3.2, we reviewed crosslingual methods

applied to resource-poor languages.

2.3.1 Supervised dependency parsing

Grammar-based approach

Phrase structure tree has a long history. Many algorithms are developed to parse phrase structure tree. Naturally, people want to apply the phrase structure approaches to dependency parsing. One of the notable approach is using context free dependency grammar in similar vein to context free grammar of phrase structure parsing. However, in context free dependency grammar, all non-terminal nodes (e.g. *S, NP, VP*) are replaced with an actual word. This is the simplest way to convert from phrase structure grammar to dependency grammar. Nivre (2002); Eisner and Blatz (2007); Johnson (2007) proposed more complicated but also more efficient method for conversion. After the dependency grammar is constructed, we can directly use any phrase structure parsing algorithm such as CKY (Younger 1967). One disadvantage of using dependency grammar is that it's very hard to capture long distance relations or apply to non-projective parsing.

Another adaptation for dependency parsing use the tree conversion rules. That is, the sentence is parsed using phrase structure. Phrases are converted to dependency relation using head-rules (de Marneffe *et al.* 2006; Yamada and Matsumoto 2003). However, these rules are language specific and normally need a lot of expertise to build.

Transition-based approach

Transition based parsing is more recently developed (Nivre 2008). It is similar to finite state automaton which consists of set of *configurations* and *transitions*. The parsing algorithm will choose a list of transitions that transform the initial configuration to the terminal configuration.

Configuration Given a sentence $w = w_0, w_1, ...w_n$, with w_0 is the dummy $ROOT$, a configuration is defined as a triple $c = (S, Q, A)$ where

- S is the stack of partially proceeded words.
- Q is the queue of remaining words
- A is the set of arcs that form partially parsed tree.

Each configuration c aims at capturing a partial analysis of a sentence. The initial configuration for the above sentence w is defined as

$$c_{init} = (S_0, Q_0, A_0)$$

Where $S_0 = [w_0]$ contains only the dummy $ROOT$. $Q_0 = [w_1, w_2, ...w_n]$ contains all the remaining words and $A_0 = [\texttt{Empty}]$. The terminal configuration is defined as

$$c_{terminal} = (S_{ter}, Q_{ter}, A_{ter})$$

Where $Q_{ter} = [\texttt{Empty}]$ for any S_{ter} and A_{ter}. That is, the algorithm terminates when there isn't any word left needed to proceed in the queue regardless of S_{ter} and A_{ter}.

Transition The set of transitions are used to transform the initial configuration c_{init} to the terminal configuration $c_{terminal}$. Basically, there are 3 transitions.

- `left-arc`(r) : $(S|w_i,\ w_j|Q,\ A) => (S,\ w_j|Q,\ A \cup \{(w_j, r, w_i)\})$
- `right-arc`(r) : $(S|w_i,\ w_j|Q,\ A) => (S,\ w_i|Q,\ A \cup \{(w_i, r, w_j)\})$
- `shift` : $(S,\ w_i|Q,\ A) => (S|w_i,\ Q,\ A)$

The `left-arc`(r) for dependency relation r with w_i at the top of stack S and w_j at the first position of queue Q, add a dependency arc (w_j, r, w_i) to A and pop the stack. Pre-condition for `left-arc` is that both stack and queue are non-empty and $i \neq 0$. The `right-arc`(r) for dependency relation r adds the dependency (w_i, r, w_j) to A but pop the stack and replace the first element of queue with w_i. The precondition is both stack and queue are non-empty. `shift` simply remove the first word of queue and put at top of stack. The precondition is that buffer is non-empty.

Parsing Algorithm The parsing algorithm decides what transition is applied to a given configuration. Nivre (2008) formalized it as a supervised classification task. The dependency treebank is converted to the training data using some heuristic rules.[12] The classifier is trained on this data set. Crucially, the classifier should give confidence/ranking for each prediction. Since if the first prediction is not applicable (i.e. the pre-condition is not satisfied), the second one will be considered etc. The algorithm always terminate in $O(n)$ steps. Each `left-arc` or `right-arc` reduces the stack size by 1 and `shift` increase the stack size by 1. There are maximum n `shift` operation since each `shift` also reduces the queue size by 1. Therefore, maximum number of `left-arc` and `right-arc` is also n. Thus, maximum number of transitions is $2 \times n$. Moreover, the parsing algorithm can always pick a valid transition for each configuration because `shift` is always a valid one (except if the current configuration is the terminal configuration). There are many variations of the transition based parsing. Nivre (2009) introducing `swap` transition to deal with non-projective dependency parsing. Chen and Manning (2014) exploited neural network based classifier for the parsing algorithm instead of the original support vector machine classifier which we extended for our ACL 2015 (§3.1.2), EMNLP 2015 (§3.1.3), ConLL 2015 (§3.1.4) papers.

Graph-based approach

The graph based approach formalizes dependency parsing task as finding the maximum spanning tree on the weighted fully connected graph (McDonald *et al.* 2005). The graph $G = (V, E)$ for a sentence $w = w_0, w_1, w_2....w_n$ is constructed as follow.

- V = $w_0, w_1, ...w_n$
- E = $(w_i, \text{weight}, w_j)$ for every i, j

Chu and Liu (1965) algorithm is used to find the maximum spanning tree. First, all vertexes are selected. Incoming edge with highest weight are added to the graph

[12]This training data can be generated dynamically as in Goldberg and Nivre (2012).

one by one. If the resulting graph is a tree then it's the maximum spanning tree. If it's not, the cycle is collapsed into a single node, the weights are updated and the algorithm is repeated. The remaining question is how to estimate the weight of each edge. McDonald *et al.* (2005) applied the Margin Infused Relaxed Algorithm (MIRA) for estimating those weights on the treebank.

Graph-based approach is the natural solution for non-projective dependency parsing since it makes no assumption on the word order. Unlike transition-based approach, graph-based is exact inference. Therefore, the running time is much slower. However, graph-based approach is more robust. Transition-based approach is prone to errors meaning that error in early state might accumulate and led to bias model. An interesting observation is that errors made by graph-based approach and transition-based approach is very different and mostly not overlapping. Nivre and Mcdonald (2008); Zhang and Clark (2008) proposed method to combine the strength of both approach in a hybrid approach.

Evaluation

The common evaluation metric for dependency parsing is attachment score which is the percentage of word having the correct head thanks to the single-head property of the dependency tree. There are two version of attachment score which are unlabelled attachment score (UAS) and labelled attachment score (LAS). The first one only look at head while the second metric additionally look at dependency labels.

Universal Treebank

Similar with universal POS tagset, it is highly desirable for universal annotation for dependency treebank. Zeman *et al.* (2012) pioneer on building an unify annotation (HAMLED) for treebank in multiple languages. They propose the mapping to transform each language specific treebank to the Prague Dependency Treebank style (Böhmová *et al.* 2001). McDonald *et al.* (2013) took the different approach and built the Google universal treebank for many languages using the Stanford depen-

dency style (de Marneffe and Manning 2008) and the universal POS tagset (Petrov *et al.* 2012). Rosa *et al.* (2014) extended Zeman *et al.* (2012) to build HAMLED 2.0 which covers more than 30 languages using similar annotation with Google universal treebank. As an effort to better accommodate language differences and unify prior work, Nivre *et al.* (2016) proposed universal dependency treebank, employing the Stanford universal dependency annotation (de Marneffe *et al.* 2014) which currently is the largest collection of dependency treebank in more than 40 languages.

2.3.2 Low-resource Dependency Parsing

Consistent dependency treebank annotation typically requires careful guideline design, guideline testing and refinement, annotator quality control etc. We can't expect this high quality resource available for a resource-poor language. In this section, we are going to review prior approaches to build a dependency parser to a target resource-poor language.

Delexicalization approach

This approach builds a delexicalized parser from a resource-rich source language where a treebank is available. The delexicalized parser is built simply by removing lexical features and then apply any standard supervised monolingual parser. This parser is then applied directly to the target resource-poor language. The underlying hypothesis is that aside from lexical items, other features are similar between two languages. Delexicalized parser is first proposed by Zeman *et al.* (2008). They wanted to build parser for Swedish using Danish. Noted that the hypothesis hold true between Swedish and Danish since they are very similar languages. They scored 66.4% F1 labelled attachment score for Swedish. This is an encouraging result since they did not use any external resource such as bilingual dictionary or parallel data.

McDonald *et al.* (2011) also exploit the idea of delexicalized parser. They experiment with 8 European languages. The main contribution of this paper stems from

the incorporation of parallel data to the model. Søgaard (2011) is another example exploiting the delexicalized parser for a target language. Instead of choosing the source language that is similar to the target language, he investigates on choosing the data points from source language that are similar with the target language with respect to POS sequences.

So far, the delexicalized parser only uses POS information. Täckström *et al.* (2013) extended the POS features to other cross-lingual features. They adopted the WALS – World Atlas of Language Structures (Dryer and Haspelmath 2013) – typological features in the similar vein with (Naseem *et al.* 2012). WALS covered basic information such as order of *Subject*, *Object*, *Verb*; order of *Adjective* and *Noun*; order of *Adposition* and *Noun* etc. about nearly 2700 languages. They experiment with 16 languages. For each target language, the rest 15 languages will be the training data. The intuition here is very simple. They want to take advantage of multiple source-languages. Moreover, they also apply self-training and ensemble-training for relexicalized the delexicalized model.

Projection approach

In contrast to the approach using language relatedness clues, i.e. delexicalized parser. In this section, we are going to investigate on the method exploiting parallel data to either project the annotation from the source to the target language or as the constrains for a better model.

Hwa *et al.* (2005) is the first to exploit this idea. The key assumption of this paper is the direct correspondence hypothesis between parallel text. Using this assumption, they define a set of actions for each of the one-to-one, one-to-many, many-to-one, one-to-null or many-to-many alignment. Given a source-language parsed tree and the word alignment, they apply a series of defined actions to generate the target-language parsed tree. However, the performance of this direct transfer is quite poor. They resolve this by applying a set of post-processing rules which capture the language specific knowledge. They achieved 72.1 and 53.9% UAS for Spanish and Chinese respectively. However The approach of Hwa *et al.*

	de	el	es	it	nl	pt	sv	Avg (7)
Direct Transfer	47.2	63.9	53.3	57.7	60.8	69.2	58.3	58.6
Täckström *et al.* (2012)	50.7	63.0	62.9	68.8	54.3	71.0	56.9	61.1
McDonald *et al.* (2011)	50.9	66.8	55.8	60.8	67.8	71.3	61.3	62.1
Ma and Xia (2014)	57.3	67.4	60.3	64.0	68.2	75.1	66.7	65.6
Täckström *et al.* (2013)	61.5	69.6	66.9	73.4	60.2	79.9	65.5	68.1

Table 2.5: Unlabelled attachment score (UAS) of different models across seven languages.

(2005) contains many heuristics and rules, which will be difficult to adapt to different languages.

Täckström *et al.* (2012) built the delexicalized parser but additionally use cross-lingual word clustering induced from parallel data as a feature. The algorithm they used to induce cross-lingual word cluster is an extension of the traditional Brown algorithm (Brown *et al.* 1992). They incorporate the monolingual language model and the alignment information to the final model which is trained on massive amount of parallel sentences.

Ma and Xia (2014) transfer the parameters of dependency parsers from source language to target language using parallel data and target language monolingual data. They trained a supervised English dependency parser as the source parser. They optimize the objective function that (1) minimize the uncertainty of the target language using monolingual data (2) the distribution of the target parser should be similar to the source parser through the word alignment.

2.3.3 Summary of Approaches

There are 2 main approaches for low-resource dependency parsing using delexicalized parser and projection. Table 2.5 summaries the performance of different models across 7 common languages. The models listed in Table 2.5 are sorted in the ascending average performance. Direct transfer is the delexicalized model of Mc-

Donald *et al.* (2011). These approaches differ in the resource requirement. The baseline Direct Transfer requires nothing specific about target language aside from the consensus POS tagset. Täckström *et al.* (2012) needed huge amount of parallel data for induce cross-lingual word clusters. McDonald *et al.* (2011) also need parallel data to constrain the model however, the amount of parallel data used is much less. Ma and Xia (2014) also use parallel data to project the parameters from source to target language. Täckström *et al.* (2013) didn't use any parallel data however they combine clues from many different source languages to a single target language. All in all, the common denominator among these approaches is that they all use the delexicalized model. In our ConLL 2015 (§3.1.4) paper, we propose a method to improve the delexicalized parser using no additional resources which is bound to complement all other methods. In our ACL 2015 (§3.1.2) and EMNLP 2015 (§3.1.3) papers, we further improve the performance by combining delexicalized parser with a model trained on a small annotated treebank. This gives give a big boost in the accuracy especially in the low-resource scenario.

2.4 Crosslingual Word Embeddings

Learning crosslingual word embeddings are the third task we considered in this thesis. Crosslingual word embeddings represent lexicons in several languages in the same dense vector space which is very useful for many crosslingual nlp applications in transfer learning setting. Delexicalized parser mentioned above is an example of transfer learning. For delexicalized parser, the lexical features are removed since lexical features are different across language. However, with the help of crosslingual word embeddings, we can add lexical features back to the model which is shown to improve the performance in our ConLL 2015 (§3.1.4) paper.

2.4.1 Monolingual Word Embeddings

Since most crosslingual word embedding techniques are derived from monolingual word embeddings methods. We first review methods for monolingual word embeddings.

Monolingual word embeddings is the extension of the conventional count-based word vector space model following distributional hypothesis. This hypothesis states that the meaning of a word can be induced by the surrounding context. Therefore, each word can be represented as a vector of co-occurrence counts with words in the local context. Normally latent semantic analysis or singular value decomposition is applied to this vector to reduce the dimension which is usually the size of vocabulary. Word embeddings are the relatively new field of research, learning distributed representation of a word as oppose to the conventional distributional representation (Blacoe and Lapata 2012; Baroni *et al.* 2014). While distributional representation collects the word co-occurrence count, word embeddings are usually formalized as supervised machine learning task to predict the word that appear in a context (Collobert and Weston 2008; Mikolov *et al.* 2013c; Bengio *et al.* 2003; Turian *et al.* 2010; Huang *et al.* 2012; Pennington *et al.* 2014).

Despite relatively young research field, word embeddings attracts much attention lately, having widespread success in many NLP applications such as natural language understanding (Collobert and Weston 2008), sentiment analysis (Socher *et al.* 2013), dependency parsing (Dyer *et al.* 2015) and machine translation (Bahdanau *et al.* 2014). There are wealth of prior works on word embeddings. Bengio *et al.* (2003) pioneer on building word embeddings as part of training neural language model. The main drawback of this approach is that it is too slow to train on big dataset since the objective function is normalized over the vocabulary size which is usually big. Collobert and Weston (2008) use down-stream tasks such as POS tagging, named entity recognition, noun-phrase chunking instead of language model for learning shared compatible word embeddings across tasks. More recently, Mnih and Kavukcuoglu (2013) propose vector log-bilinear language model (vLBL) and invert vector log- bilinear language model (ivLBL) for learning word

embeddings as the by-product of neural language model. Mikolov *et al.* (2013c) proposed continuous bag-of-word (CBOW) and SkipGram model in a very similar way with vLBL and ivLBL model. In the vLBL and CBOW model, the words in the context windows are used to predict the central word, while in the ivLBL and SkipGram model, the central word is used to predict words in the context. Training in both Mnih and Kavukcuoglu (2013) and Mikolov *et al.* (2013c) are fast thanks to hierarchical softmax and noise-contrastive estimation. Hierarchical softmax use tree-structure to compute the output probability reducing the complexity to logarithm of vocabulary size. Noise contrastive estimation and negative sampling apply for unnormalized model, discriminating between samples from training data and samples from some noise distribution. This effectively reduces the algorithm complexity from vocabulary size (e.g. 100k) to the number of samples which is usually small (e.g 5). However, as the context windows slides through the training data, training in both Mikolov *et al.* (2013c) and Mnih and Kavukcuoglu (2013) is still proportional to the corpus size. Pennington *et al.* (2014) proposed global vector model (GloVe) that work directly on the global pre-computed word co-occurrence statistic. In this way, they can train the model proportional to the co-occurrence pair, scale independently to the corpus size.

There are many variations of word embeddings proposed lately. The word embeddings are usually trained on the monolingual data capturing word-context relations. However, Chen *et al.* (2014a) trained the embeddings on the dependency treebank which instead, captures the head-modifier relation. Rothe and Schütze (2015) incorporate information from knowledge base such as WordNet (Miller 1995) to the word embeddings. Iacobacci *et al.* (2015), Chen *et al.* (2014b) and Tian *et al.* (2014) learn the sense embeddings instead of word embeddings since a word might have several senses. Other work go over word boundary and learn phrase, sentence or document embeddings (Kiros *et al.* 2015; Tai *et al.* 2015; Kalchbrenner *et al.* 2014; Le and Mikolov 2014).

Evaluation

Monolingaul word embeddings are usually evaluated on word similarity tasks. Given tuples of $(\text{word}_1, \text{word}_2, \texttt{s})$ where $\texttt{s}$ is a scalar denoting the semantic similarity between word_1 and word_2 given by human annotators. Good word embeddings should produce the score correlated with human judgement. There are many dataset like that to test different syntactic and semantic relations such as WordSim353 (Finkelstein *et al.* 2001), RareWord (Luong *et al.* 2015), MEN (Bruni *et al.* 2012) and SimLex-999 (Hill *et al.* 2015).

Monolingual word embeddings are also usually evaluated on analogy tasks proposed by Mikolov *et al.* (2013c). This task aims at answering the question "a is to b as c is to d" where a,b,c is given and the system must predict d. For example system must answer "Japan" to the following question "Paris is to France as Tokyo is to what?". There are two main datasets for this task which are MSR dataset (Mikolov *et al.* 2013c) and Google dataset (Mikolov *et al.* 2013a).

Levy *et al.* (2015) and Baroni *et al.* (2014) shed the light in understanding and comparing embeddings models with respect to the count-based methods in a controlled setting. Comparing various embeddings models, they observed that CBOW and skipgram with negative sampling achieved consistently high results across different settings. This is why we extended CBOW with negative sampling in both our EMNLP 2016 (§3.1.5) and EACL 2017 (§3.1.6) paper (remove if not accepted).

2.4.2 Building Crosslingual Word Embeddings

There is a wealth of prior work on crosslingual word embeddings, which all exploit some kind of bilingual resource. This is often in the form of a parallel bilingual text, using word alignments as a bridge between tokens in the source and target languages, such that translations are assigned similar embedding vectors (Luong *et al.* 2015; Klementiev *et al.* 2012; Zou *et al.* 2013). Klementiev *et al.* (2012) and Zou *et al.* (2013) build the alignment matrix $\mathbf{A}$ of size $|V_e| \times |V_f|$ where V_e and V_f are vocabulary of source and target language. This matrix is then used to relate source and target embeddings as part of the training. Luong *et al.* (2015),

on the other hand, use the alignment directly by extending the SkipGram model from Mikolov *et al.* (2013c). They predict the target language context using source language word which is specified by the alignment.

These approaches are affected by errors from automatic word alignments, motivating other approaches which operate at the sentence level (Chandar A P *et al.* 2014; Hermann and Blunsom 2014; Gouws *et al.* 2015). Hermann and Blunsom (2014) learn compositional vector representations of sentences from individual word embeddings and constrains that sentences and their translations representations closely match. Chandar A P *et al.* (2014) extend the approach to emphrasize the monolingual property of learned embedding. They minimize the reconstruction cost from source to target, target to source, source to source and target to target jointly. Gouws *et al.* (2015) adopted the idea but the monolingual constrains is from external monolingual data. The word embeddings learned this way capture translational equivalence, despite not using explicit word alignments. Nevertheless, these approaches demand large parallel corpora, which are not available for many language pairs.

Vulić and Moens (2015) use bilingual comparable text, sourced from Wikipedia. Their approach creates a psuedo-document by forming a bag-of-words from the lemmatized nouns in each comparable document concatenated over both languages. These pseudo-documents are then used for learning vector representations using `Word2Vec`. Their system, despite its simplicity, performed surprisingly well on a bilingual lexicon induction task. Their approach is compelling due to its lesser resource requirements, although comparable bilingual data is scarce for many languages too. Related, Søgaard *et al.* (2015) exploit the comparable part of Wikipedia. They represent word using Wikipedia entries which are shared for many languages.

A bilingual dictionary is an alternative source of bilingual information. Gouws and Søgaard (2015) randomly replace the text in a monolingual corpus with a random translation, using this corpus for learning word embeddings. Their approach doesn't handle polysemy, as very few of the translations for each word will be valid in context. They maximize the probability of a word given context $p(w_i|h)$ where w_i is the middle word and h is computed from k surrounding words

$\{w_{i-k}, w_{i-k+1}, ..., w_{i+k-1}, w_{i+k}\}$. Assuming that each word in the context of window k have q translations, there can be as much as q^{2k} possible contexts and out of that only a handful is correct. For this reason a high coverage or noisy dictionary with many translations might lead to poor outcomes. Mikolov *et al.* (2013b), Xiao and Guo (2014) and Faruqui and Dyer (2014) filter a bilingual dictionary for one-to-one translations, thus side-stepping the problem, however discarding much of the information in the dictionary. Our approach in EMNLP 2016 (§3.1.5) and EACL 2017 (§3.1.6) also uses a dictionary, however we use all the translations and explicitly disambiguate translations during training.

Aside from bilingual data requirement, another distinguishing feature on the related work is the method for training embeddings. Mikolov *et al.* (2013b) and Faruqui and Dyer (2014) use a cascade style of training where the word embeddings in both source and target language are trained separately and then combined later using the dictionary. Mikolov *et al.* (2013b) learn the linear transformation to transform the source embeddings to the same space with the target embeddings. Faruqui and Dyer (2014), on the other hand, use canonical correlation analysis to map both source and target embeddings to the same space. Most of the other works train multlingual models jointly where the embeddings of both source and target are learned together satisfying some constraints. This appears to have better performance over cascade training (Gouws *et al.* 2015). For this reason we also use a form of joint training in this thesis.

The other important factor for crosslingual word embeddings is the ability to extend to multiple languages. Previous work mainly focuses on building word embeddings for a pair of languages, typically with English on one side, with the exception of Coulmance *et al.* (2015), Søgaard *et al.* (2015) and Ammar *et al.* (2016). Coulmance *et al.* (2015) extend the bilingual skipgram model from Luong *et al.* (2015), training jointly over many languages using the Europarl corpora. That is instead of using the source language word to predict a target language context, they jointly predict target language in multiple languages. Huang *et al.* (2015) adapted for multiple languages also using bilingual corpora based on the observation that crosslingual word embeddings must be invariant to translation be-

Paper	Bilingual resource	External mono	Multi langs
Zou *et al.* (2013)	parallel corpus	no	no
Klementiev *et al.* (2012)	parallel corpus	no	no
Luong *et al.* (2015)	parallel corpus	yes	no
Chandar A P *et al.* (2014)	parallel corpus	no	no
Hermann and Blunsom (2014)	parallel corpus	no	no
Gouws *et al.* (2015)	parallel corpus	yes	no
Vulić and Moens (2015)	comparable corpus	no	no
Gouws and Søgaard (2015)	dictionary	yes	no
Mikolov *et al.* (2013b)	dictionary	yes	no
Faruqui and Dyer (2014)	dictionary	yes	no
Xiao and Guo (2014)	dictionary	yes	no
Søgaard *et al.* (2015)	Wikipedia entries	no	yes
Coulmance *et al.* (2015)	parallel corpus	yes	yes
Ammar *et al.* (2016)	dictionary	yes	yes
Huang *et al.* (2015)	parallel corpus	no	yes

Table 2.6: Summary of crosslingual word embeddings papers according to the bilingual resources used, support for incorporation of external monolingual data and support for extension to multiple languages.

tween languages. However, big parallel data is an expensive resource for many low-resource languages. While Coulmance *et al.* (2015) use English as the pivot language, Søgaard *et al.* (2015) learn multilingual word embeddings for many languages using Wikipedia entries which are the same for many languages. However, their approach is limited to languages covered in Wikipedia and seems to underperform other methods. Ammar *et al.* (2016) propose two algorithms namely MultiCluster and MultiCCA for multilingual word embeddings using set of bilingual dictionaries. MultiCluster first builds the graph where nodes are lexicon and edges are translations. Each cluster in this graph is an anchor point for building multi-

lingual word embeddings. MultiCCA is an extension of Faruqui and Dyer (2014), performing canonical correlation analysis (CCA) for multiple languages using English as the pivot language. A shortcoming of MultiCCA is that it ignores polysemous translations by retaining on only one-to-one dictionary pairs, disregarding much information (Gouws *et al.* 2015).

Table 2.6 summaries crosslingual word embeddings papers and their differences in term of resource usage and ability to incorporate monolingual data and extension to multiple languages. Incorporation of monolingual data is important for capturing monolingual similarity, however, some methods are not capable of doing so mostly because of complicated objective function (Luong *et al.* 2015). Extending to multiple languages is also desirable as we have a share space for multiple languages enabling multilingual applications such as multi-source machine translation (Zoph and Knight 2016) and multi-source transfer dependency parsing (McDonald *et al.* 2011). Our work start by building crosslingual word embeddings for a pair of language (EMNLP 2016 (§3.1.5)) using noisy dictionary form Panlex and monolingual data. In this way, our approach can be applied to more languages as PanLex covers more than a thousand languages. Afterwards, we extend to multiple languages, jointly learning multilingual word embeddings (EACL 2017 (§3.1.6)).

2.4.3 Evaluation

Evaluating crosslingual word embeddings aim at testing the distances among lexical items from the embedding space. The monolingual distances are usually tested in the same way as monolingual word embeddings, using monolingual word similarity and monolingual word analogy datasets.

The bilingual distance can be tested in several ways. Camacho-Collados *et al.* (2015) propose several crosslingual word similarity datasets, similar with monolingual word similarity dataset, containing tuples of $(\text{word}_1, \text{word}_2, \text{s})$, however word_1 and word_2 are in different language. Vulić and Moens (2015) propose to test the bilingual distance using the bilingual lexicon induction task. Given a word

in a source language, the bilingual lexicon induction task is to predict its translation in the target language using the crosslingual word embeddings. The difficulty of this task is that it is evaluated using the recall of the top ranked word. The model must be very discriminative in order to score well.

The usefulness of crosslingual word embeddings is also evaluated using downstream tasks. Klementiev *et al.* (2012) propose crosslingual document classification task. In this task, the document classifier is trained on a source language and then applied directly to classify a document in the target language. This is convenient for a target low-resource language where we do not have document annotations. The documents are represented as the bag of word embeddings weighted by `tf.idf`. Thanks to the crosslingual word embeddings, the document representation in the target language embeddings is in the same space with the source language, enabling transfer learning. Crosslingual dependency parsing is another commonly used task evaluating crosslingual word embeddings (Ammar *et al.* 2016; Upadhyay *et al.* 2016). In this setup, the source language parser is trained on the source language treebank using only word embeddings i.e. removing all the other features such as part-of-speech and morphology. The source language parser is applied directly to the target language. By removing all other features, this evaluation emphasize the contribution of crosslingual word embeddings.

2.5 Unwritten Language Processing

As mentioned earlier, the tasks we selected to present in our thesis should be representative for processing low-resource languages. We want to cover both semantic and syntactic tasks with multiple input formats. In this section, we discuss the forth task concerning with the extreme case of unwritten language processing. This task is, in fact, substantially different with previous proposed tasks as we did not assume any writing system available for such language. However, this is the common scenario since half of the 7000 languages in the world do not have the orthography system (Lewis 2009). This leave us with no other choice than to work

directly with the speech signal. Unlike previous proposed task, processing unwritten language is a very new task. That is why in this section, we are going to review the current techniques and data requirement for unwritten language processing first before proposing our task.

2.5.1 Unsupervised segmentation and lexical discovery

The input to this task is just the raw speech in an unknown language and the system must be able to segment the continuous speech signal to find word boundary and detect repeated lexical item. Infants are very competitive at this task even during their first year. Toward the end of their first year, they can distinguish between phonetic contrasts (i.e. consonants and vowels), start segmenting continuous speech into words and understand a few words even before starting to talk (Räsänen 2012). While computer system struggle with this task, infants do this naturally without any direct supervision while robust to environmental noise. Zero-resource speech processing challenge (Versteegh *et al.* 2016) is the task set-up to "reverse-engineering" this ability from infants. In this zero-resource setting, the model must jointly learn the representation of the speech signal which help to distinguish between different linguistic unit such as word or phone and then group speech into meaningful words. This will be useful for many tasks such as voice query over raw speech signal (Park 2007) or unsupervised term detection (Jansen and Durme 2011).

Unsupervised term detection (UTD) is the task of finding meaningful spoken words or phrase from the speech signal. Most of the approaches extend segmental dynamic time wrapped proposed by Park (2007). Dynamic time wrapped (DTW) (Berndt and Clifford 1994) is the technique based on dynamic programming to calculate the distance between two temporal sequence of speech of variable length. Segmental DTW calculate the distance based on segments of speech rather than the whole sequence. Most of work on UTD build the graph where each node is an speech segment and the edge is weighted by segmental DTW. How-

ever, Zhang *et al.* (2012) focus more on robust speech feature representation using Boltmann machine. Lyzinski *et al.* (2015) focus more on graph clustering algorithm and Jansen and Durme (2011) focus on improving the efficiency.

UTD aims at segmenting and finding repetitive spoken term for a small subset of vocabulary where the words or phrases are frequent. Full-coverage term discovery, on the other hand, aims at segmenting and clustering the whole vocabulary (Lee *et al.* 2015; Kamper *et al.* 2016a; Kamper *et al.* 2015c; Räsänen *et al.* 2015). Lee *et al.* (2015) build the pseudo-phone acoustic model with the speech segmentation learned as part of training. They also add the noisy-channel to model the phonological variability (i.e. difference between phonetic context and stress) and exploit adaptor grammar (Johnson *et al.* 2006) to group several pseudo-phone units to become syllable and then word. When trained on MIT lecture corpus, most high `tf.idf` words are discovered. Kamper *et al.* (2015c) observed that if we have the speech segmentation, we can convert each segment into a fix sized vector and then the term discovery task will be reduced to clustering which can be done using Gausian Mixture Model (GMM) acoustic model. Moreover, if we have the GMM acoustic model, we can use dynamic programming for finding the speech segmentation. This is the motivation for their Bayesian sampling model for jointly learn the segmentation and GMM acoustic model. However, due to the complexity of the algorithm, they only experimented on small vocabulary dataset of digits from TIDigits dataset (Bocchieri and Doddington 1986). Kamper *et al.* (2016a) extended Kamper *et al.* (2015c) to be able to run on bigger vocabulary. Instead of sampling for all possible segmentations, Kamper *et al.* (2016a) only sample from a small pre-defined set of segmentations as the outputs from Räsänen *et al.* (2015). Also, they employed much simpler method for computing acoustic embeddings from speech features based on down-sampling which basically a technique for averaging with some smoothing. These modifications help to to scale up to large vocabulary unsupervised term discovery and scored well on zero-resource challenges.

Speech segmentation is usually a part of UTD as segmentation is jointly induced. However, speech segmentation can be done separately to provide system with candidate word boundaries. Khanagha *et al.* (2014) used microcanonical multiscale formalism to segment speech analysis to phone-like unit. Currently, they achieved state-of-the-art performance on phone segmentation task on the full TIMIT dataset (Garofolo *et al.* 1993). Räsänen *et al.* (2015) experimented with several speech segmentation methods including (a) VSeg which determine syllable based on velocity of low-pass filtered amplitude envelope (Villing *et al.* 2004). (b) envelope minima detector which find rhythm-based segmentation (Villing *et al.* 2006) and (c) amplitude envelope-driven oscillator (Ghitza 2011). Both Räsänen *et al.* (2015) and Khanagha *et al.* (2014) focus on recall rather than precision, that is they all seem to over segment the speech signal. However, these segmentations provide the list of candidates for further applications (Kamper *et al.* 2016a). In our NAACL 2016 (§3.1.7) paper, we also experimented with speech presegmented using Khanagha *et al.* (2014).

Speech feature extraction is usually the first step when working with speech. Feature extraction convert the digitalized waveform into feature frames. This reduce the variability of speech such as pitch, excitation, voices which makes it easier to process. Mel-frequency cepstral coefficients (MFCC) can be considered as a standard feature extraction method, deriving the ceptral representation of audio (Fang *et al.* 2001). People usually use 12 coefficients with energy together with first and second derivative to capture the change over time resulting in 39 dimensional vector. This is usually calculated for 25 millisecond frame length with 10 millisecond frame step. However, MFCC is criticized for it sensitivity for noise (Shrawankar and Thakare 2013). In our NAACL 2016 (§3.1.7) paper, we represent speech waveform as a sequence of Perceptual Linear Prediction (PLP) vectors (Hermansky 1990). PLP features encode the power spectrum of the speech signal with the focus on minimizing the speaker differences and noise.

Recently, deep neural network inspired features are becoming popular over traditional representation such as MFCC or PLP. Hermansky *et al.* (2000) propose

tandem features which is the top layer of a multilayer perceptron trained to classify phones. Tandem feature represents the probability of each phone given the input. That is why tandem features have the size equal with number of phones. Grezl *et al.* (2007) proposed bottle-neck features which use the middle layer of a bigger multilayer perceptron as the representation instead of the output layer. The bottle-neck layer have much smaller size compared with input and other hidden layer, condensing the information. Kamper *et al.* (2015a) proposed auto encoder based features with top-down constraints. They use a separate UTD to get a set of word pair that served as week top-down supervision. The word pairs are aligned to get frame level alignment using dynamic time wrapped. Each frame-level pairs (e.g. `A` and `B`) are used to train an auto-encoder to minimize the reconstruction cost from `A` to `B` and vice versa. A middle layer of this auto encoder will be used as the features representation. In this way, the feature representation will be better discriminating words.

The input layer for deep neural network feature representation is the conventional representation such as MFCC or PLP but the output is usually more robust to noise, speaker difference and environmental conditions. Moreover, neural network based features extraction can be used to bootstrap a low-resource language. For example, features extracted with Hermansky *et al.* (2000) or Grezl *et al.* (2007) need the phone annotation since they are trained to classify phones. However, this annotation might not be available or very limit for a target low-resource language. However, based on phone annotation from several source languages, we can build the phone-discriminative representation in the target low-resource language (Vesel *et al.* 2012; Stolcke *et al.* 2006; Thomas *et al.* 2012).

2.5.2 Speech recognition for low-resource language

The input for unsupervised lexical discovery is just the speech signal. This is suitable for low-resource language and even unwritten language. However, UTD is not enough to analyse a language. In this section, we will review low-resource language speech recognition which is not particularly suitable for unwritten lan-

guages unless the writing system is invented for that target low-resource language[13]. However, the task we propose latter is similar with low-resource speech recognition which is why it is important to review techniques adapted for low-resource speech recognition.

Conventional approach

Speech recognition outputs the transcription from speech wave form. Conventional approach for speech recognition exploit some Hidden Markov Model (HMM) architecture. There are three mains of components of conventional HMM based speech recognition which are acoustic model, lexical model and language model. Basically, acoustic model convert speech signal to list of sub-word (e.g. phoneme or syllable) representation. Lexical model build the pronunciation dictionary to convert list of sub-word unit into words. Language model gathers words to form the output sentence.

Acoustic model aims at representing the speech wave form as some written representation. It can be mono-phone or syllable for context-independent acoustic model or tri-phone or penta-phone for context-dependent acoustic model. The purpose of this step is to reduce variability and complexity of speech signal. Also it is easier to work with some written form rather than directly with the wave-form. The first step of acoustic model is feature extraction mentioned earlier using methods such as MFCC, PLP, bottle-neck or tandem features. In the conventional context-independent HMM-phoneme based acoustic model, the sequence of feature frames are converted to phoneme representation. The system is trained on the phoneme transcription which is usually the output of manual labelling (extremely time-consuming) or mapping between pronunciation dictionary and word transcription (usually used). However, the phone transcription might not be available for many low-resource languages. However, the acoustic model can be ported from higher resource languages. Le and Besacier (2009) proposed several meth-

[13] As done for Levantine Arabic and Iraqui Arabic as part of DARPA projects

ods for phone mapping between source and target languages, successfully applied for low-resourced language Vietnamese. Siniscalchi *et al.* (2013) proposed a set of shared fundamental unit that is universal across languages, facilitating the acoustic model sharing. Stuker *et al.* (2003) proposed similar set but based on International Phonetic Alphabet (IPA). In fact, there are many speech related work that by-pass the complexity of speech signal processing by assuming the availability of phoneme sequence or phoneme lattice as the output of multilingual acoustic model (Stahlberg *et al.* 2012; Adams *et al.* 2016).

Lexical model is used to create pronunciation dictionary specifying the decomposition of word to sub-word spoken unit (e.g. phoneme). This pronunciation dictionary is usually used in acoustic model as mentioned earlier. Usually, the pronunciation is built manually. The crowd-source dictionary from Wiktionary [14] providing phonemic annotation written in IPA, is a great source of pronunciation dictionary (Schlippe *et al.* 2014). Instead of using phoneme, grapheme based approach can be used to build pronunciation dictionary. This is very useful for languages where there is a close relation between grapheme and phoneme. For example, in Vietnamese, children can always pronounce an unknown word correctly given the written form. This approach has been applied successfully to extract grapheme pronunciation dictionary for Vietnamese (Le and Besacier 2009), Thai (Stüker 2008). However, the problem with grapheme- based dictionary is that the acoustic model have to work on grapheme too which is usually not shareable across languages. Grapheme-to-phoneme is the solution to map grapheme back to phoneme. The mapping can be created manually or automatically using machine translation approaches (Karanasou and Lamel 2010; Cucu *et al.* 2012).

Language model put the constraint on certain order and co-occurrence of words, help to distinguishing between words or phrases that sound similar which may confuse lexical and acoustic model. Language model is usually trained on large monolingual data using n-gram language model or more recently proposed neural

[14] Wiktionary.org

language model (Collobert and Weston 2008; Mikolov *et al.* 2013c; Turian *et al.* 2010; Huang *et al.* 2012; Pennington *et al.* 2014).

Modern approach

Conventional speech recognition require speech waveform, word transcription and pronunciation dictionary exploiting some HMM framework. Modern approaches for speech recognition usually only require speech signal and the transcription exploiting deep neural network. Maas *et al.* (2015) extended Graves *et al.* (2013) and use bidirectional recurrent neural network with connectionist temporal classification loss function (Graves *et al.* 2006) to generate the transcription character by character. Chorowski *et al.* (2014) is the first to train the end-to-end speech recognition system based on attentional model proposed originally for machine translation task by Bahdanau *et al.* (2014). Unlike machine translation, there is no reordering in speech recognition, that is why they added a constraint to prefer the monotonic alignment between speech frame and transcription. In our NAACL 2016 (§3.1.7) paper, we also experimented with automatic speech recognition task and observed substantial improvement adding this monotonic constrain. Chan *et al.* (2015) also use the similar sequence to sequence framework with attention mechanism but make it more robust by randomly introducing noise during training which leads to substantial improvement. They also introduce pyramidal structure for condensing speech signal which we adopted for our NAACL 2016 (§3.1.7) paper. All these modification makes their result close to the state-of-the-art HMM-based speech recognizer. Moreover, in term of resource, modern approaches are more suitable for low-resource languages since no pronunciation dictionary is required.

2.5.3 Low-resource Speech Data Collection

The unsupervised term discovery task just requires speech signal which can be cheaply collected for many low-resource languages through radio broadcast, field work or public speech. However, the raw speech waveform is not very

meaningful, that is why field workers usually collect some annotation for a target low-resource language. Transcription is usually used for language that have some writing system. Vries *et al.* (2014) introduces smartphone-based data collection tool, Woefzela, to collect speech and transcription with the focus on quality control. They collected almost 800 hours of speech on their South African data collection project demonstrating the usefulness of smartphone devices to cheaply and efficiently collect data. Aikuma (Bird *et al.* 2014b) is another smartphone application in this line. However, they use Aikuma to collect the parallel speech between the unwritten language and a higher resource language for language preservation purpose. This is motivated by the fact that usually aside from their mother language, people speak another higher resource language. The higher resource language can be used to approximate and understand the unwritten language. By recording the parallel speech, even when the unwritten language die out, we still have the footprint of that language. For the initial experiment, Bird *et al.* (2014a) managed to collect around 10 hours of parallel speech from indigenous communities in Brazil and Nepal. Blachon *et al.* (2016) used an extended version of Aikuma to collect more than 80 hours of parallel speech from Congo-Brazzaville.

2.5.4 The Propose Task

As mention before, unsupervised term discovery is not very useful, automatic speech recognition is not suitable for unwritten language. Given the availability of tools and data initially collected by Bird *et al.* (2014a) and Blachon *et al.* (2016), we propose a task to model the relationship in the parallel speech between an unwritten language and the target higher resource language. Since the target language is high resource, we can crowd-source or apply automatic speech recognition for getting target language transcription. If we do this, we reduce the task to speech translation where the source is the unwritten language speech and the target is the translation text in the target language.

This is closely related to the speech recognition task where instead of the transcription in the same language, we use a different language translation to under-

stand the semantic of the speech. However, this pose alot of challenges since the monotonic constraint is no longer hold. In our NAACL 2016 (§3.1.7) paper, we apply a deep neural network approach using attentional architecture on this data. We shows that we can learn meaningful relationship directly from this data.

Chapter 3

Research Summary

This chapter is the main contribution of our thesis targeting at building natural language processing (NLP) framework for low-resource languages. However, due to limited time frame, we only focus on four NLP tasks for low-resource languages in this thesis including (1) part-of-speech (2) dependency parsing, (3) cross-lingual word embeddings and (4) unwritten language processing. These tasks are carefully selected to be representative for a language covering both semantic and syntactic aspects, also multi-modal inputs including both text and speech. We believe that those tasks are crucial to process a low-resource language. Since it is usually lack of annotated data for building high quality supervised model, we took the approach of unsupervised or semi-supervised learning for many proposed tasks in this thesis. Specifically, we have successfully applied transfer learning to transfer the knowledge from a source resource rich language to the target resource poor languages resulting in several publications which will be covered in detail latter. Moreover, processing unwritten language is very challenging task where we have to work on speech signal directly. After carefully evaluate the data requirement, we model directly between speech signal and the translation in the target high-resource language. This also result in a publication in our NAACL 2016 (§3.1.7) paper. The rest of this chapter is organized as follow. First, we list all the publications in §3.1 which form the back-bone of this thesis. We then critically evaluate the contribution of this thesis with respect to the research question (§3.2), followed

by future work (§3.3) and conclusion (§3.4).

3.1 Publications

In this section, we will list all related publications, for each publication we will give (1) the full bibliography, (2) the background research process and (3) the retrospective view with critical analysis of contribution toward the thesis target.

3.1.1 EMNLP 2014

Long Duong, Trevor Cohn, Karin Verspoor, Steven Bird and Paul Cook. 2014. What Can We Get From 1000 Tokens? A Case Study of Multilingual POS Tagging for Resource-Poor Languages. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*. 886–897, Doha, Qatar.

Research process

I wanted to continue my master topic which is also about low-resource language processing to the PhD level. During my master, I published two papers about low-resource part-of-speech tagging (Duong *et al.* 2013b; Duong *et al.* 2013a). However, the performance is not very good with the high dependence on quality of parallel data which is the motivation for this paper. This paper can be understand as the wrap-up of my master thesis. Moreover, this fit nicely with the general thesis target, solving the first task, providing analysis about syntax.

In this paper, I implemented the algorithm and write the first draft of the paper. Other co-authors which are my supervisors contribute ideas during our weekly meeting and participated in the paper writing.

Retrospective view

In this paper, we assume the availability of some small POS annotated corpus. Basically we show that we can learn better model compared with purely super-

vised learning taking into consideration parallel corpora. It is good to see that the model still works well when we lower the quality of parallel corpora as in the case for low-resource Malagasy language. However, the main concern is that the performance gain will diminish as we have more annotated data as also shown in the paper. Moreover, the cost for POS annotation is relatively cheap (Garrette and Baldridge 2013). Probably just simply annotate more data and apply any simple supervised learning method will have better benefit-cost ratio.

What Can We Get From 1000 Tokens? A Case Study of Multilingual POS Tagging For Resource-Poor Languages

Long Duong,[12] **Trevor Cohn,**[1] **Karin Verspoor,**[1] **Steven Bird,**[1] and **Paul Cook**[1]
[1]Department of Computing and Information Systems,
The University of Melbourne
[2]National ICT Australia, Victoria Research Laboratory
lduong@student.unimelb.edu.au
{t.cohn, karin.verspoor, sbird, paulcook}@unimelb.edu.au

Abstract

In this paper we address the problem of multilingual part-of-speech tagging for resource-poor languages. We use parallel data to transfer part-of-speech information from resource-rich to resource-poor languages. Additionally, we use a small amount of annotated data to learn to "correct" errors from projected approach such as tagset mismatch between languages, achieving state-of-the-art performance (91.3%) across 8 languages. Our approach is based on modest data requirements, and uses minimum divergence classification. For situations where no universal tagset mapping is available, we propose an alternate method, resulting in state-of-the-art 85.6% accuracy on the resource-poor language Malagasy.

1 Introduction

Part-of-speech (POS) tagging is a crucial task for natural language processing (NLP) tasks, providing basic information about syntax. Supervised POS tagging has achieved great success, reaching as high as 95% accuracy for many languages (Petrov et al., 2012). However, supervised techniques need manually annotated data, and this is either lacking or limited in most resource-poor languages. Fully unsupervised POS tagging is not yet useful in practice due to low accuracy (Christodoulopoulos et al., 2010). In this paper, we propose a semi-supervised method to narrow the gap between supervised and unsupervised approaches. We demonstrate that even a small amount of supervised data leads to substantial improvement.

Our method is motivated by the availability of parallel data. Thanks to the development of multilingual documents from government projects, book translations, multilingual websites, and so forth, parallel data between resource-rich and resource-poor languages is relatively easy to acquire. This parallel data provides the bridge that permits us to transfer POS information from a resource-rich to a resource-poor language.

Systems that make use of cross-lingual tag projection typically face several issues, including mismatches between the tagsets used for the languages, artifacts from noisy alignments and cross-lingual syntactic divergence. Our approach compensates for these issues by training on a small amount of annotated data on the target side, demonstrating that only 1k tokens of annotated data is sufficient to improve performance.

We first tag the resource-rich language using a supervised POS tagger. We then project POS tags from the resource-rich language to the resource-poor language using parallel word alignments. The projected labels are noisy, and so we use various heuristics to select only "good" training examples. We train the model in two stages. First, we build a maximum entropy classifier T on the (noisy) projected data. Next, we train a supervised classifier P on a small amount of annotated data (1,000 tokens) in the target language, using a minimum divergence technique to incorporate the first model, T. Compared with the state of the art (Täckström et al., 2013), we make more-realistic assumptions (e.g. relying on a tiny amount of annotated data rather than a huge crowd-sourced dictionary) and use less parallel data, yet achieve a better overall result. We achieved 91.3% average accuracy over 8 languages, exceeding Täckström et al. (2013)'s result of 88.8%.

The test data we employ makes use of mappings from language-specific POS tag inventories to a universal tagset (Petrov et al., 2012). However, such a mapping might not be available for resource-poor languages. Therefore, we also propose a variant of our method which removes the

need for identical tagsets between the projection model T and the correction model P, based on a two-output maximum entropy model over tag pairs. Evaluating on the resource-poor language Malagasy, we achieved 85.6% accuracy, exceeding the state-of-the-art of 81.2% (Garrette et al., 2013).

2 Background and Related Work

There is a wealth of prior work on multilingual POS tagging. The simplest approach takes advantage of the typological similarities that exist between languages pairs such as Czech and Russian, or Serbian and Croatian. They build the tagger — or estimate part of the tagger — on one language and apply it to the other language (Reddy and Sharoff, 2011, Hana et al., 2004).

Yarowsky and Ngai (2001) pioneered the use of parallel data for projecting tag information from a resource-rich language to a resource-poor language. Duong et al. (2013b) used a similar method on using sentence alignment scores to rank the goodness of sentences. They trained a seed model from a small part of the data, then applied this model to the rest of the data using self-training with revision.

Das and Petrov (2011) also used parallel data but additionally exploited graph-based label propagation to expand the coverage of labelled tokens. Each node in the graph represents a trigram in the target language. Each edge connects two nodes which have similar context. Originally, only some nodes received a label from direct label projection, and then labels were propagated to the rest of the graph. They only extracted the dictionary from the graph because the labels of nodes are noisy. They used the dictionary as the constraints for a feature-based HMM tagger (Berg-Kirkpatrick et al., 2010). Both Duong et al. (2013b) and Das and Petrov (2011) achieved 83.4% accuracy on the test set of 8 European languages.

Goldberg et al. (2008) pointed out that, with the presence of a dictionary, even an incomplete one, a modest POS tagger can be built using simple methods such as expectation maximization. This is because most of the time, words have a very limited number of possible tags, thus a dictionary that specifies the allowable tags for a word helps to restrict the search space. With a gold-standard dictionary, Das and Petrov (2011) achieved an accuracy of approximately 94% on the same 8 languages. The effectiveness of a gold-standard dictionary is undeniable, however it is costly to build one, especially for resource-poor languages. Li et al. (2012) used the dictionary from Wiktionary,[1] a crowd-sourced dictionary. They scored 84.8% accuracy on the same 8 languages. Currently, Wiktionary covers over 170 languages, but the coverage varies substantially between languages and, unsurprisingly, it is poor for resource-poor languages. Therefore, relying on Wiktionary is not effective for building POS taggers for resource-poor languages.

Täckström et al. (2013) combined both token information (from direct projected data) and type constraints (from Wiktionary's dictionary) to form the state-of-the-art multilingual tagger. They built a tag lattice and used these token and type constraints to prune it. The remaining paths are the training data for a CRF tagger. They achieved 88.8% accuracy on the same 8 languages.

Table 1 summarises the performance of the above models across all 8 languages. Note that these methods vary in their reliance on external resources. Duong et al. (2013b) use the least, i.e. only the Europarl Corpus (Koehn, 2005). Das and Petrov (2011) additionally use the United Nation Parallel Corpus. Li et al. (2012) didn't use any parallel text but used Wiktionary instead. Täckström et al. (2013) exploited more parallel data than Das and Petrov (2011) and also used a dictionary from Li et al. (2012).

Another approach for resource-poor languages is based on the availability of a small amount of annotated data. Garrette et al. (2013) built a POS tagger for Kinyarwanda and Malagasy. They didn't use parallel data but instead exploited four hours of manual annotation to build ∼4,000 tokens or ∼3,000 word-types of annotated data. These tokens or word-types were used to build a tag dictionary. They employed label propagation for expanding the coverage of this dictionary in a similar vein to Das and Petrov (2011). They built training examples using this dictionary and then trained the tagger on this data. They achieved 81.9% and 81.2% accuracy for Kinyarwanda and Malagasy respectively.

The method we propose in this paper is similar in only using a small amount of annotation. However, we directly use the annotated data to train the model rather than using a dictionary. We argue

[1] http://www.wiktionary.org/

	da	nl	de	el	it	pt	es	sv	Average
Das and Petrov (2011)	83.2	79.5	82.8	82.5	86.8	87.9	84.2	80.5	83.4
Duong et al. (2013b)	85.6	84.0	85.4	80.4	81.4	86.3	83.3	81.0	83.4
Li et al. (2012)	83.3	86.3	85.4	79.2	86.5	84.5	86.4	86.1	84.8
Täckström et al. (2013)	88.2	85.9	90.5	89.5	89.3	91.0	87.1	88.9	88.8

Table 1: Previously published token-level POS tagging accuracy for various models across 8 languages — Danish (da), Dutch (nl), German (de), Greek (el), Italian (it), Portuguese (pt), Spanish (es), Swedish (sv) — evaluated on CoNLL data (Buchholz and Marsi, 2006).

that with a proper "guide", we can take advantage of very limited annotated data.

2.1 Annotated data

Our annotated data mainly comes from CoNLL shared tasks on dependency parsing (Buchholz and Marsi, 2006). The language specific tagsets are mapped into the universal tagset. We will use this annotated data mainly for evaluation. Table 2 shows the size of annotated data for each language. The 8 languages we are considering in this experiment are not actually resource-poor languages. However, running on these 8 languages makes our system comparable with previously proposed methods. Nevertheless, we try to use as few resources as possible, in order to simulate the situation for resource-poor languages. Later in Section 6 we adapt the approach for Malagasy, a truly resource-poor language.

2.2 Universal tagset

We employ the universal tagset from (Petrov et al., 2012) for our experiment. It consists of 12 common tags: *NOUN, VERB, ADJ* (adjective), *ADV* (adverb), *PRON* (pronoun), *DET* (determiner and article), *ADP* (preposition and postposition), *CONJ* (conjunctions), *NUM* (numerical), *PRT* (particle), *PUNC* (punctuation) and *X* (all other categories including foreign words and abbreviations). Petrov et al. (2012) provide the mapping from each language-specific tagset to the universal tagset.

The idea of using the universal tagset is of great use in multilingual applications, enabling comparison across languages. However, the mapping is not always straightforward. Table 2 shows the size of the annotated data for each language, the number of tags presented in the data, and the list of tags that are not matched. We can see that only 8 tags are presented in the annotated data for Danish, i.e, 4 tags (*DET, PRT, PUNC,* and *NUM*) are missing.[2] Thus, a classifier using all 12 tags will be heavily penalized in the evaluation.

Li et al. (2012) considered this problem and tried to manually modify the Danish mappings. Moreover, *PRT* is not really a universal tag since it only appears in 3 out of the 8 languages. Plank et al. (2014) pointed out that *PRT* often gets confused with *ADP* even in English. We will later show that the mapping problem causes substantial degradation in the performance of a POS tagger exploiting parallel data. The method we present here is more target-language oriented: our model is trained on the target language, in this way, only relevant information from the source language is retained. Thus, we automatically correct the mapping, and other incompatibilities arising from incorrect alignments and syntactic divergence between the source and target languages.

Lang	Size(k)	# Tags	Not Matched
da	94	8	DET, PRT, PUNC, NUM
nl	203	11	PRT
de	712	12	
el	70	12	
it	76	11	PRT
pt	207	11	PRT
es	89	11	PRT
sv	191	11	DET
AVG	205		

Table 2: The size of annotated data from CoNLL (Buchholz and Marsi, 2006), and the number of tags included and missing for 8 languages.

3 Directly Projected Model (DPM)

In this section we describe a maximum entropy tagger that only uses information from directly

[2]Many of these are mistakes in the mapping, however, they are indicative of the kinds of issues expected in low-resource languages.

projected data.

3.1 Parallel data

We first collect Europarl data having English as the source language, an average of 1.85 million parallel sentences for each of the 8 language pairs. In terms of parallel data, we use far less data compared with other recent work. Das and Petrov (2011) used Europarl and the ODS United Nation dataset, while Täckström et al. (2013) additionally used parallel data crawled from the web. The amount of parallel data is crucial for alignment quality. Since DPM uses alignments to transfer tags from source to target language, the performance of DPM (and other models that exploit projection) largely depends on the quantity of parallel data. The "No LP" model of Das and Petrov (2011), which only uses directly projected labels (without label propagation), scored 81.3% for 8 languages. However, using the same model but with more parallel data, Täckström et al. (2013) scored 84.9% on the same test set.

3.2 Label projection

We use the standard alignment tool Giza++ (Och and Ney, 2003) to word align the parallel data. We employ the Stanford POS tagger (Toutanova et al., 2003) to tag the English side of the parallel data and then project the label to the target side. It has been confirmed in many studies (Täckström et al., 2013, Das and Petrov, 2011, Toutanova and Johnson, 2008) that directly projected labels are noisy. Thus we need a method to reduce the noise. We employ the strategy of Yarowsky and Ngai (2001) of ranking sentences using a their alignment scores from IBM model 3.

Firstly, we want to know how noisy the projected data is. Thus, we use the test data to build a simple supervised POS tagger using the TnT tagger (Brants, 2000) which employs a second-order Hidden Markov Model (HMM). We tag the projected data and compare the label from direct projection and from the TnT tagger. The labels from the TnT Tagger are considered as pseudo-gold labels. Column "Without Mapping" from Table 3 shows the average accuracy for the first n-sentences ($n = 60k, 100k, 200k, 500k$) for 8 languages according to the ranking. Column "Coverage" shows the percentages of projected label (the other tokens are Null aligned). We can see that when we select more data, both coverage and accuracy fall. In other words, using the sentence alignment score, we can rank sentences with high coverage and accuracy first. However, even after ranking, the accuracy of projected labels is less than 80% demonstrating how noisy the projected labels are.

Table 3 (column "With Mapping") additionally shows the accuracy using simple tagset mapping, i.e. mapping each tag to the tag it is assigned most frequently in the test data. For example *DET, PRT, PUNC, NUM*, missing from Danish gold data, will be matched to *PRON, X, X, ADJ* respectively. This simple matching yields a $\sim$ 4% (absolute) improvement in average accuracy. This illustrates the importance of handling tagset mapping carefully.

3.3 The model

In this section, we introduce a maximum entropy tagger exploiting the projected data. We select the first 200k sentences from Table 3 for this experiment. This number represents a trade-off between size and accuracy. More sentences provide more information but at the cost of noisier data. Duong et al. (2013b) also used sentence alignment scores to rank sentences. Their model stabilizes after using 200k sentences. We conclude that 200k sentences is enough and capture most information from the parallel data.

Features	Descriptions
W@-1	Previous word
W@+1	Next word
W@0	Current word
CAP	First character is capitalized
NUMBER	Is number
PUNCT	Is punctuation
SUFFIX@k	Suffix up to length 3 ($k <= 3$)
WC	Word class

Table 4: Feature template for a maximum entropy tagger

We ignore tokens that don't have labels, which arise from null alignments and constitute approximately 14% of the data. The remaining data ($\sim$1.4 million tokens) are used to train a maximum entropy (MaxEnt) model. MaxEnt is one of the simplest forms of probabilistic classifier, and is appropriate in this setting due to the incomplete sequence data. While sequence models such as HMMs or CRFs can provide more accurate models of label sequences, they impose a more strin-

Data Size (k)	Coverage (%)	Without Mapping	With Mapping
60	91.5	79.9	84.2
100	89.1	79.4	83.6
200	86.1	79.1	82.9
500	82.4	78.0	81.5

Table 3: The coverage, and POS tagging accuracy with and without tagset mapping of directly projected labels, averaged over 8 languages for different data sizes

Model	da	nl	de	el	it	pt	es	sv	Avg
All features	64.4	83.3	86.3	79.7	82.0	86.5	82.5	76.5	80.2
- Word Class	64.7	82.6	86.6	79.0	82.8	84.6	82.2	76.9	79.9
- Suffix	64.0	82.8	86.3	78.1	81.0	85.9	82.3	76.2	79.6
- Prev, Next Word	62.6	82.5	87.4	79.0	81.9	86.5	82.2	74.8	79.6
- Cap, Num, Punct	64.0	81.9	84.0	78.0	79.1	86.3	81.8	75.6	78.8

Table 5: The accuracy of Directed Project Model (DPM) with different feature sets, removing one feature set at a time

gent training requirement.[3] We also experimented with a first-order linear chain CRF trained on contiguous sub-sequences but observed $\sim 4\%$ (absolute) drop in performance.

The maximum entropy classifier estimates the probability of tag t given a word w as

$$P(t|w) = \frac{1}{Z(w)} \exp \sum_{j=1}^{D} \lambda_j f_j(w,t)\,,$$

where $Z(w) = \sum_t \exp \sum_{j=1}^{D} \lambda_j f_j(w,t)$ is the normalization factor to ensure the probabilities $P(t|w)$ sum to one. Here f_j is a feature function and λ_j is the weight for this feature, learned as part of training. We use Maximum A Posteriori (MAP) estimation to maximize the log likelihood of the training data, $\mathcal{D} = \{w_i, t_i\}_{i=1}^{N}$, subject to a zero-mean Gaussian regularisation term,

$$\begin{aligned}\mathcal{L} &= \log P(\Lambda) \prod_{i=1}^{N} P(t^{(i)}|w^{(i)}) \\ &= -\sum_{j=1}^{D} \frac{\lambda_j^2}{2\delta^2} + \sum_{i=1}^{N} \sum_{j=1}^{D} \lambda_j f_j(w_i, t_i) - \log Z(w_i)\end{aligned}$$

where the regularisation term limits over-fitting, an important concern when using large feature sets. For our experiments we set $\delta^2 = 1$. We use L-BFGS which performs gradient ascent to maximize $\mathcal{L}$. Table 4 shows the features we considered for building the DPM. We use $mkcls$, an unsupervised method for word class induction which is widely used in machine translation (Och, 1999). We run $mkcls$ to obtain 100 word classes, using only the target language side of the parallel data.

Table 5 shows the accuracy of the DPM evaluated on 8 languages ("All features model"). DPM performs poorly on Danish, probably because of the tagset mapping issue discussed above. The DPM result of 80.2% accuracy is encouraging, particularly because the model had no explicit supervision.

To see what features are meaningful for our model, we remove features in turn and report the result. The result in Table 5 disagrees with Täckström et al. (2013) on the word class features. They reported a gain of approximately 3% (absolute) using the word class. However, it seems to us that these features are not especially meaningful (at least in the present setting). Possible reasons for the discrepancy are that they train the word class model on a massive quantity of external monolingual data, or their algorithms for word clustering are better (Uszkoreit and Brants, 2008). We can see that the most informative features are Capitalization, Number and Punctuation. This makes sense because in languages such as German, capitalization is a strong indicator of *NOUN*. Number and punctuation features ensure that we classify *NUM* and *PUNCT* tags correctly.

[3] Täckström et al. (2013) train a CRF on incomplete data, using a tag dictionary heuristic to define a 'gold standard' lattice over label sequences.

4 Correction Model

In this section we incorporate the directly projected model into a second *correction* model trained on a small supervised sample of 1,000 annotated tokens. Our DPM model is not very accurate; as we have discussed it makes many errors, due to invalid or inconsistent tag mappings, noisy alignments, and cross-linguistic syntactic divergence. However, our aim is to see how effectively we can exploit the strengths of the DPM model while correcting for its inadequacies using direct supervision. We select only 1,000 annotated tokens to reflect a low resource scenario. A small supervised training sample is a more realistic form of supervision than a tag dictionary (noisy or otherwise). Although used in most prior work, a tag dictionary for a new language requires significant manual effort to construct. Garrette and Baldridge (2013) showed that a 1,000 token dataset could be collected very cheaply, requiring less than 2 hours of non-expert time.

Our correction model makes use of a *minimum divergence* (MD) model (Berger et al., 1996), a variant of the maximum entropy model which biases the target distribution to be similar to a static reference distribution. The method has been used in several language applications including machine translation (Foster, 2000) and parsing (Plank and van Noord, 2008, Johnson and Riezler, 2000). These previous approaches have used various sources of reference distribution, e.g., incorporating information from a simpler model (Johnson and Riezler, 2000) or combining in- and out-of-domain models (Plank and van Noord, 2008). Plank and van Noord (2008) concluded that this method for adding prior knowledge only works with high quality reference distributions, otherwise performance suffers.

In contrast to these previous approaches, we consider the specific setting where both the learned model and the reference model $s_o = P(t|w)$ are both maximum entropy models. In this case we show that the MD setup can be simplified to a regularization term, namely a Gaussian prior with a non-zero mean. We model the classification probability, $P'(t|w)$ as the product between a base model and a maximum entropy classifier,

$$P'(t|w) \propto P(t|w) \exp \sum_{j=1}^{D} \gamma_j f_j(w,t)$$

where here we use the DPM model as base model $P(t|w)$. Under this setup, where P' uses the same features as P, and both are log-linear models, this simplifies to

$$\begin{aligned} P'(t|w) &\propto \exp\left(\sum_{j=1}^{D} \lambda_j f_j(w,t) + \sum_{j=1}^{D} \gamma_j f_j(w,t)\right) \\ &\propto \exp \sum_{j=1}^{D} (\lambda_j + \gamma_j) f_j(w,t) \end{aligned} \quad (1)$$

where the constant of proportionality is $Z'(w) = \sum_t \exp \sum_{j=1}^{D} (\lambda_j + \gamma_j) f_j(w,t)$. It is clear that Equation (1) also defines a maximum entropy classifier, with parameters $\alpha_j = \lambda_j + \gamma_j$, and consequently this might seem to be a pointless exercise. The utility of this approach arises from the prior: MAP training with a zero mean Gaussian prior over γ is equivalent to a Gaussian prior over the aggregate weights, $\alpha_j \sim \mathcal{N}(\lambda_j, \sigma^2)$. This prior enforces parameter sharing between the two models by penalising parameter divergence from the underlying DPM model λ. The resulting training objective is

$$\mathcal{L}^{\text{corr}} = \log P(\mathbf{t}|\mathbf{w}, \alpha) - \frac{1}{2\sigma^2} \sum_{j=1}^{D} (\alpha_j - \lambda_j)^2$$

which can be easily optimised using standard gradient-based methods, e.g., L-BFGS. The contribution of the regulariser is scaled by the constant $\frac{1}{2\sigma^2}$.

4.1 Regulariser sensitivity

Careful tuning of the regularisation term σ^2 is critical for the correction model, both to limit overfitting on the very small training sample of 1,000 tokens, and to control the extent of the influence of the DPM model over the correction model. A larger value of σ^2 lessens the reliance on the DPM and allows for more flexible modelling of the training set, while a small value of σ^2 forces the parameters to be close to the DPM estimates at the expense of data fit. We expect the best value to be somewhere between these extremes, and use line-search to find the optimal value for σ^2. For this purpose, we hold out 100 tokens from the 1,000 instance training set, for use as our development set for hyper-parameter selection.

From Figure 1, we can see that the model performs poorly on small values of σ^2. This is understandable because the small σ^2 makes the model

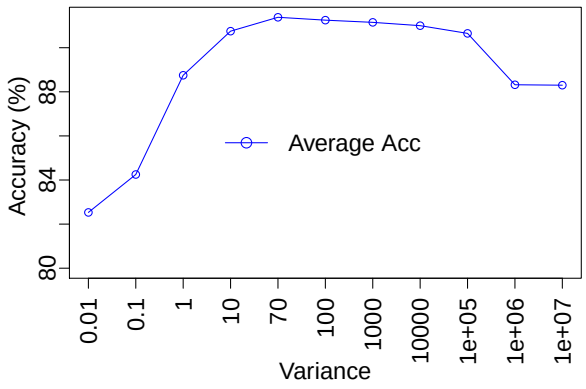


Figure 1: Sensitivity of regularisation parameter σ^2 against the average accuracy measured on 8 languages on the development set

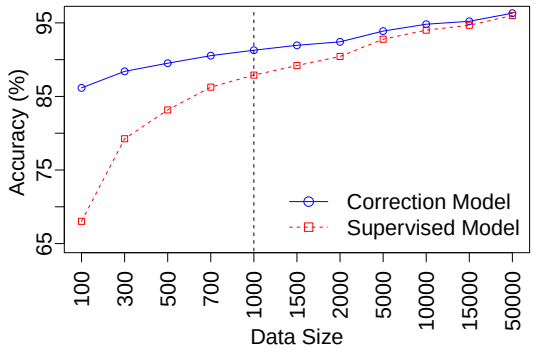


Figure 2: Learning curve for correction model and supervised model: the x-axis is the size of data (number of tokens); the y-axis is the average accuracy measured on 8 languages; the dashed line shows the data condition reported in Table 6

too similar to DPM, which is not very accurate (80.2%). At the other extreme, if σ^2 is large, the DPM model is ignored, and the correction model is equivalent with the supervised model ($\sim$ 88% accuracy). We select the value of $\sigma^2 = 70$, which maximizes the accuracy on the development set.

4.2 The model

Using the value of $\sigma^2 = 70$, we retrain the model on the whole 1,000-token training set and evaluate the model on the rest of the annotated data. Table 6 shows the performance of DPM, Supervised model, Correction model and the state-of-the-art model (Täckström et al., 2013). The supervised model trains a maximum entropy tagger using the same features as in Table 4 on this 1000 tokens. The only difference between the supervised model and the correction model is that in the correction model we additionally incorporate DPM as the prior.

The supervised model performs surprisingly well confirming that our features are meaningful in distinguishing between tags. This model achieves high accuracy on Danish compared with other languages probably because Danish is easier to learn since it contains only 8 tags. Despite the fact that the DPM is not very accurate, the correction model consistently outperforms the supervised model on all considered languages, approximately 4.3% (absolute) better on average. This shows that our method of incorporating DPM to the model is efficient and robust.

The correction model performs much better than the state-of-the-art for 7 languages but slightly worse for 1 language. On average we achieve 91.3% accuracy compared with 88.8% for the state-of-the-art, an error rate reduction of 22.3%. This is despite using fewer resources and only modest supervision.

5 Analysis

Tagset mismatch In the correction model, we implicitly resolve the mismatched tagset issue. DPM might contain tags that don't appear in the target language or generally are errors in the mapping. However, when incorporating DPM into the correction model, only the feature weight of tags that appear in the target language are retained. In general, because we don't explicitly do any mapping between languages, we might have trouble if the tagset size of the target language is bigger than the source language tagset. However, this is not the case for our experiment because we choose English as the source-side and English has the full 12 tags.

Learning curve We investigate the impact of the number of available annotated tokens on the correction model. Figure 2 shows the learning curve of the correction model and the supervised model. We can clearly see the differences between 2 models when the size of training data is small. For example, at 100 tokens, the difference is very large, approximately 18% (absolute), it is also 6% (absolute) better than DPM. This difference diminishes as we add more data. This make sense because when we add more data, the supervised model become stronger, while the effective-

Model	da	nl	de	el	it	pt	es	sv	Avg
DPM	64.4	83.3	86.3	79.7	82.0	86.5	82.5	76.5	80.2
Täckström et al. (2013)	88.2	85.9	90.5	89.5	89.3	91.0	87.1	**88.9**	88.8
Supervised model	90.1	84.6	89.6	88.2	81.4	87.6	88.9	85.4	87.0
Correction Model	**92.1**	**91.1**	**92.5**	**92.1**	**89.9**	**92.5**	**91.6**	88.7	**91.3**
DPM (with dict)	65.2	83.9	87.0	79.1	83.5	87.1	83.0	77.5	80.8
Correction Model (with dict)	93.3	92.2	93.7	93.2	92.2	93.1	92.8	90.0	92.6

Table 6: The comparison of our Directly Projected Model, Supervised Model, Correction Model and the state-of-the-art system (Täckström et al., 2013). The best performance for each language is shown in bold. The models that are built with a dictionary are provided for reference.

ness of the DPM prior on the correction model is wearing off. An interesting observation is that the correction model is always better, even when we add massive amounts of annotated data. At 50,000 tokens, when the supervised model reaches 96% accuracy, the correction model is still 0.3% (absolute) better, reaching 96.3%. It means that even at that high level of confidence, some information can still be added from DPM to the correction model. This improvement probably comes from the observation that the ambiguity in one language is explained through the alignment. It also suggests that this method could improve the performance of a supervised POS tagger even for resource-rich languages.

Our methods are also relevant for annotation projects for resource-poor languages. Assuming that it is very costly to annotate even 100 tokens, applying our methods can save annotation effort but maintain high performance. For example, we just need 100 tokens to match the accuracy of a supervised method trained on 700 tokens, or we just need 500 tokens to match the performance with nearly 2,000 tokens of supervised learning.

Our method is simple, but particularly suitable for resource-poor languages. We need a small amount of annotated data for a high performance POS tagger. For example, we need only around 300 annotated tokens to reach the same accuracy as the state-of-the-art unsupervised POS tagger (88.8%).

Tag dictionary Although, it is not our objective to rely on the dictionary, we are interested in whether the gains from the correction model still persist when the DPM performance is improved. We attempt to improve DPM, following the method of Li et al. (2012) by building a tag dictionary using Wiktionary. This dictionary is then used as a feature which fires for word-tag pairings present in the dictionary. We expect that when we add this additional supervision, the DPM model should perform better. Table 6 shows the performance of DPM and the correction model when incorporating the dictionary. The DPM model only increases 0.6% absolute but the correction model increases 1.3%. Additionally, it shows that our model can improve further by incorporating external information where available.

CRF Our approach of using simple classifiers begs the question of whether better results could be obtained using sequence models, such as conditional random fields (CRFs). As mentioned previously, a CRF is not well suited for incomplete data. However, as our second 'correction' model is trained on complete sequences, we now consider using a CRF in this stage. The training algorithm is as follows: first we estimate the DPM feature weights on the incomplete data as before, and next we incorporate the feature weights into a CRF trained on the 1,000 annotated tokens. This is complicated by the different feature sets between the MaxEnt classifier and the CRF, however the classifier uses a strict subset of the CRF features. Thus, we use the minimum divergence prior for the token level features, and a standard zero-mean prior for the sequence features. That is, the objective function of the CRF correction model becomes:

$$\mathcal{L}_{\text{crf}}^{\text{corr}} = \log P(\mathbf{t}|\mathbf{w}, \alpha) - \frac{1}{2\delta_1^2}\sum_{j\in F_1}(\alpha_j - \lambda_j)^2 - \frac{1}{2\delta_2^2}\sum_{j\in F_2}\alpha_j^2 \quad (2)$$

where F_1 is the set of features referring to only one label as in the DPM maxent model and F_2 is the set of features over label pairs. The union of $F = F_1 \cup F_2$ is the set of all features for the CRF. We perform grid search using held out

data as before for δ_1^2 and δ_2^2. The CRF correction model scores 88.1% compared with 86.5% of the supervised CRF model trained on the 1,000 tokens. Clearly, this is beneficial, however, the CRF correction model still performs worse than the MaxEnt correction model (91.3%). We are not sure why but one reason might be overfitting of the CRF, due to its large feature set and tiny training sample. Moreover, this CRF approach is orthogonal to Täckström et al. (2013): we could use their CRF model as the DPM model and train the CRF correction model using the same minimum divergence method, presumably resulting in even higher performance.

6 Two-output model

Garrette and Baldridge (2013) also use only a small amount of annotated data, evaluating on two resource-poor languages Kinyarwanda (KIN) and Malagasy (MLG). As a simple baseline, we trained a maxent supervised classifier on this data, achieving competitive results of 76.4% and 80.0% accuracy compared with their published results of 81.9% and 81.2% for KIN and MLG, respectively. Note that the Garrette and Baldridge (2013) method is more complicated than this baseline.

We want to further improve the accuracy of MLG using parallel data. Applying the technique from Section 4 will not work directly, due to the tagset mismatch (the Malagasy tagset contains 24 tags) which results in highly different feature sets. Moreover, we don't have the language expertise to manually map the tagset. Thus, in this section, we propose a method capable of handling tagset mismatch. For data, we use a parallel English-Malagasy corpus of ∼100k sentences,[4] and the POS annotated dataset developed by Garrette and Baldridge (2013), which comprises 4230 tokens for training and 5300 tokens for testing.

6.1 The model

Traditionally, MaxEnt classifiers are trained using a single label.[5] The method we propose is trained with pairs of output labels: one for the Malagasy tag (t_M) and one for the universal tag (t_U), which are both predicted conditioned on a Malagasy word (w_M) in context. Our two-output model is defined as

$$P(t_M, t_U|w_M) = \frac{1}{Z(w_M)} \exp\left(\sum_{j=1}^{D} \lambda_j f_j^M(w, t_M) + \sum_{j=1}^{E} \gamma_j f_j^U(w, t_U) + \sum_{j=1}^{F} \alpha_j f_j^B(w, t_M, t_U)\right) \quad (3)$$

where f^M, f^U, f^B are the feature functions considering t_M only, t_U only, and over both outputs t_M and t_U respectively, and $Z(w_M)$ is the partition function. We can think of Eq. (3) as the combination of 3 models: the Malagasy maxent supervised model, the DPM model, and the tagset mapping model. The central idea behind this model is to learn to predict not just the MLG tags, as in a standard supervised model, but also to learn the mapping between MLG and the noisy projected universal tags. Framing this as a two output model allows for information to flow both ways, such that confident taggings in either space can inform the other, and accordingly the mapping weights α are optimised to maximally exploit this effect.

One important question is how to obtain labelled data for training the two-output model, as our small supervised sample of MLG text is only annotated for MLG labels t_M. We resolve this by first learning the DPM model on the projected labels, after which we automatically label our correction training set with predicted tags from the DPM model. That is, we augment the annotated training data from (t_M, w_M) to become (t_M, t_U, w_M). This is then used to train the two-output maxent classifier, optimising a MAP objective using standard gradient descent. Note that it would be possible to apply the same minimum divergence technique for the two-output maxent model. In this case the correction model would include a regularization term over the λ to bias towards the DPM parameters, while γ and α would use a zero-mean regularizer. However, we leave this for future work.

Table 7 summarises the performance of the state-of-the-art (Garrette et al., 2013), the supervised model and the two-output maxent model evaluated on the Malagasy test set. The two-output maxent model performs much better than the supervised model, achieving ∼5.3% (absolute) improvement. An interesting property of this ap-

[4] http://www.ark.cs.cmu.edu/global-voices/

[5] Or else a sequence of labels, in the case of a conditional random field (Lafferty et al., 2001). However, even in this case, each token is usually assigned a single label. An exception is the factorial CRF (Sutton et al., 2007), which models several co-dependent sequences. Our approach is equivalent to a factorial CRF without edges between tags for adjacent tokens in the input.

Model	Accuracy (%)
Garrette et al. (2013)	81.2
MaxEnt Supervised	80.0
2-output MaxEnt (Universal tagset)	85.3
2-output MaxEnt (Penn tagset)	85.6

Table 7: The performance of different models for Malagasy.

proach is that we can use different tagsets for the DPM. We also tried the original Penn treebank tagset which is much larger than the universal tagset (48 vs. 12 tags). We observed a small improvement reaching 85.6%, suggesting that some pertinent information is lost in the universal tagset. All in all, this is a substantial improvement over the state-of-the-art result of 81.2% (Garrette et al., 2013) and an error reduction of 23.4%.

7 Conclusion

In this paper, we thoroughly review the work on multilingual POS tagging of the past decade. We propose a simple method for building a POS tagger for resource-poor languages by taking advantage of parallel data and a small amount of annotated data. Our method also efficiently resolves the tagset mismatch issue identified for some language pairs. We carefully choose and tune the model. Comparing with the state-of-the-art, we are using the more realistic assumption that a small amount of labelled data can be made available rather than requiring a crowd-sourced dictionary. We use less parallel data which as we pointed out in section 3.1, could have been a huge disadvantage for us. Moreover, we did not exploit any external monolingual data. Importantly, our method is simpler but performs better than previously proposed methods. With only 1,000 annotated tokens, less than 1% of the test data, we can achieve an average accuracy of 91.3% compared with 88.8% of the state-of-the-art (error reduction rate $\sim$22%). Across the 8 languages we are substantially better at 7 and slightly worse at one. Our method is reliable and could even be used to improve the performance of a supervised POS tagger.

Currently, we are building the tagger and evaluating through several layers of mapping. Each layer might introduce some noise which accumulates and leads to a biased model. Moreover, the tagset mappings are not available for many resource-poor languages. We therefore also proposed a method to automatically match between tagsets based on a two-output maximum entropy model. On the resource-poor language Malagasy, we achieved the accuracy of 85.6% compared with the state-of-the-art of 81.2% (Garrette et al., 2013). Unlike their method, we additionally use a small amount of parallel data.

In future work, we would like to improve the performance of DPM by collecting more parallel data. Duong et al. (2013a) pointed out that using a different source language can greatly alter the performance of the target language POS tagger. We would like to experiment with different source languages other than English. We assume that we have 1,000 tokens for each language. Thus, for the 8 languages we considered we will have 8,000 annotated tokens. Currently, we treat each language independently, however, it might also be interesting to find some way to incorporate information from multiple languages simultaneously to build the tagger for a single target language.

Acknowledgments

We would like to thank Dan Garreette, Jason Baldridge and Noah Smith for Malagasy and Kinyarwanda datasets. This work was supported by the University of Melbourne and National ICT Australia (NICTA). NICTA is funded by the Australian Federal and Victoria State Governments, and the Australian Research Council through the ICT Centre of Excellence program. Dr Cohn is the recipient of an Australian Research Council Future Fellowship (project number FT130101105).

References

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Proceeding of HLT-NAACL*, pages 582–590.

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *COMPUTATIONAL LINGUISTICS*, 22:39–71.

Thorsten Brants. 2000. TnT: A statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing (ANLP '00)*, pages 224–231, Seattle, Washington, USA.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*.

Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. 2010. Two decades of unsupervised pos induction: How far have we come? In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 575–584.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 600–609.

Long Duong, Paul Cook, Steven Bird, and Pavel Pecina. 2013a. Increasing the quality and quantity of source language data for Unsupervised Cross-Lingual POS tagging. Proceedings of the Sixth International Joint Conference on Natural Language Processing, pages 1243–1249. Asian Federation of Natural Language Processing.

Long Duong, Paul Cook, Steven Bird, and Pavel Pecina. 2013b. Simpler unsupervised POS tagging with bilingual projections. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 634–639. Association for Computational Linguistics.

George Foster. 2000. A maximum entropy/minimum divergence translation model. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 45–52.

Dan Garrette and Jason Baldridge. 2013. Learning a part-of-speech tagger from two hours of annotation. pages 138–147, June.

Dan Garrette, Jason Mielens, and Jason Baldridge. 2013. Real-world semi-supervised learning of pos-taggers for low-resource languages. pages 583–592, August.

Yoav Goldberg, Meni Adler, and Michael Elhadad. 2008. Em can find pretty good hmm pos-taggers (when given a good start. In *In Proc. ACL*, pages 746–754.

Jiri Hana, Anna Feldman, and Chris Brew. 2004. A resource-light approach to Russian morphology: Tagging Russian using Czech resources. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP '04)*, pages 222–229, Barcelona, Spain, July.

Mark Johnson and Stefan Riezler. 2000. Exploiting auxiliary distributions in stochastic unification-based grammars. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, NAACL 2000, pages 154–161.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*, pages 79–86, Phuket, Thailand. AAMT.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289.

Shen Li, João V. Graça, and Ben Taskar. 2012. Wiki-ly supervised part-of-speech tagging. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1389–1398.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29(1):19–51, March.

Franz Josef Och. 1999. An efficient method for determining bilingual word classes. In *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics*, EACL '99, pages 71–76.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

Barbara Plank and Gertjan van Noord. 2008. Exploring an auxiliary distribution based approach to domain adaptation of a syntactic disambiguation model. In *Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, CrossParser '08, pages 9–16.

Barbara Plank, Dirk Hovy, and Anders Søgaard. 2014. Learning part-of-speech taggers with inter-annotator agreement loss. In *Proceedings of the 14th Conference of the European Chapter of the Association for*

Computational Linguistics, pages 742–751, Gothenburg, Sweden, April.

Siva Reddy and Serge Sharoff. 2011. Cross language POS taggers (and other tools) for Indian languages: An experiment with Kannada using Telugu resources. In *Proceedings of IJCNLP workshop on Cross Lingual Information Access: Computational Linguistics and the Information Need of Multilingual Societies. (CLIA 2011 at IJNCLP 2011)*, Chiang Mai, Thailand, November.

Charles Sutton, Andrew McCallum, and Khashayar Rohanimanesh. 2007. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *J. Mach. Learn. Res.*, 8:693–723, May.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 1:1–12.

Kristina Toutanova and Mark Johnson. 2008. A bayesian lda-based model for semi-supervised part-of-speech tagging. In J.C. Platt, D. Koller, and Y. Singer a nd S.T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1521–1528. Curran Associates, Inc.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1 (NAACL '03)*, pages 173–180, Edmonton, Canada.

Jakob Uszkoreit and Thorsten Brants. 2008. Distributed word clustering for large scale class-based language modeling in machine translation. In *In ACL International Conference Proceedings*.

David Yarowsky and Grace Ngai. 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, NAACL '01, pages 1–8.

3.1.2 ACL 2015

Long Duong, Trevor Cohn, Steven Bird, Paul Cook. 2015. Low Resource Dependency Parsing: Cross-lingual Parameters Sharing in a Neural Network Parser. In *Proceeding of the 53rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 845–850, Beijing, China

Research process

After part-of-speech (POS) tagging, dependency parsing is the natural extension informing deeper layer of syntax. Working on dependency parsing is significantly harder compared with POS tagging. Instead of just outputting the tag label, we have to predict the tree-like structure of the sentence. Since we have successfully applied transfer learning for POS tagging taking advantage of small annotated corpus, the natural question is, can we do the same thing for dependency parsing. This paper forms a part of solving the second task about dependency parsing.

In this paper, I design and run experiments. Other co-authors which are my supervisors contribute ideas during our weekly meeting and participated in the paper writing.

Retrospective view

It is nice to see that the transfer learning technique through regularization terms is still applicable for dependency parsing task. Moreover, we only require the same POS annotation and dependency type between source and target language. This is a much better assumption compared with parallel data as used in EMNLP 2014 (§3.1.1) paper. Nevertheless, this paper still suffer similar drawback with the assumption of small annotated corpus in the target language. However, since annotating dependency treebank is much more costly and time consuming than POS annotation, applying our technique instead of annotating more data, is more compelling.

Low Resource Dependency Parsing: Cross-lingual Parameter Sharing in a Neural Network Parser

Long Duong,[12] **Trevor Cohn,**[1] **Steven Bird,**[1] and **Paul Cook**[3]
[1]Department of Computing and Information Systems, University of Melbourne
[2]National ICT Australia, Victoria Research Laboratory
[3]Faculty of Computer Science, University of New Brunswick
lduong@student.unimelb.edu.au {t.cohn,sbird}@unimelb.edu.au paul.cook@unb.ca

Abstract

Training a high-accuracy dependency parser requires a large treebank. However, these are costly and time-consuming to build. We propose a learning method that needs less data, based on the observation that there are underlying shared structures across languages. We exploit cues from a different source language in order to guide the learning process. Our model saves at least half of the annotation effort to reach the same accuracy compared with using the purely supervised method.

1 Introduction

Dependency parsing is a crucial component of many natural language processing systems, for tasks such as text classification (Özgür and Güngör, 2010), statistical machine translation (Xu et al., 2009), relation extraction (Bunescu and Mooney, 2005), and question answering (Cui et al., 2005). Supervised approaches to dependency parsing have been successful for languages where relatively large treebanks are available (McDonald et al., 2005). However, for many languages, annotated treebanks are not available. They are costly to create, requiring careful design, testing and subsequent refinement of annotation guidelines, along with assessment and management of annotator quality (Böhmová et al., 2001). The Universal Treebank Annotation Guidelines aim at providing unified annotation for many languages enabling cross-lingual comparison (Nivre et al., 2015). This project provides a starting point for developing a treebank for resource-poor languages. However, a mature parser requires a large treebank for training, and this is still extremely costly to create. Instead, we present a method that exploits shared structure across languages to achieve a more accurate parser. Structural information from the source resource-rich language is incorporated as a prior in the supervised training of a resource-poor target language parser using a small treebank. When compared with a supervised model, the gain is as high as 8.7%[1] on average when trained on just 1,000 tokens. As we add more training data, the gains persist, though they are more modest. Even at 15,000 tokens we observe a 2.9% improvement.

There are two main approaches for building dependency parsers for resource-poor languages: delexicalized parsing and projection (Täckström et al., 2013). The delexicalized approach was proposed by Zeman et al. (2008). A parser is built without any lexical features, and trained on a treebank in a resource-rich source language. It is then applied directly to parse sentences in the target resource-poor languages. Delexicalized parsing relies on the fact that identical part-of-speech (POS) inventories are highly informative of dependency relations, enough to make up for cross-lingual syntactic divergence.

In contrast, projection approaches use parallel data to project source language dependency relations to the target language (Hwa et al., 2005). McDonald et al. (2011) and Ma and Xia (2014) exploit both delexicalized parsing and parallel data. They use parallel data to constrain the model which is usually initialized by the English delexicalized parser.

In summary, existing work generally starts with a delexicalized parser and uses parallel data to improve it. In this paper, we start with a source language parser and refine it with help from dependency annotations instead of parallel data. This choice means our method can be applied in cases where linguists are dependency-annotating small amounts of field data, such as in Karuk, a nearly-extinct language of Northwest California (Garrett et al., 2013).

[1]We use absolute values herein.

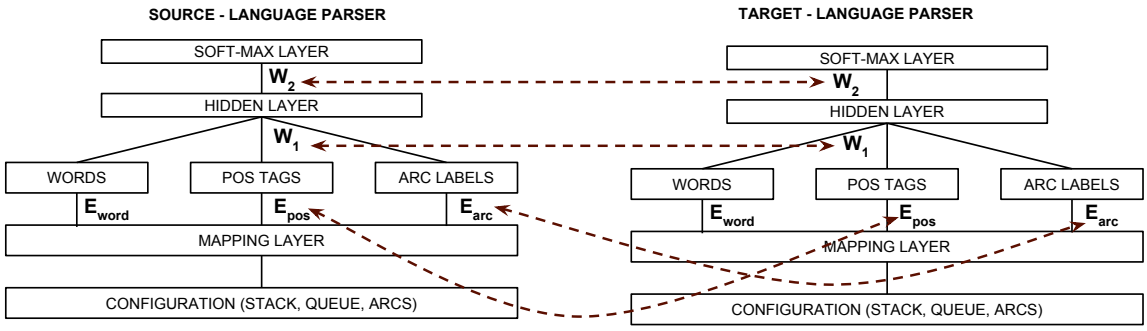


Figure 1: Neural Network Parser Architecture from Chen and Manning (2014) (left). Our model (left and right) with soft parameter sharing between the source and target language shown with dashed lines.

2 Supervised Neural Network Parser

In this section we review the parsing model which we use for both the source language and target language parsers. It is based on the work of Chen and Manning (2014). This parser can take advantage of target language monolingual data through word embeddings, data which is usually available for resource-poor languages. Chen and Manning's parser also achieved state-of-the-art monolingual parsing performance. They built a transition-based dependency parser (Nivre, 2006) using a neural-network. The neural network classifier decides which transition is applied for each configuration.

The architecture of the parser is illustrated in Figure 1 (left), where each layer is fully connected to the layer above. For each configuration, the selected list of words, POS tags and labels from the Stack, Queue and Arcs are extracted. Each word, POS or label is mapped to a low-dimension vector representation (embedding) through the Mapping Layer. This layer simply concatenates the embeddings which are then fed into a two-layer neural network classifier to predict the next parsing action. The set of parameters for the model is $E_{word}, E_{pos}, E_{labels}$ for the mapping layer, W_1 for the cubic hidden layer and W_2 for the softmax output layer.

3 Cross-lingual parser

Our model takes advantage of underlying structure shared between languages. Given the source language parsing structure as in Figure 1 (left), the set of parameters E_{word} will be different for the target language parser shown in Figure 1 (right) but we hypothesize that E_{pos}, E_{arc}, W_1 and W_2 can be shared as indicated with dashed lines. In particular we expect this to be the case when languages use the same POS tagset and arc label sets, as we presume herein. This assumption is motivated by the development of unified annotation for many languages (Nivre et al., 2015; Petrov et al., 2012; McDonald et al., 2013).

To allow parameter sharing between languages we could jointly train the parser on the source and target language simultaneously. However, we leave this for future work. Here we take an alternative approach, namely regularization in a similar vein to Duong et al. (2014). First we train a lexicalized neural network parser on the source resource-rich language (English), as described in Section 2. The learned parameters are $E^{en}_{word}, E^{en}_{pos}, E^{en}_{arc}, W^{en}_1, W^{en}_2$. Second, we incorporate English parameters as a prior for the target language training. This is straightforward when we use the same architecture, such as a neural network parser, for the target language. All we need to do is modify the learning objective function so that it includes the regularization part. However, we don't want to regularize the part related to E^{en}_{word} since it will be very different between source and target language. Letting $W_1 = (W_1^{word}, W_1^{pos}, W_1^{arc})$, the learning objective over training data $\mathcal{D} = \{x^{(i)}, y^{(i)}\}_{i=1}^N$, becomes:[2]

$$\mathcal{L} = \sum_{i=1}^{N} \log P(y^{(i)}|x^{(i)}) - \frac{\lambda_1}{2}\Big[\|W_1^{pos} - W_1^{en:pos}\|_F^2 + \|W_1^{arc} - W_1^{en:arc}\|_F^2 + \|W_2 - W_2^{en}\|_F^2\Big] - \frac{\lambda_2}{2}\Big[\|E_{pos} - E^{en}_{pos}\|_F^2 + \|E_{arc} - E^{en}_{arc}\|_F^2\Big] \quad (1)$$

This is applicable where we use the same POS

[2] All other parameters, i.e. W_1^{word} and E_{word}, are regularized using a zero-mean Gaussian regularization term, with weight $\lambda = 10^{-8}$, as was done in the original paper.

	Train	Dev	Test	Total
cs	1173.3	159.3	173.9	1506.5
de	269.6	12.4	16.6	298.6
en	204.6	25.1	25.1	254.8
es	382.4	41.7	8.5	432.6
fi	162.7	9.2	9.1	181.0
fr	354.7	38.9	7.1	400.7
ga	16.7	3.2	3.8	23.7
hu	20.8	3.0	2.7	26.5
it	194.1	10.5	10.2	214.8
sv	66.6	9.8	20.4	96.8

Table 1: Number of tokens ($\times$ 1,000) for each language in the Universal Dependency Treebank collection.

tagset and arc label annotation for the source and target language. The same POS tagset is required so that the source language parser has similar structure with the target language parser. The requirement of same arc label annotation is mainly needed for evaluation using the Labelled Attachment Score (LAS).[3] We fit two separate regularization sensitivity parameters, λ_1 and λ_2, since they correspond to different parts of the model. λ_1 is used for the shared (universal) part, while λ_2 is used for the language specific parts. Together λ_1 and λ_2 control the contribution of the source language parser towards the target resource-poor model. In the extreme case where λ_1 and λ_2 are large, the target model parameters are tied to the source model, except for the word embeddings E_{word}. In the opposite case, where they are small, the target language parser is similar to the purely supervised model. We expect that the best values fall between these extremes. We use stochastic gradient descent to optimize this objective function with respect to $W_1, W_2, E_{word}, E_{pos}, E_{arc}$.

4 Experiments

In this part we want to see how much our cross-lingual model helps to improve the supervised model, for various data sizes.

4.1 Dataset

We experimented with the Universal Dependency Treebank collection V1.0 (Nivre et al., 2015) which contains treebanks for 10 languages.[4] These treebanks have many desirable properties for our model: the dependency types and coarse POS are the same across languages. This removes the need for mapping the source and target language tagsets to a common tagset. Moreover, the dependency types are also common across languages allowing LAS evaluation. Table 1 shows the dataset size of each language in the collection. Some languages have over $400k$ tokens such as *cs, fr* and *es*, meanwhile, *hu* and *ga* have only around $25k$ tokens.

4.2 Monolingual Word Embeddings

We initialize the target language word embeddings E_{word} of our neural network cross-lingual model with pre-trained embeddings. This is an advantage since we can incorporate monolingual data which is usually available for resource-poor languages. We collect monolingual data for each language from the Machine Translation Workshop (WMT) data,[5] Europarl (Koehn, 2005) and EU Bookshop Corpus (Skadiņš et al., 2014). The size of monolingual data also varies significantly. There are languages such as English and German with more than 400 million words, whereas, Irish only has 4 million. We use the skip-gram model from `word2vec` to induce 50-dimension word embeddings (Mikolov et al., 2013).

4.3 Coarse vs Fine-Grain POS

Our model uses the source language parser as the prior for the target language parser. The requirement is that the source and target should use the same POS tagset. It is clear that information will be lost when using the coarser shared-POS tagset. Here, we simply want to quantify this loss. We run the supervised neural network parser on the coarse-grained Universal POS (UPOS) tagset, and the language-specific fine-grained POS tagset for languages where both are available in the Universal Dependency Treebank.[6] Table 2 shows the average LAS for coarse- and fine-grained POS tagsets with various data sizes. For the smaller dataset, using the coarse-grained POS tagset performed better. Even when we used all the data, the coarse-grained POS tagset still performed reasonably well, approaching the performance obtained using the fine-grained POS tagset. Thus, the choice of the coarse-grained Universal POS tagset

[3] However, same arc-label set also informs some information about the structure.

[4] Czech (cs), German (de), English (en), Spanish (es), Finnish (fi), French (fr), Irish (ga), Hungarian (hu), Italian (it), Swedish (sv)

[5] `http://www.statmt.org/wmt14/`

[6] Czech, English, Finnish, Irish, Italian, and Swedish

Tokens	Coarse UPOS	Fine POS
1k	46.8	42.3
3k	54.3	52.4
5k	56.9	55.8
10k	59.9	59.8
15k	61.5	61.4
All	74.7	75.2

Table 2: Average LAS for supervised learning using the modified version of the Universal POS tagset and the fine-grained POS tagset across various training data sizes.

instead of the original POS tagset is relevant, given that we assume there will only be a small treebank in the target language. Moreover, even when we have a bigger treebank, using the UPOS tagset does not hurt the performance much.[7]

4.4 Tuning regularization sensitivity

As shown in equation 1, λ_1 and λ_2 control the contribution of the source language parser toward the target language parser. We tune these parameters separately using development data. Firstly, we tune λ_1 by fixing $\lambda_2 = 0.1$. The reason for choosing such a large value of 0.1 is that we expect the POS and arc label embeddings to be fairly similar across languages. Figure 2 shows the average LAS for all 9 languages (except English) on different data sizes using different values of λ_1. We observed that $\lambda_1 = 0.001$ gives the optimum value on the development data consistently across different data sizes. We compare the performance at two extreme values of λ_1. For small data size, at 1k tokens, $\lambda_1 = 100$ is better than when $\lambda_1 = 10^{-8}$. This shows that when trained using a small data set, the source language parser is more accurate than the supervised model. However, at 3k tokens, the supervised model is starting to perform better.

We now fix $\lambda_1 = 0.001$ to tune λ_2 in the same range as λ_1. However, the average LAS didn't change much for different values of λ_2. It appears that λ_2 has very little effect on parsing accuracy. This is understandable since λ_2 affects only a small number of parameters (POS and arc embeddings). Thus, we choose $\lambda_1 = 0.001$ and $\lambda_2 = 0.1$ for our experiments.

[7]This is because UPOS generalizes better, and when aggregating with lexical information, it has similar distinguishing power compared with the fine-grained POS tagset.

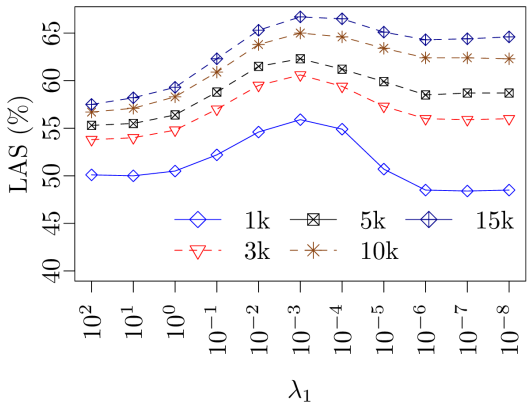


Figure 2: Sensitivity of regularization parameter λ_1 against the average LAS measured on all 9 languages (except English) on the development set for various data sizes (tokens)

4.5 Learning Curve

We choose English as our source language to build different target parsers for each language in the Universal Dependency Treebank collection. We train the supervised neural network parser as mentioned in Section 2 on the Universal Dependency English treebank using UPOS tagset. The UAS and LAS for the English parser is 85.2% and 82.9% respectively, when evaluated on the English test set. We use the English parser as the prior for our cross-lingual model, as described in Section 3. Figure 3 shows the learning curve for both the supervised neural network parser and our cross-lingual model with respect to our implemention of McDonald et al.'s (2011) delexicalized parser, i.e. their basic model which uses no parallel data and no target language supervision. Overall, both the supervised model and the cross-lingual model are much better than this baseline. For small data sizes, our cross-lingual model is superior when compared with the supervised model, giving as much as an 8.7% improvement. This improvement lessens as the size of training data increases. This is to be expected, because the supervised model becomes stronger as the size of training data increases, while the contribution of the source language parser is reduced. However, at 15k tokens we still observed a 2.9% average improvement, demonstrating the robustness of our cross-lingual model. Using our model also reduced the standard deviation ranges on each data point from 12% to 7%.

Using our cross-lingual model can save the annotation effort that is required in order to reach

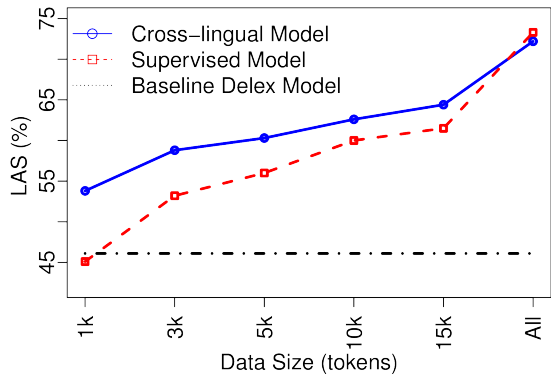


Figure 3: Learning curve for cross-lingual model and supervised model with respect to the baseline delexicalized parser from McDonald et al. (2011): the x-axis is the size of data (number of tokens); the y-axis is the average LAS measured on 9 languages (except English).

the same accuracy compared with the supervised model. For example, we only need 1k tokens in order to surpass the supervised model performance on 3k tokens, and we only need 5k tokens to match the supervised model trained on 10k tokens. The error rate reduction is from 15.8% down to 6.5% for training data sizes from 1k to 15k tokens. However, when we use all the training data, the supervised model is slightly better.

5 Conclusions

Thanks to the availability of the Universal Dependency Treebank, creating a treebank for a target resource-poor language has becoming easier. This fact motivates the work reported here, where we assume that only a tiny treebank is available in the target language. We tried to make the most out of the target language treebank by incorporating a source-language parser as a prior in learning a neural network parser. Our results show that we can achieve a more accurate parser using the same training data. In future work, we would like to investigate joint training on the source and target languages.

Acknowledgments

This work was supported by the University of Melbourne and National ICT Australia (NICTA). Trevor Cohn is the recipient of an Australian Research Council Future Fellowship (project number FT130101105).

References

Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. 2001. The Prague Dependency Treebank: A Three-Level Annotation Scenario. In Anne Abeillé, editor, *Treebanks: Building and Using Syntactically Annotated Corpora*, pages 103–127. Kluwer Academic Publishers.

Razvan C. Bunescu and Raymond J. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 724–731, Stroudsburg, PA, USA. ACL.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar. ACL.

Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. 2005. Question answering passage retrieval using dependency relations. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 400–407, New York, NY, USA. ACM.

Long Duong, Trevor Cohn, Karin Verspoor, Steven Bird, and Paul Cook. 2014. What can we get from 1000 tokens? a case study of multilingual pos tagging for resource-poor languages. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 886–897, Doha, Qatar. ACL.

Andrew Garrett, Clare Sandy, Erik Maier, Line Mikkelsen, and Patrick Davidson. 2013. Developing the Karuk Treebank. Fieldwork Forum, Department of Linguistics, UC Berkeley.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11:311–325.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*, pages 79–86, Phuket, Thailand.

Xuezhe Ma and Fei Xia. 2014. Unsupervised dependency parsing with transferring distribution via parallel guidance and entropy regularization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1337–1348. Association for Computational Linguistics.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 91–98, Stroudsburg, PA, USA. ACL.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 62–72.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria. ACL.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C.j.c. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Joakim Nivre, Cristina Bosco, Jinho Choi, Marie-Catherine de Marneffe, Timothy Dozat, Richárd Farkas, Jennifer Foster, Filip Ginter, Yoav Goldberg, Jan Hajič, Jenna Kanerva, Veronika Laippala, Alessandro Lenci, Teresa Lynn, Christopher Manning, Ryan McDonald, Anna Missilä, Simonetta Montemagni, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Maria Simi, Aaron Smith, Reut Tsarfaty, Veronika Vincze, and Daniel Zeman. 2015. Universal dependencies 1.0.

Joakim Nivre. 2006. *Inductive Dependency Parsing (Text, Speech and Language Technology)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Levent Özgür and Tunga Güngör. 2010. Text classification with the support of pruned dependency patterns. *Pattern Recognition Letters*, 31(12):1598–1607.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey.

Raivis Skadiņš, Jörg Tiedemann, Roberts Rozis, and Daiga Deksne. 2014. Billions of parallel words for free: Building and using the eu bookshop corpus. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC-2014)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Oscar Täckström, Ryan McDonald, and Joakim Nivre. 2013. Target language adaptation of discriminative transfer parsers. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1061–1071, Atlanta, Georgia. ACL.

Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. 2009. Using a dependency parser to improve smt for subject-object-verb languages. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 245–253, Boulder, Colorado. ACL.

Daniel Zeman, Univerzita Karlova, and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *In IJCNLP-08 Workshop on NLP for Less Privileged Languages*, pages 35–42.

3.1.3 EMNLP 2015

Long Duong, Trevor Cohn, Steven Bird, Paul Cook. 2015. A Neural Network Model for Low-Resource Universal Dependency Parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 339–348, Lisbon, Portugal.

Research process

In our ACL 2015 (§3.1.2) paper, we train the model in the cascade style where source language parser is trained first and then used as the prior for the target language parser. However, we might benefit more from jointly train the source and target language together as this allows better parameter sharing, which is the motivation for this paper.

In this paper, I design and run experiments. Other co-authors which are my supervisors contribute ideas during our weekly meeting and participate in the paper writing.

Retrospective view

As shown in the paper, joint training is substantially better than cascade training across various data size. Also, joint training is more flexible, we can even relax the requirement of the same POS or dependency type annotation imposed in cascade training in ACL 2015 (§3.1.2) paper. The model can automatically learn the annotation mapping between source and target language as part of the training. However, it is usually slower and more challenging to efficiently train the join model. Moreover, in this paper, we mainly compare with prior work using similar neural network transitional based parser. Despite the fact that the proposed joint model is generic and can apply to various architectures, how this work apply to or compare with prior work using different parsing architecture such as graph based, is unknown.

A Neural Network Model for Low-Resource Universal Dependency Parsing

Long Duong,[12] **Trevor Cohn,**[1] **Steven Bird,**[1] and **Paul Cook**[3]
[1]Department of Computing and Information Systems, University of Melbourne
[2]National ICT Australia, Victoria Research Laboratory
[3]Faculty of Computer Science, University of New Brunswick
lduong@student.unimelb.edu.au {t.cohn,sbird}@unimelb.edu.au paul.cook@unb.ca

Abstract

Accurate dependency parsing requires large treebanks, which are only available for a few languages. We propose a method that takes advantage of shared structure across languages to build a mature parser using less training data. We propose a model for learning a shared "universal" parser that operates over an interlingual continuous representation of language, along with language-specific mapping components. Compared with supervised learning, our methods give a consistent 8-10% improvement across several treebanks in low-resource simulations.

1 Introduction

Dependency parsing is an important task for Natural Language Processing (NLP) with application to text classification (Özgür and Güngör, 2010), relation extraction (Bunescu and Mooney, 2005), question answering (Cui et al., 2005), statistical machine translation (Xu et al., 2009), and sentiment analysis (Socher et al., 2013). A mature parser normally requires a large treebank for training, however such resources are rarely available and are costly to build. Ideally, we would be able to construct a high quality parser with less training data, thereby enabling accurate parsing for low-resource languages.

In this paper we formalize the dependency parsing task for a low-resource language as a domain adaptation task, in which a *target* resource-poor language treebank is treated as *in-domain*, while a much larger treebank in a high-resource language forms the *out-of-domain* data. In this way, we can apply well-understood domain adaptation techniques to the dependency parsing task. However, a crucial requirement for domain adaptation is that the in-domain and out-of-domain data have compatible representations. In applying our approach to data from several languages, we must learn such a cross-lingual representation. Here we frame this representation learning as part of a neural network training. The underlying hypothesis for the joint learning is that there are some shared-structures across languages that we can exploit. This hypothesis is motivated by the excellent results of the cross-lingual application of unlexicalised parsing (McDonald et al., 2011), whereby a delexicalized parser constructed on one language is applied directly to another language.

Our approach works by jointly training a neural network dependency parser to model the syntax in both a source and target language. Many of the parameters of the source and target language parsers are shared, except for a small handful of language-specific parameters. In this way, the information can flow back and forth between languages, allowing for the learning of a compatible cross-lingual syntactic representation, while also allowing for the parsers to mutually correct one another's errors. We include some language-specific components, in order to better model the lexicon of each language and allow learning of the syntactic idiosyncrasies of each language. Our experiments show that this outperforms a purely supervised setting, on both small and large data conditions, with a gain as high as 10% for small training sets. Our proposed joint training method also out-performs the conventional cascade approach where the parameters between source and target languages are related together through a regularization term (Duong et al., 2015).

Our model is flexible, allowing easy incorporation of peripheral information. For example, assuming the presence of a small bilingual dictionary is befitting of a low-resource setting as this is prototypically one of the first artifacts generated by field linguists. We incorporate a bilingual dictionary as a set of soft constraints on the

model, such that it learns similar representations for each word and its translation(s). For example, the representation of *house* in English should be close to *haus* in German. We empirically show that adding a bilingual dictionary improves parser performance, particularly when target data is limited.

The final contribution of the paper concerns the learned word embeddings. We demonstrate that these encode meaningful syntactic phenomena, both in terms of the observable clusters and through a verb classification task. The code for this paper is published as an open source project.[1]

2 Related Work

This work is motivated by the idea of delexicalized parsing, in which a parser is built without any lexical features and trained on a treebank for a resource-rich source language (Zeman et al., 2008). It is then applied directly to parse sentences in the target resource-poor languages. Delexicalized parsing relies on the fact that identical part-of-speech (POS) inventories are highly informative of dependency relations, and that there exists shared dependency structures across languages.

Building a dependency parser for a resource-poor language usually starts with the delexicalized parser and then uses other resources to refine the model. McDonald et al. (2011) and Ma and Xia (2014) exploited parallel data as the bridge to transfer constraints from the source resource-rich language to the target resource-poor languages. Täckström et al. (2012) also used parallel data to induce cross-lingual word clusters which added as features for their delexicalized parser. Durrett et al. (2012) constructed the set of language-independent features and used a bilingual dictionary as the bridge to transfer these features from source to target language. Täckström et al. (2013) additionally used high-level linguistic features extracted from the World Atlas of Language Structures (WALS) (Dryer and Haspelmath, 2013).

For low-resource languages, no large parallel corpus is available. Some linguists are dependency-annotating small amounts of field data, e.g. for Karuk, a nearly-extinct language of Northwest California (Garrett et al., 2013). Accordingly, we adopt a different resource requirement: a small treebank in the target low-resource language.

Domain adaptation or joint-training is a different branch of research, and falls outside the scope of this paper. Nevertheless, we would like to contrast our work with Senna (Collobert et al., 2011), a neural network framework to perform a variety of NLP tasks such as part-of-speech (POS) tagging, named entity recognition (NER), chunking, and so forth. Both approaches exploit common linguistic properties of the data through joint learning. However, Collobert et al's target is to find a single input representation that can work well for many tasks. Our goal is different: we allow the joint-training inputs to be different but constrain the parameter weights in the upper layer to be identical. Consequently, our method applies to the task where inputs are different, possibly from different languages or domains. Their method applies for different tasks in the same language/domain where the inputs are fairly similar.

2.1 Supervised Neural Network Parser

This section describes the monolingual neural network dependency parser structure of Chen and Manning (2014). This parser achieves excellent performance, and has a highly flexible formulation allowing auxilliary inputs. The model is based on a transition-based dependency parser (Nivre, 2006) formulated as a neural-network classifier to decide which transition to apply to each parsing state configuration.[2] That is, for each configuration, the selected list of words, POS tags and labels from the Stack, Queue and Arcs are extracted. Each word, POS and label is mapped into a low-dimension vector representation using an embedding matrix, which is then fed into a two-layer neural network classifier to predict the next parsing action. The set of parameters for the model is $E = \{E^{word}, E^{\text{pos}}, E^{\text{arc}}\}$ for the embedding layer, W_1 for the fully connected cubic hidden layer and W_2 for the softmax output layer. The model prediction function is

$$P(Y|X = \vec{x}, W_1, W_2, E) = \text{softmax}\Big(W_2 \times \text{cube}(W_1 \times \Phi\left[\vec{x}, E\right])\Big) \quad (1)$$

[1] http://github.com/longdt219/universal_dependency_parser

[2] Our approach is focused on a technique for transfer learning which can be more widely applied to other types of dependency parser (and models, generally) regardless of whether they are transition-based or graph-based.

where cube is a non-linear activation function, Φ is the embedding function that returns a vector representation of parsing state x using an embedding matrix E. We refer the reader to Chen and Manning (2014) for a more detailed description.

3 A Joint Interlingual Model

We assume a small treebank in a target resource-poor language, as well as a larger treebank in the source langugage. Our objective is to learn a model of both languages subject to the constraint that both models are similar overall, while allowing for some limited language variability. Instead of just training two different parsers on source and then on target, we train them jointly, in order to learn an interlingual parser. This allows the method to take maximum advantage of the limited treebank data available, resulting in highly accurate predicted parses.

Training a monolingual parser as described in section 2.1 requires optimizing the a simple cross-entropy learning objective, $\mathcal{L} = -\sum_{i=1}^{|D|} \log P(Y = \vec{y}^{(i)}|X = \vec{x}^{(i)})$, where $P(Y|X)$ is given by the equation 1 and $D = \{\vec{x}^{(i)}, \vec{y}^{(i)}\}_{i=1}^{n}$ is the training data. Joint training of a parser over the source and target languages can be achieved by simply adding two such cross-entropy objectives, i.e.,

$$\mathcal{L}_{\text{joint}} = -\sum_{i=1}^{|D_s|} \log P(Y_s = \vec{y}_s^{(i)}|X_s = \vec{x}_s^{(i)}) - \sum_{i=1}^{|D_t|} \log P(Y_t = \vec{y}_t^{(i)}|X_t = \vec{x}_t^{(i)}) \,, \quad (2)$$

where the training data, $D = D_s \cup D_t$, comprises data in both the source and target language. However training the model according to equation 2 will result in two independent parsers. To enforce similarity between the two parsers, we adopt parameter sharing: the neural network parameters, W_1 and W_2, are identical in both parsers. Thereby

$$P(Y_\alpha|X_\alpha = \vec{x}) = P(Y|X = \vec{x}, W_1, W_2, E_\alpha) \,,$$

where the subscript $\alpha \in \{s, t\}$ denotes the source or target language. We allow the embedding matrix E_α to differ in order to accommodate language-specific features, in terms of the representations of lexical types, E_s^{word}, part-of-speech, E_s^{pos} and dependency arc labels E_s^{arc}. This reflects the fact that different languages have different lexicon, parts-of-speech often exhibit different roles, and dependency edges serve different functions, e.g. in Korean a *static verb* can serve as an *adjective* (Kim, 2001). During training, the language-specific errors are back propagated through different branches according to the language, guiding learning towards an interlingual representation that informs parsing decisions in both languages. The set of parameters for the model is W_1, W_2, E_s, E_t where E_s, E_t are the embedding matrices for the source and target languages.

Generally speaking, we can understand the model as building the universal dependency parser that parses the universal language. Specifically, the model is the combination of two parts. The universal part (W_1, W_2) that is shared between the languages, and the conversion parts (E_s, E_t) that map a language-specific representation into the universal language. Naturally, we could stack several non-linear layers in the conversion components such that the model can better transform the input into the universal representation; we leave this exploration for future work. Currently, our cross-lingual word embeddings are meaningful for a pair of source and target languages. However, our model can easily be used for joint training over $k > 2$ languages. We also leave this avenue of enquiry for future work

One concern from equation 2 is that when the source language treebank D_s is much bigger than target language treebank D_t, it is likely to dominate, and consequently, learning will mainly focus on optimizing the source language parser. We adjust for this disparity by balancing the two datasets, D_s and D_t, during training. When selecting mini-batches for online gradient updates, we select an equal number of classification instances from the source and target languages. Thus, for each step $|D_s| = |D_t|$, effectively reweighting the cross-entropy components in (2) to ensure parity between the languages.

The other concern is over-fitting, especially when we only have a small treebank in the target language. As suggested in Chen and Manning (2014), we apply dropout, a form of regularization for both source and target language. That is, we randomly drop some of the activation units from both hidden layer and input layer. Following Srivastava et al. (2014), we randomly dropout 20% of the input layer and 50% of the hidden layer. Em-

pirically, we observed a substantial improvement applying dropout to the model over MLE or l_2 regularization.

3.1 Incorporating a Dictionary

Our model is flexible, enabling us to freely add additional components. In this section, we assume the presence of a bilingual dictionary between the source and target language. We seek to incorporate this dictionary as a part of model learning, to encode the intuition that if two lexical items are translation of one another, the parser should treat them similarly.[3] Recall that the mapping layer is the combination of word, pos and arc embedding, i.e., $E_\alpha = \{E_\alpha^{\text{word}}, E_\alpha^{\text{pos}}, E_\alpha^{\text{arc}}\}$. We can easily add bilingual dictionary constraints to the model in the form of regularization to minimize the l_2 distance between word representations, i.e., $\sum_{\langle i,j\rangle \in \mathcal{D}} \|E_s^{\text{word}(i)} - E_t^{\text{word}(j)}\|_F^2$, where $\mathcal{D}$ comprises translation pairs, word(i) and word(j).

When the languages share the same POS tagset and Arc set,[4] we can also add further constraints such as their language-specific embeddings are close together. This results a regularised training objective,

$$\mathcal{L}_{\text{dict}} = \mathcal{L}_{\text{joint}} - \lambda \Bigg(\sum_{\langle i,j\rangle \in \mathcal{D}} \|E_s^{\text{word}(i)} - E_t^{\text{word}(j)}\|_F^2 + \|E_s^{\text{pos}} - E_t^{\text{pos}}\|_F^2 + \|E_s^{\text{arc}} - E_t^{\text{arc}}\|_F^2 \Bigg), \quad (3)$$

where $\lambda \in [0, \infty]$ controls to what degree we bind these words or pos tags or arc labels together, with high λ tying the parameters and small λ allowing independent learning. We expect the best value of λ to fall somewhere between these extremes. Finally, we use a mini-batch size of 1000 instance pairs and adaptive learning rate trainer, `adagrad` (Duchi et al., 2011) to build our two separate models corresponding to equations 2 and 3.

4 Experiments

In this section, we compare our joint training approach with baseline methods of supervised learning in the target language, and cascaded learning of source and target parsers.

[3] However, this is not always the case. For example, modal or auxiliary verbs in English often have no translations in different languages or map to words with different syntactic functions.

[4] As was the case for our experiments.

4.1 Dataset

We experiment with the Universal Dependency Treebank (UDT) V1.0 (Nivre et al., 2015), simulating low resource settings.[5] This treebank has many desirable properties for our model: the dependency types (arc labels set) and coarse POS tagset are the same across languages. This removes the need for mapping the source and target language tagsets to a common tagset. Moreover, the dependency types are also common across languages allowing evaluation of the labelled attachment score (LAS). The treebank covers 10 languages,[6] with some languages very highly resourced—Czech, French and Spanish have $400k$ tokens—and only modest amounts of data for other languages—Hungarian and Irish have only around $25k$ tokens. Cross-lingual models assume English as the source language, for which we have a large treebank, and only a small treebank of 3K tokens exists in each target language, simulated by subsampling the corpus.

4.2 Baseline Cascade Model

We compare our approach to a baseline interlingual model based on the same parsing algorithm as presented in section 2.1, but with cascaded training (Duong et al., 2015). This works by first learning the source language parser, and then training the target language parser using a regularization term to minimise the distance between the parameters of the target parser and the source parser (which is fixed). In this way, some structural information from the source parser can be used in the target parser, however it is likely that the representation will be overly biased towards the source language and consequently may not prove as useful for modelling the target.

4.3 Monolingual Word Embeddings

While the E^{pos} and E^{arc} are randomly initialized, we initialize both the source and target language word embeddings $E_s^{\text{word}}, E_t^{\text{word}}$ of our neural network models with pre-trained embeddings. This is an advantage since we can incorporate the monolingual data which is often available, even for

[5] Evaluating on truly resource-poor languages would be preferable to simulation. However for ease of training and evaluation, which requires a small treebank in the target language, we simulate the low-resource setting using a small part of the UDT.

[6] Czech (cs), English (en), Finnish (fi), French (fr), German (de), Hungarian (hu), Irish (ga), Italian (it), Spanish (es), Swedish (sv).

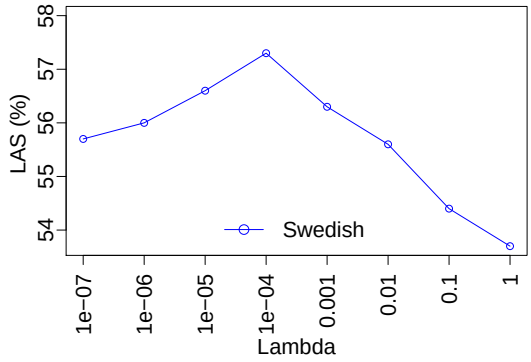


Figure 1: Sensitivity of regularization parameter λ against the LAS measured on the Swedish development set trained on 1000 (tokens).

resource-poor languages. We collect monolingual data for each language from the Machine Translation Workshop (WMT) data,[7] Europarl (Koehn, 2005) and EU Bookshop Corpus (Skadiņš et al., 2014). The size of monolingual data also varies significantly, with as much as 400 million tokens for English and German, and as few as 4 million tokens for Irish. We use the skip-gram model (Mikolov et al., 2013b) to induce 50-dimensional word embeddings.

4.4 Bilingual Dictionary

For the extended model as described in section 3.1, we also need a bilingual dictionary. We extract dictionaries from PanLex (Kamholz et al., 2014) which currently covers around 1300 language varieties and about 12 million expressions. This dataset is growing and aims at covering all languages in the world and up to 350 million expressions. The translations in PanLex come from various sources such as glossaries, dictionaries, automatic inference from other languages, etc. Naturally, the bilingual dictionary size varies greatly among resource-poor and resource-rich languages.

4.5 Regularization Parameter Tuning

Joint training with a dictionary (see equation 3) includes a regularization sensitivity parameter λ. This parameter controls to what extent we should bind the source words and their target translation, common POS tags and arcs together. In this section we measure the sensitivity of our approach with respect to this parameter. In a real world scenario, getting development data to tune this parameter is difficult. Thus, we want a parameter that can work well cross-lingually. To simulate this, we only tune the parameter on one language and apply it directly to different languages. We trained on a small Swedish treebank with 1k tokens, testing several different values of λ. We evaluated on the Swedish development dataset. Figure 1 shows the labelled attachment score (LAS) for different λ. It's clearly visible that $\lambda = 0.0001$ gives the maximum LAS on the development set. Thus, we use this value for all the experiments involving a dictionary hereafter.

4.6 Results

For our initial experiments we assume that we have only a small target treebank with 3000 tokens (around 200 sentences). Ideally the much larger source language (English) treebank should be able to improve parser performance versus simple supervised learning on such a small collection. We apply the joint model (equation 2) and joint model with the dictionary constraints (equation 3) for each target language,

The results are reported in Table 1. The supervised neural network dependency parser performed worst, as expected, and the baseline cascade model consistently outperformed the supervised model on all languages by an average margin of 5.6% (absolute).[8] The joint model also consistently out-performed both baselines giving a further 1.9% average improvement over the cascade. This was despite the fact that the cascaded model had the benefit of tuning for the regularization parameters on a development corpus, while the joint model had no parameter tuning. Note that the improvement varies substantially across languages, and is largest for Czech but is only minor for Swedish. The joint model with the bilingual dictionary outperforms the joint model, however, the improvement is modest (0.7%).Nevertheless, this model gives substantial improvements compared with the cascaded and the supervised model (2.6% and 8.2%).

5 Analysis

5.1 Learning Curve

In section 4.6, we used a 3k token treebank in the target language. What if we have more or less target language data? Figure 2 shows the learning

[7] `http://www.statmt.org/wmt14/`

[8] We use absolute percentage comparisons herein.

	cs	de	es	fi	fr	ga	hu	it	sv	μ
Supervised	43.1	47.3	60.3	46.4	56.2	59.4	48.4	65.4	52.6	53.2
Baseline Cascaded	49.6	59.2	66.4	49.5	63.2	59.5	50.5	69.9	61.4	58.8
Joint	55.2	61.2	69.1	51.4	65.3	60.6	51.2	71.2	61.4	60.7
Joint + Dict	55.7	61.8	70.5	51.5	67.2	61.1	51.0	71.3	62.5	61.4

Table 1: Labelled attachment score (LAS) for each model type trained on 3000 tokens for each target language (columns). All bar the supervised model also use a large English treebank.

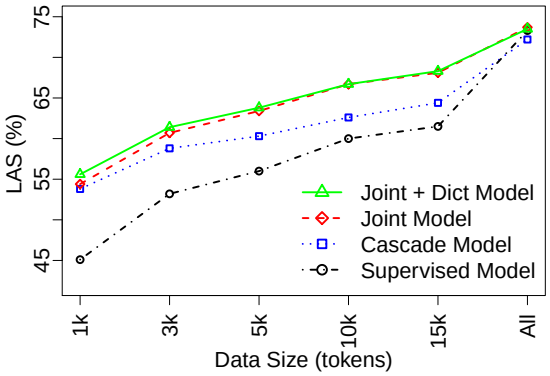


Figure 2: Learning curve for Joint model, Joint + Dict model, Baseline cascaded and Supervised model: the x-axis is the size of data (number of tokens); the y-axis is the average LAS measured on 9 languages (except English).

curve with respect to various models on different data sizes averaged over all target languages. For small datasets of 1k training tokens, the cascaded model, joint model and joint + dict model performed similarly well, out-performing the supervised model by about 10% (absolute). With more training data, we see interesting changes to the relative performance of the different models. While the baseline cascade model still outperforms the supervised model, the improvement is diminishing, and by 15k, the difference is only 2.9%. On the other hand, compared with supervised model, the joint and joint + dict models perform consistently well at all sizes, maintaining an 8% lead at 15k. This shows the superiority of joint training compared with single language training.

To understand this pattern of performance differences for the cascade versus the joint model, one needs to consider the cascade model formulation. In this approach, the target language parameters are tied (softly) with the source language parameters through regularization. This is a benefit for small datasets, providing a smoothing function to limit overtraining. However, when we have more training data, these constraints limit the capacity of the model to describe the target data. This is compounded by the problem that the source representation may not be appropriate for modelling the target language, and there is no way to correct for this. In contrast the joint model learns a mutually compatible representation automatically during joint training.

The performance results for the joint model with and without the dictionary are overall similar. Only on small datasets (1k,3k), is the difference notable. From 5k tokens, the bilingual dictionary doesn't confer additional information, presumably as there is sufficient data for learning syntactic word representations. Moreover, translation entries exist between syntactically related word types as well as semantically related pairs, with the latter potentially limiting the beneficial effect of the dictionary.

When training on all the target language data, the supervised model does well, surpassing the cascade model. Surprisingly, the joint models outperform slightly, yielding a 0.4% improvement. This is an interesting observation suggesting that our method has potential for use not only for low resource problems, but also high resource settings.

5.2 Different Tagsets

In the above experiments, we used the universal POS tagset for all the languages in the corpus. However, for some languages,[9] the UDT also provides language specific POS tags. We use this data to test the relative performance of the model using a universal tagset cf. language specific tagsets. In this experiment, we applied the same joint model (see §3) but with a language specific tagset instead of UPOS for these languages. We expect the joint model to automatically learn to project the different tagsets into a common space, i.e., implicitly

[9] en, cs, fi, ga, it and sv.

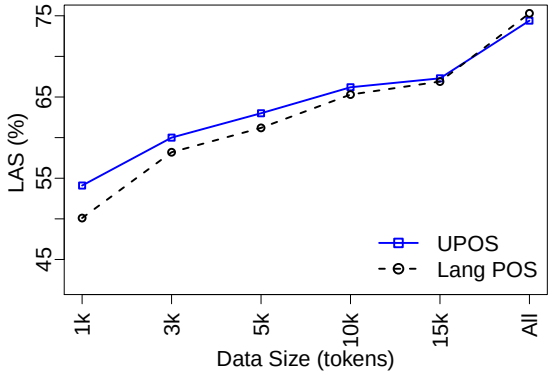


Figure 3: Learning curve for joint model using the UPOS tagset or language specific POS tagset: the x-axis is the size of data (number of tokens); the y-axis is the average LAS measured on 5 languages (except English).

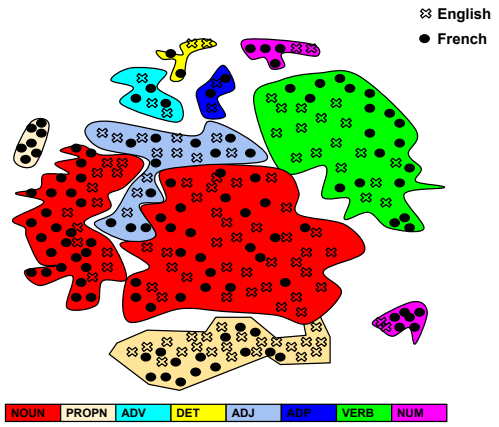


Figure 4: Universal Language visualization according to language and POS. (This should be viewed in colour.)

learn a tagset mapping between languages. Figure 3 shows the learning curve comparing the joint model with the two types of POS tagsets. For small dataset, it is clear that the data is insufficient for the model to learn a good tagset mapping, especially for a morphologically rich language like Czech. However, with more data, the model is better able to learn the tagset mapping as part of joint training. Beyond 15k tokens, the joint model using language specific POS tagset out-performs UPOS. Clearly there is some information lost in the UPOS tagset, although at the same time the UPOS mapping provides implicit linguistic supervision. This explains why the UPOS might be useful in small data scenarios, but detrimental at scale. Using all the target data ("All") the language specific POS provides a 1% (absolute) gain over UPOS.

5.3 Universal Representation

As described in section 3, we can consider our joint model as the combination of two parts: a universal parser and a language-specific embedding E_s or E_t that converts the source and target language into the universal representation. We now seek to analyse qualitatively this universal representation through visualization. For this purpose we use a joint model of English and French, using all the available French treebank (more than 350k tokens) as well as a bilingual dictionary.[10] Figure 4 shows the t-SNE (Van Der Maaten, 2014) projection of the 50 dimensional word embeddings in both languages. We can see that English and French are mixed nicely together. The colouring denotes the POS tag, showing clearly that the words with similar POS tags are grouped together regardless of languages. This is partially understandable since word embeddings for dependency parsing need to convey the dependency context rather than surrounding words, as in most distributional embedding models. Words having similar dependency relation should be grouped together as they are treated similarly by the parser.

Some of the learned cross-lingual word-embeddings are shown in Table 2, which includes the five nearest neighbours to selected English words according to the monolingual word embedding (section 4.3) and our cross-lingual dependency word embeddings, trained using PanLex. The monolingual sets appear to be strongly characterised by distributional similarity. The cross-lingual embeddings display greater semantic similarity, while being more variable morphosyntactically. In many cases, the top five words of English and French are translations of each other, but with varying inflectional endings in the French forms. For example, "*buy*" vs "*vendez*" or "*invest*" vs "*investir*". This is a direct consequence of incorporating the bilingual lexicon. Moreover, the top five closest words of both English and French mostly have the same part of speech. This is consistent with the finding in Figure 4.

[10] We also visualized the cross-lingual word embeddings without the dictionary, however the results were rather odd. Although we saw coherent POS clusters, the two languages were largely disjoint. We speculate that many components of the embeddings are use for only one language, and these outnumber the shared components, and thus more careful projection is needed for meaningful visualisation.

Words	Mono	Cross lingual embedding En	Cross lingual embedding Fr
sell	buy	buy	revendre
	eat	invest	vendez
	produce	integrate	acheter
	compete	guide	achètent
	burn	eat	investir
playing	serving	sailing	jouait
	acting	play	navigue
	paying	moving	jouent
	pursuing	faces	pièce
	running	ran	jouer
hard	difficult	crazy	dur
	harder	strange	dures
	easy	beautiful	hard
	magnificent	friendly	fou
	painful	difficult	folles
initially	originally	originally	réellement
	previously	previously	déjà
	officially	officially	récemment
	basically	actually	dernièrement
	already	already	surroît
university	teachers	school	universitaire
	student	education	université
	teacher	student	école
	student	medicine	scolaire
	training	participant	school
mobile	wireless	computers	mobile
	goods	Web	mobiles
	online	Internet	ordinateurs
	freight	computer	Web
	broadband	web	internet

Table 2: Examples of 5 nearest neighbours with the target English word using the original monolingual word embedding and our cross-lingual dependency based word embedding.

Levin (1993) has shown that there is a strong connection between a verb's meaning and its syntactic behaviour. We compare the English side of our cross-lingual dependency based word embeddings with various other pre-trained monolingual English word embeddings and our monolingual embedding (section 4.3) on Verb-143 dataset (Baker et al., 2014). This dataset contains 143 pairs of verbs that are manually given score from 1 to 10 according to the meaning similarity. Table 3 shows the Pearson correlation with human judgment for our embeddings and other pre-trained embeddings. As expected, our cross-lingual embeddings out-perform others embeddings on this dataset. This is partly because the syntactic behaviour is well encoded in our word embeddings through dependency relation.

	Correlation
Senna (Collobert et al., 2011)	0.36
Skip-gram (Mikolov et al., 2013a)	0.27
RNN (Mikolov et al., 2011)	0.31
Our monolingual embedding	0.39
Our crosslingual embedding	0.44

Table 3: Compare the English side of our cross-lingual embeddings with various other embeddings evaluated on Verb-143 dataset (Baker et al., 2014). We directly use the pre-trained models from corresponding papers.

Our embeddings encode not just cross-lingual correspondences, but also capture dependency relations which we expect might be beneficial for other NLP tasks based on dependency parsing, e.g., cross-lingual semantic role labelling where long-distance relationship can be captured by word embedding.

6 Conclusion

In this paper, we present a training method for building a dependency parser for a resource-poor language using a larger treebank in a high-resource language. Our approach takes advantage of the shared structure among languages to learn a universal parser and language-specific mappings to the lexicon, parts of speech and dependency arcs. Compared with supervised learning, our joint model gives a consistent 8-10% improvement over several different datasets in simulation low-resource scenarios. Interestingly, some small but consistent gains are still realised by joint cross-lingual training even on large complete treebanks. This suggests that our approach has utility not just in low resource settings. Our joint model is flexible, allowing the incorporation of a bilingual dictionary, which results in small improvements particularly for tiny training scenarios.

As the side-effect of training our joint model, we obtain cross-lingual word embeddings specialized for dependency parsing. We expect these embeddings to be beneficial to other syntatic and semantic tasks. In future work, we plan to extend joint training to several languages, and further explore the idea of learning and exploiting cross-lingual embeddings.

Acknowledgments

This work was supported by the University of Melbourne and National ICT Australia (NICTA). Trevor Cohn is the recipient of an Australian Research Council Future Fellowship (project number FT130101105).

References

Simon Baker, Roi Reichart, and Anna Korhonen. 2014. An unsupervised model for instance level subcategorization acquisition. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 278–289.

Razvan C. Bunescu and Raymond J. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 724–731, Stroudsburg, PA, USA. Association for Computational Linguistics.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar, October. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, November.

Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. 2005. Question answering passage retrieval using dependency relations. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 400–407, New York, NY, USA. ACM.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, July.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 845–850, Beijing, China, July. Association for Computational Linguistics.

Greg Durrett, Adam Pauls, and Dan Klein. 2012. Syntactic transfer using a bilingual lexicon. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1–11, Stroudsburg, PA, USA. Association for Computational Linguistics.

Andrew Garrett, Clare Sandy, Erik Maier, Line Mikkelsen, and Patrick Davidson. 2013. Developing the Karuk Treebank. Fieldwork Forum, Department of Linguistics, UC Berkeley.

David Kamholz, Jonathan Pool, and Susan Colowick. 2014. Panlex: Building a resource for panlingual lexical translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3145–50, Reykjavik, Iceland. European Language Resources Association (ELRA).

Min-joo Kim. 2001. Does korean have adjectives. In *MIT Working Papers 43. Proceedings of HUMIT 2001*, pages 71–89. MIT Working Papers.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*, pages 79–86, Phuket, Thailand.

B. Levin. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press.

Xuezhe Ma and Fei Xia. 2014. Unsupervised dependency parsing with transferring distribution via parallel guidance and entropy regularization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1337–1348. Association for Computational Linguistics.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 62–72.

Tomas Mikolov, Stefan Kombrink, Anoop Deoras, Lukar Burget, and Jan Honza Cernocky. 2011. Rnnlm – recurrent neural network language modeling toolkit. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, December.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Joakim Nivre, Cristina Bosco, Jinho Choi, Marie-Catherine de Marneffe, Timothy Dozat, Richárd Farkas, Jennifer Foster, Filip Ginter, Yoav Goldberg, Jan Hajič, Jenna Kanerva, Veronika Laippala, Alessandro Lenci, Teresa Lynn, Christopher Manning, Ryan McDonald, Anna Missilä, Simonetta Montemagni, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Maria Simi, Aaron Smith, Reut Tsarfaty, Veronika Vincze, and Daniel Zeman. 2015. Universal dependencies 1.0.

Joakim Nivre. 2006. *Inductive Dependency Parsing (Text, Speech and Language Technology)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Levent Özgür and Tunga Güngör. 2010. Text classification with the support of pruned dependency patterns. *Pattern Recogn. Lett.*, 31(12):1598–1607, September.

Raivis Skadiņš, Jörg Tiedemann, Roberts Rozis, and Daiga Deksne. 2014. Billions of parallel words for free: Building and using the eu bookshop corpus. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC-2014)*, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 477–487. Association for Computational Linguistics.

Oscar Täckström, Ryan McDonald, and Joakim Nivre. 2013. Target language adaptation of discriminative transfer parsers. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1061–1071, Atlanta, Georgia, June. Association for Computational Linguistics.

Laurens Van Der Maaten. 2014. Accelerating t-sne using tree-based algorithms. *J. Mach. Learn. Res.*, 15(1):3221–3245, January.

Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. 2009. Using a dependency parser to improve smt for subject-object-verb languages. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 245–253, Boulder, Colorado, June. Association for Computational Linguistics.

Daniel Zeman, Univerzita Karlova, and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *In IJCNLP-08 Workshop on NLP for Less Privileged Languages*, pages 35–42.

3.1.4 CoNLL 2015

Long Duong, Trevor Cohn, Steven Bird, Paul Cook. 2015. Cross-lingual Transfer for Unsupervised Dependency Parsing Without Parallel Data. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning (CoNLL 2015)*. 113–122, Beijing, China.

Research process

In both ACL 2015 (§3.1.2) and EMNLP 2015 (§3.1.3) papers, we assume a small annotated treebank in the target language. However, this treebank might not be available for many low-resource languages. That is why we want to further relax this requirement, motivating this paper.

In this paper, I design and run experiments. Other co-authors which are my supervisors contribute ideas during our weekly meeting and participate in the paper writing.

Retrospective view

It is good to see that we still improve the delexicalized parser without using any additional resource. This is thank to the syntactic word embeddings that only requires the same POS annotation between source and target language. However, for a single source language the improvement is modest. The biggest gain is from taking advantage of multiple source languages. However, the gain is not consistent across languages and usually higher for languages that share some commonalities with source languages.

Cross-lingual Transfer for Unsupervised Dependency Parsing Without Parallel Data

Long Duong,[12] **Trevor Cohn,**[1] **Steven Bird,**[1] and **Paul Cook**[3]
[1]Department of Computing and Information Systems, University of Melbourne
[2]National ICT Australia, Victoria Research Laboratory
[3]Faculty of Computer Science, University of New Brunswick
lduong@student.unimelb.edu.au {t.cohn,sbird}@unimelb.edu.au paul.cook@unb.ca

Abstract

Cross-lingual transfer has been shown to produce good results for dependency parsing of resource-poor languages. Although this avoids the need for a target language treebank, most approaches have still used large parallel corpora. However, parallel data is scarce for low-resource languages, and we report a new method that does not need parallel data. Our method learns syntactic word embeddings that generalise over the syntactic contexts of a bilingual vocabulary, and incorporates these into a neural network parser. We show empirical improvements over a baseline delexicalised parser on both the CoNLL and Universal Dependency Treebank datasets. We analyse the importance of the source languages, and show that combining multiple source-languages leads to a substantial improvement.

1 Introduction

Dependency parsing is a crucial component of many natural language processing (NLP) systems for tasks such as relation extraction (Bunescu and Mooney, 2005), statistical machine translation (Xu et al., 2009), text classification (Özgür and Güngör, 2010), and question answering (Cui et al., 2005). Supervised approaches to dependency parsing have been very successful for many resource-rich languages, where relatively large treebanks are available (McDonald et al., 2005a). However, for many languages, annotated treebanks are not available, and are very costly to create (Böhmová et al., 2001). This motivates the development of unsupervised approaches that can make use of unannotated, monolingual data. However, purely unsupervised approaches have relatively low accuracy (Klein and Manning, 2004; Gelling et al., 2012).

Most recent work on unsupervised dependency parsing for low-resource languages has used the idea of delexicalized parsing and cross-lingual transfer (Zeman et al., 2008; Søgaard, 2011; McDonald et al., 2011; Ma and Xia, 2014). In this setting, a delexicalized parser is trained on a resource-rich *source* language, and is then applied directly to a resource-poor *target* language. The only requirement here is that the source and target languages are POS tagged must use the same tagset. This assumption is pertinent for resource-poor languages since it is relatively quick to manually POS tag the data. Moreover, there are many reports of high accuracy POS tagging for resource-poor languages (Duong et al., 2014; Garrette et al., 2013; Duong et al., 2013b). The cross-lingual delexicalized approach has been shown to significantly outperform unsupervised approaches (McDonald et al., 2011; Ma and Xia, 2014).

Parallel data can be used to boost the performance of a cross-lingual parser (McDonald et al., 2011; Ma and Xia, 2014). However, parallel data may be hard to acquire for truly resource-poor languages.[1] Accordingly, we propose a method to improve the performance of a cross-lingual delexicalized parser using only monolingual data.

Our approach is based on augmenting the delexicalized parser using syntactic word embeddings. Words from both source and target language are mapped to a shared low-dimensional space based on their syntactic context, without recourse to parallel data. While prior work has struggled to efficiently incorporate word embedding information into the parsing model (Bansal et al., 2014; Andreas and Klein, 2014; Chen et al., 2014), we present a method for doing so using a neural net-

[1]Note that most research in this area (as do we) evaluates on simulated low-resource languages, through selective use of data in high-resource languages. Consequently parallel data is plentiful, however this is often not the case in the real setting, e.g., for Tagalog, where only scant parallel data exists (e.g., dictionaries, Wikipedia and the Bible).

work parser. We train our parser using a two stage process: first learning cross-lingual syntactic word embeddings, then learning the other parameters of the parsing model using a source language treebank. When applied to the target language, we show consistent gains across all studied languages.

This work is a stepping stone towards the more ambitious goal of a universal parser that can efficiently parse many languages with little modification. This aspiration is supported by the recent release of the Universal Dependency Treebank (Nivre et al., 2015) which has consensus dependency relation types and POS annotation for many languages.

When multiple source languages are available, we can attempt to boost performance by choosing the best source language, or combining information from several source languages. To the best of our knowledge, no prior work has proposed a means for selecting the best source language given a target language. To address this, we introduce two metrics which outperform the baseline of always picking English as the source language. We also propose a method for combining all available source languages which leads to substantial improvement.

The rest of this paper is organized as follows: Section 2 reviews prior work on unsupervised cross-lingual dependency parsing. Section 3 presents the methods for improving the delexicalized parser using syntactic word embeddings. Section 4 describes experiments on the CoNLL dataset and Universal Dependency Treebank. Section 5 presents methods for selecting the best source language given a target language.

2 Unsupervised Cross-lingual Dependency Parsing

There are two main approaches for building dependency parsers for resource-poor languages without using target-language treebanks: delexicalized parsing and projection (Hwa et al., 2005; Ma and Xia, 2014; Täckström et al., 2013; McDonald et al., 2011).

The delexicalized approach was proposed by Zeman et al. (2008). They built a delexicalized parser from a treebank in a resource-rich source language. This parser can be trained using any standard supervised approach, but without including any lexical features, then applied directly to parse sentences from the resource-poor language. Delexicalized parsing relies on the fact that parts-of-speech are highly informative of dependency relations. For example, an English lexicalized discriminative arc-factored dependency parser achieved 84.1% accuracy, whereas a delexicalized version achieved 78.9% (McDonald et al., 2005b; Täckström et al., 2013). Zeman et al. (2008) build a parser for Swedish using Danish, two closely-related languages. Søgaard (2011) adapt this method for less similar languages by choosing sentences from the source language that are similar to the target language. Täckström et al. (2012) additionally use cross-lingual word clustering as a feature for their delexicalized parser. Also related is the work by Naseem et al. (2012) and Täckström et al. (2013) who incorporated linguistic features from the World Atlas of Language Structures (WALS; Dryer and Haspelmath (2013)) for joint modelling of multi-lingual syntax.

In contrast, projection approaches use parallel data to project source language dependency relations to the target language (Hwa et al., 2005). Given a source-language parse tree along with word alignments, they generate the target-language parse tree by projection. However, their approach relies on many heuristics which would be difficult to adapt to other languages. McDonald et al. (2011) exploit both delexicalized parsing and parallel data, using an English delexicalized parser as the seed parser for the target languages, and updating it according to word alignments. The model encourages the target-language parse tree to look similar to the source-language parse tree with respect to the head-modifier relation. Ma and Xia (2014) use parallel data to transfer source language parser constraints to the target side via word alignments. For the null alignment, they used a delexicalized parser instead of the source language lexicalized parser.

In summary, existing work generally starts with a delexicalized parser, and uses parallel data typological information to improve it. In contrast, we want to improve the delexicalized parser, but without using parallel data or any explicit linguistic resources.

3 Improving Delexicalized Parsing

We propose a novel method to improve the performance of a delexicalized cross-lingual parser without recourse to parallel data. Our method uses no additional resources and is designed to com-

plement other methods. The approach is based on syntactic word embeddings where a word is represented as a low-dimensional vector in syntactic space. The idea is simple: we want to relexicalize the delexicalized parser using word embeddings, where source and target language lexical items are represented in the same space.

Word embeddings typically capture both syntactic and semantic information. However, we hypothesize (and later show empirically) that for dependency parsing, word embeddings need to better reflect syntax. In the next subsection, we review some cross-lingual word embedding methods and propose our syntactic word embeddings. Section 4 empirically compares these word embeddings when incorporated into a dependency parser.

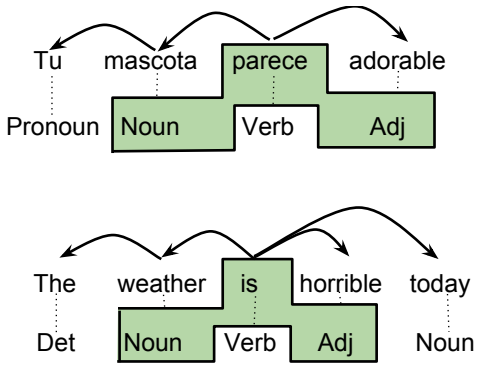


Figure 1: Examples of the syntactic word embeddings for Spanish and English. In each case, the highlighted tags are predicted by the highlighted word. The Spanish sentence means "*your pet looks lovely*".

3.1 Cross-lingual word embeddings

We review methods that can represent words in both source and target languages in a low-dimensional space. There are many benefits of using a low-dimensional space. Instead of the traditional "one-hot" representation with the number of dimensions equal to vocabulary size, words are represented using much fewer dimensions. This confers the benefit of generalising over the vocabulary to alleviate issues of data sparsity, through learning representations encoding lexical relations such as synonymy.

Several approaches have sought to learn cross-lingual word embeddings from parallel data (Hermann and Blunsom, 2014a; Hermann and Blunsom, 2014b; Xiao and Guo, 2014; Zou et al., 2013; Täckström et al., 2012). Hermann and Blunsom (2014a) induced a cross-lingual word representation based on the idea that representations for parallel sentences should be close together. They constructed a sentence level representation as a bag-of-words summing over word-level representations, and then optimized a hinge loss function to match a latent representation of both sides of a parallel sentence pair. While this might seem well suited to our needs as a word representation in cross-lingual parsing, it may lead to overly semantic embeddings, which are important for translation, but less useful for parsing. For example, "*economic*" and "*economical*" will have a similar representation despite having different syntactic features.

Also related is (Täckström et al., 2012) who build cross-lingual word representations using a variant of the Brown clusterer (Brown et al., 1992) applied to parallel data. Bansal et al. (2014) and Turian et al. (2010) showed that for monolingual dependency parsing, the simple Brown clustering based algorithm outperformed many word embedding techniques. In this paper we compare our approach to forming cross-lingual word embeddings with those of both Hermann and Blunsom (2014a) and Täckström et al. (2012).

3.2 Syntactic Word Embedding

We now propose a novel approach for learning cross-lingual word embeddings that is more heavily skewed towards syntax. Word embedding methods typically exploit word co-occurrences, building on traditional techniques for distributional similarity, e.g., the co-occurrences of words in a context window about a central word. Bansal et al. (2014) suggested that for dependency parsing, word embeddings be trained over dependency relations, instead of adjacent tokens, such that embeddings capture head and modifier relations. They showed that this strategy performed much better than surface embeddings for monolingual dependency parsing. However, their method is not applicable to our low resource setting, as it requires a parse tree for training. Instead we consider a simpler representation, namely part-of-speech contexts. This requires only POS tagging, rather than full parsing, while providing syntactic information linking words to their POS context, which we expect to be informative for characterising dependency relations.

Algorithm 1 Syntactic word embedding

1: Match the source and target tagsets to the Universal Tagset.
2: Extract word n-gram sequences for both the source and target language.
3: For each n-gram, keep the middle word, and replace the other words by their POS.
4: Train a skip-gram word embedding model on the resulting list of word and POS sequences from both the source and target language

We assume the same POS tagset is used for both the source and target language,[2] and learn word embeddings for each word type in both languages into the same syntactic space of nearby POS contexts. In particular, we develop a predictive model of the tags to the left and right of a word, as illustrated in Figure 1 and outlined in Algorithm 1. Figure 1 illustrates two training contexts extracted from our English source and Spanish target language, where the highlighted fragments reflect the tags being predicted around each focus word. Note that for this example, the POS contexts for the English and Spanish verbs are identical, and therefore the model would learn similar word embeddings for these terms, and bias the parser to generate similar dependency structures for both terms.

There are several motivations for our approach: (1) POS tags are too coarse-grained for accurate parsing, but with access to local context they can be made more informative; (2) leaving out the middle tag avoids duplication because this is already known to the parser; (3) dependency edges are often local, as shown in Figure 1, i.e., there are dependency relations between most words and their immediate neighbours. Consequently, training our embeddings to predict adjacent tags is likely to learn similar information to training over dependency edges.[3] Bansal et al. (2014) studied the effect of word embeddings on dependency parsing, and found that larger embedding windows captured more semantic information, while smaller windows better reflected syntax. Therefore we choose a small ± 1 word window in our experiments. We also experimented with bigger windows ($\pm 2, \pm 3$) but observed performance degradation in these cases, supporting the argument above.

[2]Later we consider multiple source languages, but for now assume a single source language.

[3]For the 16 languages in the CoNLL-X and CoNLL-07 datasets we observed that approx. 50% of dependency relations span a distance of one word and 20% span two words. Thus our POS context of a ± 1 word window captures the majority of dependency relations.

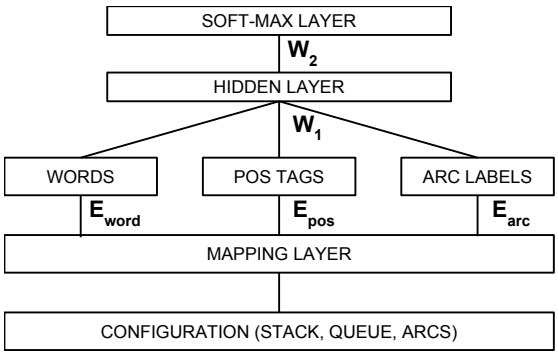


Figure 2: Neural Network Parser Architecture from Chen and Manning (2014)

Step 4 of Algorithm 1 finds the word embeddings as a side-effect of training a neural language model. We use the skip-gram model (Mikolov et al., 2013), trained to predict context tags for each word. The model is formulated as a simple bilinear logistic classifier

$$P(t_c|w) = \frac{\exp(\mathbf{u}_{t_c}^\top \mathbf{v}_w)}{\sum_{z=1}^{T} \exp(\mathbf{u}_z^\top \mathbf{v}_w)} \qquad (1)$$

where t_c is the context tag around the current word w, $\mathbf{U} \in \mathbb{R}^{T \times D}$ is the tag embedding matrix, $\mathbf{V} \in \mathbb{R}^{V \times D}$ is the word embedding matrix, with T the number of tags, V is the total number of word types over both languages and D the capacity of the embeddings. Given a training set of word and POS contexts, $(t_i^L, w_i, t_i^R)_{i=1}^N$,[4] we maximize the log-likelihood $\sum_{i=1}^{N} \log P(t_i^L|w_i) + \log P(t_i^R|w_i)$ with respect to $\mathbf{U}$ and $\mathbf{V}$ using stochastic gradient descent. The learned $\mathbf{V}$ matrix of word embeddings is later used in parser training (the source word embeddings) and inference (the target word embeddings).

3.3 Parsing Algorithm

In this Section, we show how to incorporate the syntactic word embeddings into a parsing model. Our parsing model is built based on the work of Chen and Manning (2014). They built a transition-based dependency parser using a neural-network. The neural network classifier will decide which transition is applied for each configuration.

[4]Note that w here can be a word type in either the source or target language, such that both embeddings will be learned for all word types in both languages.

The architecture of the parser is illustrated in Figure 2, where each layer is fully connected to the layer above.

For each configuration, the selected list of words, POS tags and labels from the Stack, Queue and Arcs are extracted. Each word, POS or label is mapped to a low-dimension vector representation (embedding) through the Mapping Layer. This layer simply concatenates the embeddings which are then fed into a two-layer neural network classifier to predict the next parsing action. The set of parameters for the neural network classifier is $E_{word}, E_{pos}, E_{labels}$ for the mapping layer, W_1 for the hidden layer and W_2 for the soft-max output layer. We incorporate the syntactic word embeddings into the neural network model by setting E_{word} to the syntactic word embeddings, which remain fixed during training so as to retain the cross-lingual mapping.[5]

3.4 Model Summary

To apply the parser to a resource-poor target language, we start by building syntactic word embeddings between source and target languages as shown in algorithm 1. Next we incorporate syntactic word embeddings using the algorithm proposed in Section 3.3. The third step is to substitute source- with target-language syntactic word embeddings. Finally, we parse the target language using this substituted model. In this way, the model will recognize lexical items for the target language.

4 Experiments

We test our method of incorporating syntactic word embeddings into a neural network parser, for both the existing CoNLL dataset (Buchholz and Marsi, 2006; Nivre et al., 2007) and the newly-released Universal Dependency Treebank (Nivre et al., 2015). We employed the Unlabeled Attachment Score (UAS) without punctuation for comparison with prior work on the CoNLL dataset. Where possible we also report Labeled Attachment Score (LAS) without punctuation. We use English as the source language for this experiment.

[5]This is a consequence of only training the parser on the source language. If we were to update embeddings during parser training this would mean they no longer align with the target language embeddings.

4.1 Experiments on CoNLL Data

In this section we report experiments involving the CoNLL-X and CoNLL-07 datasets. Running on this dataset makes our model comparable with prior work. For languages included in both datasets, we use the newer one only. Crucially, for the delexicalized parser we map language-specific tags to the universal tagset (Petrov et al., 2012). The syntactic word embeddings are trained using POS information from the CoNLL data.

There are two baselines for our experiment. The first one is the unsupervised dependency parser of Klein and Manning (2004), the second one is the delexicalized parser of Täckström et al. (2012). We also compare our syntactic word embedding with the cross-lingual word embeddings of Hermann and Blunsom (2014a). These word embeddings are induced by running each language pair using Europarl (Koehn, 2005). We incorporated Hermann and Blunsom (2014a)'s cross-lingual word embeddings into the parsing model in the same way as for the syntactic word embeddings. Table 1 shows the UAS for 8 languages for several models. The first observation is that the direct transfer delexicalized parser outperformed the unsupervised approach. This is consistent with many prior studies. Our implementation of the direct transfer model performed on par with Täckström et al. (2012) on average. Table 1 also shows that using HB embeddings improve the performance over the Direct Transfer model. Our model using syntactic word embedding consistently out-performed the Direct Transfer model and HB embedding across all 8 languages. On average, it is 1.5% and 1.3% better. [6] The improvement varies across languages compared with HB embedding, and falls in the range of 0.3 to 2.6%. This confirms our initial hypothesis that we need word embeddings that capture syntactic instead of semantic information.

It is not strictly fair to compare our method with prior approaches to unsupervised dependency parsing, since they have different resource requirement, i.e. parallel data or typological resources. Compared with the baseline of the direct transfer model, our approach delivered a 1.5% mean performance gain, whereas Täckström et al. (2012) and McDonald et al. (2011) report approximately 3% gain, Ma and Xia (2014) and Naseem et al. (2012) report an approximately 6% gain. As we

[6]All performance comparisons in this paper are absolute.

	da	de	el	es	it	nl	pt	sv	Avg
Unsupervised	33.4	18.0	39.9	28.5	43.1	38.5	20.1	44.0	33.2
Täckström et al. (2012) DT	36.7	48.9	59.5	60.2	64.4	52.8	66.8	55.4	55.6
Our Direct Transfer	44.1	44.9	63.3	52.2	57.7	59.7	67.5	55.4	55.6
Our Model + HB embedding	45.0	44.5	63.8	52.2	56.7	59.8	68.7	55.6	55.8
Our Model + Syntactic embedding	45.9	45.9	64.1	52.9	59.1	61.1	69.5	58.1	57.1

Table 1: Comparative results on the CoNLL corpora showing UAS for several parsers: unsupervised induction Klein and Manning (2004), Direct Transfer (DT) delexicalized parser of Täckström et al. (2012), our implementation of Direct Transfer and our neural network parsing model using cross-lingual embeddings Hermann and Blunsom (2014a) (HB) and our proposed syntactic embeddings.

	cs	de	en	es	fi	fr	ga	hu	it	sv
Train	1173.3	269.6	204.6	382.4	162.7	354.7	16.7	20.8	194.1	66.6
Dev	159.3	12.4	25.1	41.7	9.2	38.9	3.2	3.0	10.5	9.8
Test	173.9	16.6	25.1	8.5	9.1	7.1	3.8	2.7	10.2	20.4
Total	1506.5	298.6	254.8	432.6	181	400.7	23.7	26.5	214.8	96.8

Table 2: Number of tokens ($\times$ 1000) for each language in the Universal Dependency Treebank.

have stated above, our approach is complementary to the approaches used in these other systems. For example, we could incorporate the cross-lingual word clustering feature (Täckström et al., 2012) or WALS features (Naseem et al., 2012) into our model, or use our improved delexicalized parser as the reference model for Ma and Xia (2014), which we expect would lead to better results yet.

4.2 Experiments with Universal Dependency Treebank

We also experimented with the Universal Dependency Treebank V1.0, which has many desirable properties for our system, e.g. dependency types and coarse POS are the same across languages. This removes the need for mapping the source and target language tagsets to a common tagset, as was done for the CoNLL data. Secondly, instead of only reporting UAS we can report LAS, which is impossible on CoNLL dataset where the dependency edge labels differed among languages.

Table 2 shows the size in thousands of tokens for each language in the treebank. The first thing to observe is that some languages have abundant amount of data such as Czech (cs), French (fr) and Spanish (es). However, there are languages with modest size i.e. Hungarian (hu) and Irish (ga).

We ran our model with and without syntactic word embeddings for all languages with English as the source language. The results are shown in Table 3. The first observation is that our model using syntactic word embeddings out-performed direct transfer for all the languages on both UAS and LAS. We observed an average improvement of 3.6% (UAS) and 3.1% (LAS). This consistent improvement shows the robustness of our method of incorporating syntactic word embedding to the model. The second observation is that the gap between UAS and LAS is as big as 13% on average for both models. This reflects the increase difficulty of labelling the edges, with unlabelled edge prediction involving only a 3-way classification[7] while labelled edge prediction involves an 81-way classification.[8] Narrowing the gap between UAS and LAS for resource-poor languages is an important research area for future work.

5 Different Source Languages

In the previous sections, we used English as the source language. However, English might not be the best choice. For the delexicalized parser, it is crucial that the source and target languages have similar syntactic structures. Therefore a different choice of source language might substantially change the performance, as observed in prior studies (Täckström et al., 2013; Duong et al., 2013a; McDonald et al., 2011).

[7] Since there are only 3 transitions: SHIFT, LEFT-ARC, RIGHT-ARC.

[8] Since the Universal Dependency Treebank has 40 universal relations, each relation is attached to LEFT-ARC or RIGHT-ARC. The number 81 comes from 1 (SHIFT) + 40 (LEFT-ARC) + 40 (RIGHT-ARC).

	cs	de	es	fi	fr	ga	hu	it	sv	UAS	LAS
Direct Transfer	47.2	57.9	64.7	44.9	64.8	49.1	47.8	64.9	55.5	55.2	42.7
Our Model + Syntactic embedding	50.2	60.9	67.9	51.4	66.0	51.6	52.3	69.2	59.6	58.8	45.8

Table 3: Results comparing a direct transfer parser and our model with syntactic word embeddings. Evaluating UAS over the Universal Dependency Treebank. (We observed a similar pattern for LAS.) The rightmost UAS and LAS columns shows the average scores for the respective metric across 9 languages.

		TARGET LANGUAGE											
		cs	de	en	es	fi	fr	ga	hu	it	sv	UAS	LAS
SOURCE LANGUAGE	cs	76.8	**65.9**	60.8	70.0	**53.7**	66.8	**59.0**	55.2	70.7	56.8	62.1	38.7
	de	60.0	78.2	61.7	63.1	52.4	60.6	49.8	**56.7**	64.0	59.5	58.6	45.5
	en	50.2	60.9	81.0	67.9	51.4	66.0	51.6	52.3	69.2	**59.6**	58.8	45.8
	es	**60.5**	58.5	60.4	80.9	45.7	**73.3**	53.8	46.9	**77.4**	55.3	59.1	46.2
	fi	49.0	41.8	44.5	33.6	71.5	35.2	24.4	44.6	31.7	43.1	38.7	25.5
	fr	54.2	55.7	**63.2**	**74.8**	43.6	79.2	54.7	44.3	76.2	54.8	57.9	46.3
	ga	32.8	35.3	39.8	56.3	23.5	52.6	72.3	26.0	58.3	32.6	39.7	26.7
	hu	42.3	53.4	45.4	43.8	53.3	42.1	29.2	72.1	41.2	42.5	43.7	22.7
	it	57.6	53.4	53.2	72.1	42.7	71.4	54.7	42.2	85.9	54.2	55.7	45.0
	sv	49.1	59.2	54.9	59.8	47.9	55.7	48.5	52.7	62.2	78.4	54.4	41.2

Table 4: UAS for each language pair in the Universal Dependency Treebank using our best model. The UAS/LAS column show the average UAS/LAS for all target languages, excluding the source language. The best UAS for each target language is shown in bold.

In this section we assume that we have multiple source languages. To see how the performance changes when using a different source language, we run our best model (i.e., using syntactic embeddings) for each language pair in the Universal Dependency Treebank. Table 4 shows the UAS for each language pair, and the average across all target languages for each source language. We also considered LAS, but observed similar trends, and therefore only report the average LAS for each source language. Observe that English is rarely the best source language; Czech and French give a higher average UAS and LAS, respectively. Interestingly, while Czech gives high UAS on average, it performs relatively poorly in terms of LAS.

One might expect that the relative performance from using different source languages is affected by the source corpus size, which varies greatly. We tested this question by limiting the source corpora 66K sentences (and excluded the very small *ga* and *hu* datasets), which resulted in a slight reduction in scores but overall a near identical pattern of results to the use of the full sized source corpora reported in Table 4. Only in one instance did the best source language change (for target *fi* with source *de* not *cs*), and the average rankings by UAS and LAS remained unchanged.

The ten languages considered belong to five families: *Romance* (French, Spanish, Italian), *Germanic* (German, English, Swedish), *Slavic* (Czech), *Uralic* (Hungarian, Finnish), and *Celtic* (Irish). At first glance it seems that language pairs in the same family tend to perform well. For example, the best source language for both French and Italian is Spanish, while the best source language for Spanish is French. However, this doesn't hold true for many target languages. For example, the best source language for both Finnish and German is Czech. It appears that the best choice of an appropriate source language is not predictable from language family information.

We therefore propose two methods to predict the best source language for a given target language. In devising these methods we assume that for a given resource-poor target language we do not have access to any parsed data, as this is expensive to construct. The first method is based on the Jensen-Shannon divergence between the distributions of POS n-grams ($1 < n < 6$) in a pair of languages. The second method converts each language into a vector of binary features based on word-order information from WALS, the World

	cs	de	en	es	fi	fr	ga	hu	it	sv	UAS	LAS
English	50.2	60.9	—	67.9	51.4	66.0	51.6	52.3	69.2	59.6	58.8	45.8
WALS	50.2	59.2	44.5	72.1	51.4	73.3	53.8	44.6	77.4	59.6	60.2	47.1
POS	49.1	58.5	53.2	74.8	53.7	73.3	53.8	56.7	76.2	56.8	61.4	47.7
Oracle	60.5	65.9	63.2	74.8	53.7	73.3	59.0	56.7	77.4	59.6	64.5	50.8
Combined	61.1	67.5	64.4	75.1	54.2	72.8	58.7	57.9	76.7	60.5	64.9	52.0

Table 5: UAS for target languages where the source language is selected in different ways. English uses English as the source language. WALS and POS choose the best source language using the WALS or POS ngrams based methods, respectively. Oracle always uses the best source language. Combined is the model that combines information from all available sources language. The UAS/LAS columns show the UAS/LAS average performance across 9 languages (English is excluded).

Atlas of Language Structures (Dryer and Haspelmath, 2013). These features include the relative order of adjective and noun, etc, and we compute the cosine similarity between the vectors for a pair of languages.

As an alternative to selecting a single source language, we further propose a method to combine information from all available source languages to build a parser for a target language. To do so we first train the syntactic word embeddings on all the languages. After this step, lexical items from all source languages and the target language will be in the same space. We train our parser with syntactic word embeddings on the combined corpus of all source languages. This parser is then applied to the target language directly. The intuition here is that training on multiple source languages limits over-fitting to the source language, and learns the "universal" structure of languages.

Table 5 shows the performance of each target language with the source language given by the model (in the case of models that select a single source language). Always choosing English as the source language performs worst. Using WALS features out-performs English on 7 out of 9 languages. Using POS ngrams out-performs the WALS feature model on average for both UAS and LAS, although the improvement is small. The combined model, which combines information from all available source languages, out-performs choosing a single source language. Moreover, this model performs even better than the oracle model, which always chooses the single best source language, especially for LAS. Compared with the baseline of always choosing English, our combined model gives an improvement about 6% for both UAS and LAS.

6 Conclusions

Most prior work on cross-lingual transfer dependency parsing has relied on large parallel corpora. However, parallel data is scarce for resource-poor languages. In the first part of this paper we investigated building a dependency parser for a resource-poor language without parallel data. We improved the performance of a delexicalized parser using syntactic word embeddings using a neural network parser. We showed that syntactic word embeddings are better at capturing syntactic information, and particularly suitable for dependency parsing. In contrast to the state-of-the-art for unsupervised cross-lingual dependency parsing, our method does not rely on parallel data. Although the state-of-the-art achieves bigger gains over the baseline than our method, our approach could be more-widely applied to resource-poor languages because of its lower resource requirements. Moreover, we have described how our method could be used to complement previous approaches.

The second part of this paper studied ways of improving performance when multiple source languages are available. We proposed two methods to select a single source language that both lead to improvements over always choosing English as the source language. We then showed that we can further improve performance by combining information from all the source languages. In summary, without any parallel data, we managed to improve the direct transfer delexicalized parser by about 10% for both UAS and LAS on average, for 9 languages in the Universal Dependency Treebank.

In this paper we focused only on word embeddings, however, in future work we could also build the POS embeddings and the arc-label embeddings across languages. This could help our

system to move more freely across languages, facilitating not only the development of NLP for resource-poor languages, but also cross-language comparisons.

Acknowledgments

This work was supported by the University of Melbourne and National ICT Australia (NICTA). Dr Cohn is the recipient of an Australian Research Council Future Fellowship (project number FT130101105).

References

Jacob Andreas and Dan Klein. 2014. How much do word embeddings encode about syntax? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 822–827, Baltimore, Maryland.

Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2014. Tailoring continuous word representations for dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 809–815.

Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. 2001. The Prague Dependency Treebank: A Three-Level Annotation Scenario. In Anne Abeillé, editor, *Treebanks: Building and Using Syntactically Annotated Corpora*, pages 103–127.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*.

Razvan C. Bunescu and Raymond J. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 724–731.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar.

Wenliang Chen, Yue Zhang, and Min Zhang. 2014. Feature embedding for dependency parsing. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 816–826, Dublin, Ireland.

Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. 2005. Question answering passage retrieval using dependency relations. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 400–407, New York, NY, USA.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Leipzig.

Long Duong, Paul Cook, Steven Bird, and Pavel Pecina. 2013a. Increasing the quality and quantity of source language data for Unsupervised Cross-Lingual POS tagging. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1243–1249, Nagoya, Japan.

Long Duong, Paul Cook, Steven Bird, and Pavel Pecina. 2013b. Simpler unsupervised POS tagging with bilingual projections. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 634–639, Sofia, Bulgaria.

Long Duong, Trevor Cohn, Karin Verspoor, Steven Bird, and Paul Cook. 2014. What can we get from 1000 tokens? a case study of multilingual pos tagging for resource-poor languages. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 886–897, Doha, Qatar.

Dan Garrette, Jason Mielens, and Jason Baldridge. 2013. Real-world semi-supervised learning of pos-taggers for low-resource languages. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL-2013)*, pages 583–592, Sofia, Bulgaria.

Douwe Gelling, Trevor Cohn, Phil Blunsom, and Joo Graa. 2012. The pascal challenge on grammar induction.

Karl Moritz Hermann and Phil Blunsom. 2014a. Multilingual Distributed Representations without Word Alignment. In *Proceedings of ICLR*.

Karl Moritz Hermann and Phil Blunsom. 2014b. Multilingual models for compositional distributed semantics. *CoRR*, abs/1404.4641.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11:311–325.

Dan Klein and Christopher Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*, pages 79–86, Phuket, Thailand.

Xuezhe Ma and Fei Xia. 2014. Unsupervised dependency parsing with transferring distribution via parallel guidance and entropy regularization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1337–1348.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005a. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 91–98.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005b. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 523–530.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 62–72.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C.j.c. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Tahira Naseem, Regina Barzilay, and Amir Globerson. 2012. Selective sharing for multilingual dependency parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 629–637.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932, Prague, Czech Republic.

Joakim Nivre, Cristina Bosco, Jinho Choi, Marie-Catherine de Marneffe, Timothy Dozat, Richárd Farkas, Jennifer Foster, Filip Ginter, Yoav Goldberg, Jan Hajič, Jenna Kanerva, Veronika Laippala, Alessandro Lenci, Teresa Lynn, Christopher Manning, Ryan McDonald, Anna Missilä, Simonetta Montemagni, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Maria Simi, Aaron Smith, Reut Tsarfaty, Veronika Vincze, and Daniel Zeman. 2015. Universal dependencies 1.0.

Levent Özgür and Tunga Güngör. 2010. Text classification with the support of pruned dependency patterns. *Pattern Recognition Letter*, 31:1598–1607.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey.

Anders Søgaard. 2011. Data point selection for cross-language adaptation of dependency parsers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 682–686.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 477–487.

Oscar Täckström, Ryan McDonald, and Joakim Nivre. 2013. Target language adaptation of discriminative transfer parsers. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1061–1071, Atlanta, Georgia.

Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, Uppsala, Sweden.

Min Xiao and Yuhong Guo, 2014. *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, chapter Distributed Word Representation Learning for Cross-Lingual Dependency Parsing, pages 119–129.

Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. 2009. Using a dependency parser to improve smt for subject-object-verb languages. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 245–253, Boulder, Colorado.

Daniel Zeman, Univerzita Karlova, and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *In IJCNLP-08 Workshop on NLP for Less Privileged Languages*, pages 35–42.

Will Y. Zou, Richard Socher, Daniel Cer, and Christopher D. Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1398, Seattle, Washington, USA.

3.1.5 EMNLP 2016

Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, Trevor Cohn. Learning Crosslingual Word Embeddings without Bilingual Corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*. 1285–1295, Austin, Texas, USA

Research process

Transfer learning is the core idea of all our previous paper. Lexical transfer is one of the most important part. Investigating a better way to do lexical transfer would benefit not only POS tagging or dependency parsing but also many other transfer learning task, motivating this paper. This paper is done during my internship at IBM research Tokyo where they interested in transfer learning for their data mining system.

Aside from Steven and Trevor from Melbourne university, I also have Hiroshi and Tengfei as my IBM side supervisor. In this paper, I implement the algorithm and run all experiments. Other co-authors contribute ideas during our weekly meeting and participate in the paper writing.

Retrospective view

This paper uses dictionary for building crosslingual word embeddings applied successfully for both intrinsic tasks (monolingual similarity and bilingual lexicon induction task) and extrinsic task (crosslingual word embeddings). However, because of space constrain, we haven't evaluated on syntactic extrinsic task such as crosslingual dependency parsing. Moreover, crosslingual word embeddings currently only work for a pair of language, it is shown in ConLL 2015 (§3.1.4) that crosslingual word embeddings for multiple languages are more beneficial in transfer learning.

Learning Crosslingual Word Embeddings without Bilingual Corpora

Long Duong,[12] **Hiroshi Kanayama,**[3] **Tengfei Ma,**[3] **Steven Bird**[14] and **Trevor Cohn**[1]
[1]Department of Computing and Information Systems, University of Melbourne
[2]National ICT Australia, Victoria Research Laboratory
[3]IBM Research – Tokyo
[4]International Computer Science Institute, University of California Berkeley

Abstract

Crosslingual word embeddings represent lexical items from different languages in the same vector space, enabling transfer of NLP tools. However, previous attempts had expensive resource requirements, difficulty incorporating monolingual data or were unable to handle polysemy. We address these drawbacks in our method which takes advantage of a high coverage dictionary in an EM style training algorithm over monolingual corpora in two languages. Our model achieves state-of-the-art performance on bilingual lexicon induction task exceeding models using large bilingual corpora, and competitive results on the monolingual word similarity and cross-lingual document classification task.

1 Introduction

Monolingual word embeddings have had widespread success in many NLP tasks including sentiment analysis (Socher et al., 2013), dependency parsing (Dyer et al., 2015), machine translation (Bahdanau et al., 2014). Crosslingual word embeddings are a natural extension facilitating various crosslingual tasks, e.g. through transfer learning. A model built in a source resource-rich language can then applied to the target resource poor languages (Yarowsky and Ngai, 2001; Das and Petrov, 2011; Täckström et al., 2012; Duong et al., 2015). A key barrier for crosslingual transfer is lexical matching between the source and the target language. Crosslingual word embeddings are a natural remedy where both source and target language lexicon are presented as dense vectors in the same vector space (Klementiev et al., 2012).

Most previous work has focused on down-stream crosslingual applications such as document classification and dependency parsing. We argue that good crosslingual embeddings should preserve both monolingual and crosslingual quality which we will use as the main evaluation criterion through monolingual word similarity and bilingual lexicon induction tasks. Moreover, many prior work (Chandar A P et al., 2014; Kočiský et al., 2014) used bilingual or comparable corpus which is also expensive for many low-resource languages. Søgaard et al. (2015) impose a less onerous data condition in the form of linked Wikipedia entries across several languages, however this approach tends to underperform other methods. To capture the monolingual distributional properties of words it is crucial to train on large monolingual corpora (Luong et al., 2015). However, many previous approaches are not capable of scaling up either because of the complicated objective functions or the nature of the algorithm. Other methods use a dictionary as the bridge between languages (Mikolov et al., 2013a; Xiao and Guo, 2014), however they do not adequately handle translation ambiguity.

Our model uses a bilingual dictionary from Panlex (Kamholz et al., 2014) as the source of bilingual signal. Panlex covers more than a thousand languages and therefore our approach applies to many languages, including low-resource languages. Our method selects the translation based on the context in an Expectation-Maximization style training algorithm which explicitly handles polysemy through incorporating multiple dictionary translations (word sense and translation are closely linked (Resnik and Yarowsky, 1999)). In addition to the dictionary,

our method only requires monolingual data. Our approach is an extension of the continuous bag-of-words (CBOW) model (Mikolov et al., 2013b) to inject multilingual training signal based on dictionary translations. We experiment with several variations of our model, whereby we predict only the translation or both word and its translation and consider different ways of using the different learned center-word versus context embeddings in application tasks. We also propose a regularisation method to combine the two embedding matrices during training. Together, these modifications substantially improve the performance across several tasks. Our final model achieves state-of-the-art performance on bilingual lexicon induction task, large improvement over word similarity task compared with previous published crosslingual word embeddings, and competitive result on cross-lingual document classification task. Notably, our embedding combining techniques are general, yielding improvements also for monolingual word embedding.

This paper makes the following contributions:

- Proposing a new crosslingual training method for learning vector embeddings, based only on monolingual corpora and a bilingual dictionary;
- Evaluating several methods for combining embeddings, which are shown to help in both crosslingual and monolingual evaluations; and
- Achieving consistent results which are competitive in monolingual, bilingual and crosslingual transfer settings.

2 Related work

There is a wealth of prior work on crosslingual word embeddings, which all exploit some kind of bilingual resource. This is often in the form of a parallel bilingual text, using word alignments as a bridge between tokens in the source and target languages, such that translations are assigned similar embedding vectors (Luong et al., 2015; Klementiev et al., 2012). These approaches are affected by errors from automatic word alignments, motivating other approaches which operate at the sentence level (Chandar A P et al., 2014; Hermann and Blunsom, 2014; Gouws et al., 2015) through learning compositional vector representations of sentences, in order that sentences and their translations representations closely match. The word embeddings learned this way capture translational equivalence, despite not using explicit word alignments. Nevertheless, these approaches demand large parallel corpora, which are not available for many language pairs.

Vulić and Moens (2015) use bilingual comparable text, sourced from Wikipedia. Their approach creates a psuedo-document by forming a bag-of-words from the lemmatized nouns in each comparable document concatenated over both languages. These pseudo-documents are then used for learning vector representations using `Word2Vec`. Their system, despite its simplicity, performed surprisingly well on a bilingual lexicon induction task (we compare our method with theirs on this task.) Their approach is compelling due to its lesser resource requirements, although comparable bilingual data is scarce for many languages. Related, Søgaard et al. (2015) exploit the comparable part of Wikipedia. They represent word using Wikipedia entries which are shared for many languages.

A bilingual dictionary is an alternative source of bilingual information. Gouws and Søgaard (2015) randomly replace the text in a monolingual corpus with a random translation, using this corpus for learning word embeddings. Their approach doesn't handle polysemy, as very few of the translations for each word will be valid in context. For this reason a high coverage or noisy dictionary with many translations might lead to poor outcomes. Mikolov et al. (2013a), Xiao and Guo (2014) and Faruqui and Dyer (2014) filter a bilingual dictionary for one-to-one translations, thus side-stepping the problem, however discarding much of the information in the dictionary. Our approach also uses a dictionary, however we use all the translations and explicitly disambiguate translations during training.

Another distinguishing feature on the above-cited research is the method for training embeddings. Mikolov et al. (2013a) and Faruqui and Dyer (2014) use a cascade style of training where the word embeddings in both source and target language are trained separately and then combined later using the dictionary. Most of the other works train multlingual models jointly, which appears to have better performance over cascade training (Gouws et al., 2015).

For this reason we also use a form of joint training in our work.

3 Word2Vec

Our model is an extension of the contextual bag of words (CBOW) model of Mikolov et al. (2013b), a method for learning vector representations of words based on their distributional contexts. Specifically, their model describes the probability of a token w_i at position i using logistic regression with a factored parameterisation,

$$p(w_i|w_{i\pm k\setminus i}) = \frac{\exp(\mathbf{u}_{w_i}^\top \mathbf{h}_i)}{\sum_{w\in W}\exp(\mathbf{u}_w^\top \mathbf{h}_i)}\,, \qquad (1)$$

where $\mathbf{h}_i = \frac{1}{2k}\sum_{j=-k;j\neq 0}^{k}\mathbf{v}_{w_{i+j}}$ is a vector encoding the context over a window of size k centred around position i, W is the vocabulary and the parameters $\mathbf{V}$ and $\mathbf{U} \in \mathbb{R}^{|W|\times d}$ are matrices referred to as the context and word embeddings. The model is trained to maximise the log-pseudo likelihood of a training corpus, however due to the high complexity of computing the denominator of equation (1), Mikolov et al. (2013b) propose negative sampling as an approximation, by instead learning to differentiate data from noise (negative examples). This gives rise to the following optimisation objective

$$\sum_{i\in D}\left(\log\sigma(\mathbf{u}_{w_i}^\top \mathbf{h}_i) + \sum_{j=1}^{p}\mathbb{E}_{w_j\sim P_n(w)}\log\sigma(-\mathbf{u}_{w_j}^\top \mathbf{h}_i)\right), \qquad (2)$$

where D is the training data and p is the number of negative examples randomly drawn from a noise distribution $P_n(w)$.

4 Our Approach

Our approach extends CBOW to model bilingual text, using two monolingual corpora and a bilingual dictionary. We believe this data condition to be less stringent than requiring parallel or comparable texts as the source of the bilingual signal. It is common for field linguists to construct a bilingual dictionary when studying a new language, as one of the first steps in the language documentation process. Translation dictionaries are a rich information source, capturing much of the lexical ambiguity in a language through translation. For example, the word *bank* in English might mean the *river bank* or *financial bank* which corresponds to two different translations *sponda* and *banca* in Italian. If we are able to learn to select good translations, then this implicitly resolves much of the semantic ambiguity in the language, and accordingly we seek to use this idea to learn better semantic vector representations of words.

Algorithm 1 EM algorithm for selecting translation during training, where $\theta = (\mathbf{U}, \mathbf{V})$ are the model parameters and η is the learning rate.

randomly initialize $\mathbf{V}$, $\mathbf{U}$
for $i <$ `Iter` **do**
  for $i \in D_e \cup D_f$ **do**
    $\mathbf{s} \leftarrow \mathbf{v}_{w_i} + \mathbf{h}_i$
    $\bar{w}_i = \text{argmax}_{w\in\text{dict}(w_i)}\cos(\mathbf{s}, \mathbf{v}_w)$
    $\theta \leftarrow \theta + \eta\frac{\partial\mathcal{O}(\bar{w}_i, w_i, \mathbf{h}_i)}{\partial\theta}$ {see (3) or (5)}
  end for
end for

4.1 Dictionary replacement

To learn bilingual relations, we use the context in one language to predict the translation of the centre word in another language. This is motivated by the fact that the context is an excellent means of disambiguating the translation for a word. Our method is closely related to Gouws and Søgaard (2015), however we only replace the middle word w_i with a translation $\bar{w}_i$ while keeping the context fixed. We replace each centre word with a translation on the fly during training, predicting instead $p(\bar{w}_i|w_{i\pm k\setminus i})$ but using the same formulation as equation (1) albeit with an augmented $\mathbf{U}$ matrix to cover word types in both languages.

The translation $\bar{w}_i$ is selected from the possible translations of w_i listed in the dictionary. The problem of selecting the correct translation from the many options is reminiscent of the problem faced in expectation maximisation (EM), in that cross-lingual word embeddings will allow for accurate translation, however to learn these embeddings we need to know the translations. We propose an EM-inspired algorithm, as shown in Algorithm 1, which operates over both monolingual corpora, D_e and D_f. The vector $\mathbf{s}$ is the semantic representation combining both the centre word, w_i, and the con-

text,[1] which is used to choose the best translation into the other language from the bilingual dictionary $dict(w_i)$.[2] After selecting the translation, we use $\bar{w}_i$ together with the context vector $\mathbf{h}$ to make a stochastic gradient update of the CBOW log-likelihood.

4.2 Joint Training

Words and their translations should appear in very similar contexts. One way to enforce this is to jointly learn to predict both the word and its translation from its monolingual context. This gives rise to the following joint objective function,

$$\mathcal{O} = \sum_{i \in D_e \cup D_f} \Big(\alpha \log \sigma(\mathbf{u}_{w_i}^\top \mathbf{h}_i) + (1-\alpha) \log \sigma(\mathbf{u}_{\bar{w}_i}^\top \mathbf{h}_i) + \sum_{j=1}^{p} \mathbb{E}_{w_j \sim P_n(w)} \log \sigma(-\mathbf{u}_{w_j}^\top \mathbf{h}_i) \Big), \quad (3)$$

where α controls the contribution of the two terms. For our experiments, we set $\alpha = 0.5$. The negative examples are drawn from combined vocabulary unigram distribution calculated from combined data $D_e \cup D_f$.

4.3 Combining Embeddings

Many vector learning methods learn two embedding spaces $\mathbf{V}$ and $\mathbf{U}$. Usually only $\mathbf{V}$ is used in application. The use of $\mathbf{U}$, on the other hand, is understudied (Levy and Goldberg, 2014) with the exception of Pennington et al. (2014) who use a linear combination $\mathbf{U} + \mathbf{V}$, with minor improvement over $\mathbf{V}$ alone.

We argue that with our model, $\mathbf{V}$ is better at capturing the monolingual regularities and $\mathbf{U}$ is better at capturing bilingual signal. The intuition for this is as follows. Assuming that we are predicting the word *finance* and its Italian translation *finanze* from the context (*money, loan, bank, debt, credit*) as shown in figure 1. In $\mathbf{V}$ only the context word representations are updated and in $\mathbf{U}$ only the representations of *finance, finanze* and negative samples such as *tree* and *dog* are updated. CBOW learns good embeddings because each time it updates the parameters, the words in the contexts are pushed closer to each other in the $\mathbf{V}$ space. Similarly, the target word w_i and the translation $\bar{w}_i$ are also pushed closer in the $\mathbf{U}$ space. This is directly related to poitwise mutual information values of each pair of word and context explained in Levy and Goldberg (2014). Thus, $\mathbf{U}$ is bound to better at bilingual lexicon induction task and $\mathbf{V}$ is better at monolingual word similarity task.

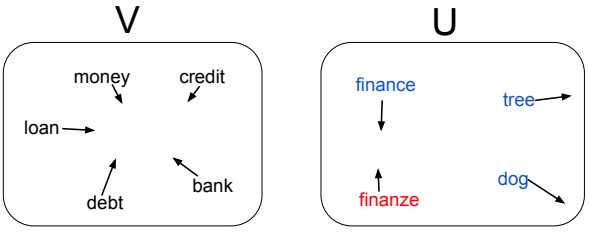


Figure 1: Example of $\mathbf{V}$ and $\mathbf{U}$ space during training.

The simple question is, how to combine both $\mathbf{V}$ and $\mathbf{U}$ to produce a better representation. We experiment with several ways to combine $\mathbf{V}$ and $\mathbf{U}$. First, we can follow Pennington et al. (2014) to *interpolate* $\mathbf{V}$ and $\mathbf{U}$ in the post-processing step. i.e.

$$\gamma \mathbf{V} + (1-\gamma)\mathbf{U} \quad (4)$$

where γ controls the contribution of each embedding space. Second, we can also *concatenate* $\mathbf{V}$ and $\mathbf{U}$ instead of interpolation such that $\mathbf{C} = [\mathbf{V} : \mathbf{U}]$ where $\mathbf{C} \in \mathbb{R}^{|W| \times 2d}$ and W is the combined vocabulary from $D_e \cup D_f$.

Moreover, we can also fuse $\mathbf{V}$ and $\mathbf{U}$ during training. For each word in the combined dictionary $V_e \cup V_f$, we encourage the model to learn similar representation in both $\mathbf{V}$ and $\mathbf{U}$ by adding a *regularization* term to the objective function in equation (3) during training.

$$\mathcal{O}' = \mathcal{O} + \delta \sum_{w \in V_e \cup V_f} \|\mathbf{u}_w - \mathbf{v}_w\|_2^2 \quad (5)$$

where δ controls to what degree we should bind two spaces together.[3]

5 Experimental Setup

Our experimental evaluation seeks to determine how well lexical distances in the learned embedding

[1] Using both embeddings gives a small improvement compared to just using context vector $\mathbf{h}$ alone.

[2] We also experimented with using expectations over translations, as per standard EM, with slight degredation in results.

[3] In the stochastic gradient update for a given word in context, we only compute the gradient of the regularisation term in (5) with respect to the words in the set of positive and negative examples.

spaces match with known lexical similarity judgements from bilingual and monolingual lexical resources. To this end, in §6 we test crosslingual distances using a bilingual lexicon induction task in which we evaluate the embeddings in terms of how well nearby pairs of words from two languages in the embedding space match with human judgements. Next, to evaluate the monolingual embeddings we evaluate word similarities in a single language against standard similarity datasets (§7). Lastly, to demonstrate the usefulness of our embeddings in a task-based setting, we evaluate on crosslingual document classification (§9).

Monolingual Data The monolingual data is taken from the pre-processed Wikipedia dump from Al-Rfou et al. (2013). The data is already cleaned and tokenized. We additionally lower-case all words. Normally monolingual word embeddings are trained on billions of words. However, obtaining that much monolingual data for a low-resource language is infeasible. Therefore, we only select the first 5 million sentences (around 100 million words) for each language.

Dictionary A bilingual dictionary is the only source of bilingual correspondence in our technique. We prefer a dictionary that covers many languages, such that our approach can be applied widely to many low-resource languages. We use Panlex, a dictionary which currently covers around 1300 language varieties with about 12 million expressions. The translations in PanLex come from various sources such as glossaries, dictionaries, automatic inference from other languages, etc. Accordingly, Panlex has high language coverage but often noisy translations.[4] Table 1 summarizes the sizes of monolingual corpora and dictionaries for each pair of language in our experiments.

[4]We also experimented with a crowd-sourced dictionary from Wiktionary. Our initial observation was that the translation quality was better but with a lower-coverage. For example, for `en-it` dictionary, Panlex and Wiktionary have a coverage of 42.1% and 16.8% respectively for the top 100k most frequent English words from Wikipedia. The average number of translations are 5.2 and 1.9 respectively. We observed similar trend using Panlex and Wiktionary dictionary in our model. However, using Panlex results in much better performance. We can run the model on the combined dictionary from both Panlex and Wiktionary but we leave it for future work.

	Source (M)	Target (M)	Dict (k)
`en-es`	120.1 (73.9%)	126.8 (74.4%)	712.0
`en-it`	120.1 (74.7%)	114.6 (67.4%)	560.1
`en-nl`	120.1 (69.1%)	80.2 (63.4%)	406.6
`en-de`	120.1 (77.8%)	90.8 (68.3%)	964.4
`en-sr`	120.1 (28.0%)	7.5 (17.5%)	35.1

Table 1: Number of tokens in millions for the source and target languages in each language pair. Also shown is the number of entries in the bilingual dictionary in thousands. The number in the parenthesis shows the token coverage in the dictionary on each monolingual corpus.

6 Bilingual Lexicon Induction

Given a word in a source language, the bilingual lexicon induction (BLI) task is to predict its translation in the target language. Vulić and Moens (2015) proposed this task to test crosslingual word embeddings. The difficulty of this is that it is evaluated using the recall of the top ranked word. The model must be very discriminative in order to score well.

We build the CLWE for 3 language pairs: `it-en`, `es-en` and `nl-en`, using similar parameters setting with Vulić and Moens (2015).[5] The remaining tunable parameters in our system are δ from Equation (5), and the choice of algorithm for combining embeddings. We use the regularization technique from §4.3 for combining context and word embeddings with $\delta = 0.01$, and word embeddings $\mathbf{U}$ are used as the output for all experiments (but see comparative experiments in §8.)

Qualitative evaluation We jointly train the model to predict both w_i and the translation $\bar{w}_i$, combine $\mathbf{V}$ and $\mathbf{U}$ during training for each language pair. Table 2 shows the top 10 closest words in both source and target languages according to cosine similarity. Note that the model correctly identifies the translation in `en` as the top candidate, and the top 10 words in both source and target languages are highly related. This qualitative evaluation initially demonstrates the ability of our CLWE to capture both the bilingual and monolingual relationship.

Quantitative evaluation Table 3 shows our results compared with prior work. We reimple-

[5]Default learning rate of 0.025, negative sampling with 25 samples, subsampling rate of value $1e^{-4}$, embedding dimension $d = 200$, window size $cs = 48$ and run for 15 epochs.

Model	es-en		it-en		nl-en		Average	
	rec_1	rec_5	rec_1	rec_5	rec_1	rec_5	rec_1	rec_5
Gouws and Søgaard (2015) + Panlex	37.6	63.6	26.6	56.3	49.8	76.0	38.0	65.3
Gouws and Søgaard (2015) + Wikt	61.6	78.9	62.6	81.1	65.6	79.7	63.3	79.9
BilBOWA: Gouws et al. (2015)	51.6	-	55.7	-	57.5	-	54.9	-
Vulić and Moens (2015)	68.9	-	68.3	-	39.2	-	58.8	-
Our model (random selection)	41.1	62.0	57.4	75.4	34.3	55.5	44.3	64.3
Our model (EM selection)	67.3	79.5	66.8	82.3	64.7	82.4	66.3	81.4
+ Joint model	68.0	80.5	70.5	83.3	68.8	84.0	69.1	82.6
+ combine embeddings ($\delta = 0.01$)	74.7	85.4	80.8	90.4	79.1	90.5	78.2	88.8
+ lemmatization	**74.9**	**86.0**	**81.3**	**91.3**	**79.8**	**91.3**	**78.7**	**89.5**

Table 3: Bilingual Lexicon Induction performance from es, it, nl to en. Gouws and Søgaard (2015) + Panlex/Wikt is our reimplementation using Panlex/Wiktionary dictionary. All our models use Panlex as the dictionary. We reported the recall at 1 and 5. The best performance is bold.

gravedad$_{es}$		*tassazione*$_{it}$	
es	en	it	en
gravitacional	**gravity***	tasse	**taxation***
gravitatoria	gravitation*	fiscale	taxes
aceleracin	acceleration	tassa	tax*
gravitacin	non-gravitational	imposte	levied
inercia	inertia	imposta	fiscal
gravity	centrifugal	fiscali	low-tax
msugra	free-falling	l'imposta	revenue
centrífuga	gravitational	tonnage	levy
curvatura	free-fall	tax	annates
masa	newton	accise	evasion

Table 2: Top 10 closest words in both source and target language corresponding to es word *gravedad* (left) and it word *tassazione* (right). They have 15 and 4 dictionary translations respectively. The en words in the dictionary translations are marked with (*). The correct translation is in bold.

ment Gouws and Søgaard (2015) using Panlex and Wiktionary dictionaries. The result with Panlex is substantially worse than with Wiktionary. This confirms our hypothesis in §2. That is the context might be corrupted if we just randomly replace the training data with the translation from noisy dictionary such as Panlex.

Our model when randomly picking the translation is similar to Gouws and Søgaard (2015), using the Panlex dictionary. The biggest difference is that they replace the training data (both context and middle word) while we fix the context and only replace the middle word. For a high coverage yet noisy dictionary such as Panlex, our approach gives better average score. Comparing our two most basic models (EM selection and random selection), it is clear that the model using EM to select the translation outperforms random selection by a significant margin.

Our joint model, as described in equation (3) which predicts both target word and the translation, further improves the performance, especially for nl-en. We use equation (5) to combine both context embeddings $\mathbf{V}$ and word embeddings $\mathbf{U}$ for all three language pairs. This modification during training substantially improves the performance. More importantly, all our improvements are consistent for all three language pairs and both evaluation metrics, showing the robustness of our models.

Our combined model out-performed previous approaches by a large margin. Vulić and Moens (2015) used bilingual comparable data, but this might be hard to obtain for some language pairs. Their performance on nl-en is poor because their comparable data between en and nl is small. Besides, they also use POS tagger and lemmatizer to filter only *Noun* and reduce the morphology complexity during training. These tools might not be available for many languages. For a fairer comparison to their work, we also use the same Treetagger (Schmid, 1995) to lemmatize the output of our combined model before evaluation. Table 3 (+lemmatization) shows some improvements but minor. It demonstrates that our model is already good at disambiguating morphology. For example, the top 2 translations for es word *lenguas* in en are *languages* and *language* which correctly prefer the plural translation.

7 Monolingual Word Similarity

Now we consider the efficacy of our CLWE on monolingual word similarity. We evaluate on En-

	Model	WS-de	WS-en	RW-en
Baselines	Klementiev et al. (2012)	23.8	13.2	7.3
	Chandar A P et al. (2014)	34.6	39.8	20.5
	Hermann and Blunsom (2014)	28.3	19.8	13.6
	Luong et al. (2015)	47.4	49.3	25.3
	Gouws and Søgaard (2015)	67.4	71.8	31.0
Mono	CBOW	62.2	70.3	42.7
	+ combine	65.8	74.1	43.1
	Yih and Qazvinian (2012)	-	81.0	-
	Shazeer et al. (2016)	-	74.8	48.3
Ours	Our joint-model	59.3	68.6	38.1
	+ combine	**71.1**	**76.2**	**44.0**

Table 4: Spearman's rank correlation for monolingual similarity measurement on 3 datasets WS-de (353 pairs), WS-en (353 pairs) and RW-en (2034 pairs). We compare against 5 baseline crosslingual word embeddings. The best CLWE performance is bold. For reference, we add the monolingual CBOW with and without embeddings combination, Yih and Qazvinian (2012) and Shazeer et al. (2016) which represents the monolingual state-of-the-art results for WS-en and RW-en.

glish monolingual similarity on WordSim353 (WS-en), RareWord (RW-en) and German version of WordSim353 (WS-de) (Finkelstein et al., 2001; Luong et al., 2013; Luong et al., 2015). Each of those datasets contain many tuples $(w_1, w_2, \mathtt{s})$ where s is a scalar denoting the semantic similarity between w_1 and w_2 given by human annotators. Good system should produce the score correlated with human judgement.

We train the model as described in §4, which is the *combine embeddings* setting from Table 3. Since the evaluation involves de and en word similarity, we train the CLWE for en-de pair. Table 4 shows the performance of our combined model compared with several baselines. Our combined model out-performed both Luong et al. (2015) and Gouws and Søgaard (2015)[6] which represent the best published crosslingual embeddings trained on bitext and monolingual data respectively.

We also compare our system with the monolingual CBOW model trained on the monolingual data for each language, using the same parameter settings from earlier (§6). Surprisingly, our combined model performs better than the monolingual CBOW baseline which makes our result close to the monolingual state-of-the-art on each different dataset. However, the best monolingual methods use much larger monolingual corpora (Shazeer et al., 2016), WordNet or the output of commercial search engines (Yih and Qazvinian, 2012).

Next we explain the gain of our combined model compared with the monolingual CBOW model. First, we compare the combined model with the joint-model with respect to monolingual CBOW model (Table 4). It shows that the improvement seems mostly come from combining $\mathbf{V}$ and $\mathbf{U}$. If we apply the combining algorithm to the monolingual CBOW model (CBOW + combine), we also observe an improvement. Clearly most of the improvement is from combining $\mathbf{V}$ and $\mathbf{U}$, however our $\mathbf{V}$ and $\mathbf{U}$ are more complementary as the gain is more marked. Other improvements can be explained by the observation that a dictionary can improve monolingual accuracy through linking synonyms (Faruqui and Dyer, 2014). For example, since *plane*, *airplane* and *aircraft* have the same Italian translation *aereo*, the model will encourage those words to be closer in the embedding space.

8 Model selection

Combining context embeddings and word embeddings results in an improvement in both monolingual similarity and bilingual lexicon induction. In §4.3, we introduce several combination methods including post-processing (interpolation and concatenation) and during training (regularization). In this section, we justify our parameter and model choices.

We use en-it pair for tuning purposes, considering the value of γ in equation 4. Figure 2 shows the performances using different values of γ. The two extremes where $\gamma = 0$ and $\gamma = 1$ corresponds to no interpolation where we just use $\mathbf{U}$ or $\mathbf{V}$ respectively. As γ increases, the performance on WS-en increases yet BLI decreases. These results confirm our hypothesis in §4.3 that $\mathbf{U}$ is better at capturing bilingual relations and $\mathbf{V}$ is better at capturing monolingual relations. As a compromise, we choose $\gamma = 0.5$ in our experiments. Similarly, we tune the regularization sensitivity δ in equation (5) which combines embeddings space during training. We test $\delta = 10^{-n}$ with $n = \{0, 1, 2, 3, 4\}$ and us-

[6] trained using the Panlex dictionary

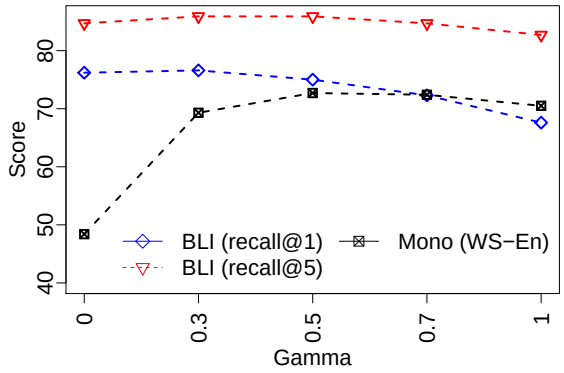


Figure 2: Performance of word embeddings interpolated using different values of γ evaluated using BLI (Recall@1, Recall@5) and English monolingual WordSim353 (WS-en).

	Model	BLI rec_1	BLI rec_5	Mono WS-en
Alone	Joint-model + $\mathbf{V}$	67.6	82.8	70.5
	Joint-model + $\mathbf{U}$	76.2	84.7	48.4
Combine	Interpolation $\left[\frac{\mathbf{V}+\mathbf{U}}{2}\right]$	75.0	85.9	72.7
	Concatenation	72.7	85.2	71.2
	Regularization + $\mathbf{V}$	80.3	89.8	45.9
	Regularization + $\mathbf{U}$	80.8	90.4	**74.8**
	Regularization + $\frac{\mathbf{V}+\mathbf{U}}{2}$	**80.9**	**91.1**	72.3

Table 5: Performance on en−it BLI and en monolingual similarity WordSim353 (WS-en) for various combining algorithms mentioned in §4.3 w.r.t just using $\mathbf{U}$ or $\mathbf{V}$ alone (after joint-training). We use $\gamma = 0.5$ for interpolation and $\delta = 0.01$ for regularization with the choice of $\mathbf{V}$, $\mathbf{U}$ or interpolation of both $\frac{\mathbf{V}+\mathbf{U}}{2}$ for the output. The best scores are bold.

ing $\mathbf{V}$, $\mathbf{U}$ or the interpolation of both $\frac{\mathbf{V}+\mathbf{U}}{2}$ as the learned embeddings, evaluated on the same BLI and WS-en. We select $\delta = 0.01$.

Table 5 shows the performance with and without using combining algorithms mentioned in §4.3. As the compromise between both monolingual and crosslingual tasks, we choose regularization + $\mathbf{U}$ as the combination algorithm. All in all, we apply the regularization algorithm for combining $\mathbf{V}$ and $\mathbf{U}$ with $\delta = 0.01$ and $\mathbf{U}$ as the output for all language pairs without further tuning.

9 Crosslingual Document Classification

In this section, we evaluate our CLWE on a downstream crosslingual document classification (CLDC) task. In this task, the document classifier is trained on a source language and then applied directly to classify a document in the target language. This is convenient for a target low-resource language where we do not have document annotations. The experimental setup is the same as Klementiev et al. (2012)[7] with the training and testing data sourced from Reuter RCV1/RCV2 corpus (Lewis et al., 2004).

Model	en → de	de → en
MT baseline	68.1	67.4
Klementiev et al. (2012)	77.6	71.1
Gouws et al. (2015)	86.5	75.0
Kočiský et al. (2014)	83.1	75.4
Chandar A P et al. (2014)	**91.8**	74.2
Hermann and Blunsom (2014)	86.4	74.7
Luong et al. (2015)	88.4	**80.3**
Our model	86.3	76.8

Table 6: CLDC performance for both en → de and de → en direction for many CLWE. The MT baseline uses phrase-based statistical machine translation to translate the source language to target language (Klementiev et al., 2012). The best scores are bold.

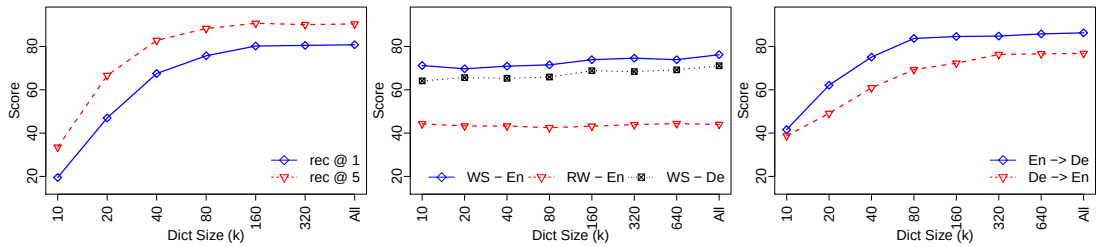
The documents are represented as the bag of word embeddings weighted by tf.idf. A multi-class classifier is trained using the average perceptron algorithm on 1000 documents in the source language and tested on 5000 documents in the target language. We use the CLWE, such that the document representation in the target language embeddings is in the same space with the source language.

We build the en-de CLWE using combined models as described in section §4. Following prior work, we also use monolingual data[8] from the RCV1/RCV2 corpus (Klementiev et al., 2012; Gouws et al., 2015; Chandar A P et al., 2014).

Table 6 shows the CLDC results for various CLWE. Despite its simplicity, our model achieves competitive performance. Note that aside from our model, all other models in Table 6 use a large bitext (Europarl) which may not exist for many low-resource languages, limiting their applicability.

[7] The data split and code are kindly provided by the authors.

[8] We randomly sample documents in RCV1 and RCV2 corpora and selected around 85k documents to form 400k monolingual sentences for both en and de. For each document, we perform basic pre-processing including: lower-casing, remove html tags and tokenization. These monolingual data are then concatenated with the monolingual data from Wikipedia to form the final training data.



(a) BLI (b) Mono Similarity (c) CLDC

Figure 3: Learning curve showing how task scores increase with increasing dictionary size; showing bilingual lexicon induction (BLI) task (left), monolingual similarity (center) and crosslingual document classification (right). BLI is trained on `en-it`, and monolingual similarity and CLDC are trained on `en-de`.

10 Low-resource languages

Our model exploits dictionaries, which are more widely available than parallel corpora. However the question remains as to how well this performs of a real low-resource language, rather than a simulated condition like above, whereupon the quality of the dictionary is likely to be worse. To test this, we evaluation on Serbian, a language with few annotated language resources. Table 1 shows the relative size of monolingual data and dictionary for `en-sr` compared with other language pairs. Both the Serbian monolingual data and the dictionary size is more than 10 times smaller than other language pairs. We build the `en-sr` CLWE using our best model (joint + combine) and evaluate on the bilingual word induction task using 939 gold translation pairs.[9] We achieved recall score of 35.8% and 45.5% at 1 and 5 respectively. Although worse than the earlier results, these numbers are still well above chance.

We can also simulate low-resource setting using our earlier datasets. For estimating the performance loss on all three tasks, we down sample the dictionary for `en-it` and `en-de` based on `en` word frequency. Figure 3 shows the performance with different dictionary sizes for all three tasks. The monolingual similarity performance is very similar across various sizes. For BLI and CLDC, dictionary size is more important, although performance levels off at around 80k dictionary pairs. We conclude that this size is sufficient for decent performance.

[9]The `sr`→`en` translations are sourced from Google Translate by translating one word at a time, followed by manually verification, after which 61 translation pairs were ruled out as being bad or questionable.

11 Conclusion

Previous CLWE methods often impose high resource requirements yet have low accuracy. We introduce a simple framework based on a large noisy dictionary. We model polysemy using EM translation selection during training to learn bilingual correspondences from monolingual corpora. Our algorithm allows to train on massive amount of monolingual data efficiently, representing monolingual and bilingual properties of language. This allows us to achieve state-of-the-art performance on bilingual lexicon induction task, competitive result on monolingual word similarity and crosslingual document classification task. Our combination techniques during training, especially using regularization, are highly effective and could be used to improve monolingual word embeddings.

Acknowledgments

This work was conducted during Duong's internship at IBM Research – Tokyo and partially supported by the University of Melbourne and National ICT Australia (NICTA). We are grateful for support from NSF Award 1464553 and the DARPA/I2O, Contract No. HR0011-15-C-0114. We thank Yuta Tsuboi and Alvin Grissom II for helpful discussions, Jan Šnajder for helping with `sr-en` evaluation.

References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual nlp. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria, August. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Sarath Chandar A P, Stanislas Lauly, Hugo Larochelle, Mitesh Khapra, Balaraman Ravindran, Vikas C Raykar, and Amrita Saha. 2014. An autoencoder approach to learning bilingual word representations. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1853–1861. Curran Associates, Inc.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 600–609.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 845–850, Beijing, China. Association for Computational Linguistics.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. 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)*, pages 334–343, Beijing, China, July. Association for Computational Linguistics.

Manaal Faruqui and Chris Dyer. 2014. 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, Gothenburg, Sweden, April. Association for Computational Linguistics.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 406–414, New York, NY, USA. ACM.

Stephan Gouws and Anders Søgaard. 2015. Simple task-specific bilingual word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1386–1390, Denver, Colorado, May–June. Association for Computational Linguistics.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. Bilbowa: Fast bilingual distributed representations without word alignments. In David Blei and Francis Bach, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 748–756. JMLR Workshop and Conference Proceedings.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 58–68, Baltimore, Maryland, June. Association for Computational Linguistics.

David Kamholz, Jonathan Pool, and Susan Colowick. 2014. Panlex: Building a resource for panlingual lexical translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3145–50, Reykjavik, Iceland. European Language Resources Association (ELRA).

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. In *Proceedings of COLING 2012*, pages 1459–1474, Mumbai, India, December. The COLING 2012 Organizing Committee.

Tomáš Kočiský, Karl Moritz Hermann, and Phil Blunsom. 2014. Learning bilingual word representations by marginalizing alignments. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 224–229, Baltimore, Maryland, June. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as a factorization. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2177–2185.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397, December.

Thang Luong, Richard Socher, and Christopher D. Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning, CoNLL 2013, Sofia, Bulgaria, August 8-9, 2013*, pages 104–113.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Bilingual word representations with monolingual quality in mind. In *NAACL Workshop on Vector Space Modeling for NLP*, Denver, United States.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013a.

Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Philip Resnik and David Yarowsky. 1999. Distinguishing systems and distinguishing senses: New evaluation methods for word sense disambiguation. *Nat. Lang. Eng.*, 5(2):113–133, June.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to german. In *In Proceedings of the ACL SIGDAT-Workshop*, pages 47–50.

Noam Shazeer, Ryan Doherty, Colin Evans, and Chris Waterson. 2016. Swivel: Improving embeddings by noticing what's missing. *CoRR*, abs/1602.02215.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October. Association for Computational Linguistics.

Anders Søgaard, Željko Agić, Héctor Martínez Alonso, Barbara Plank, Bernd Bohnet, and Anders Johannsen. 2015. Inverted indexing for cross-lingual nlp. 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)*, pages 1713–1722, Beijing, China, July. Association for Computational Linguistics.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 477–487. Association for Computational Linguistics.

Ivan Vulić and Marie-Francine Moens. 2015. Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 719–725, Beijing, China, July. Association for Computational Linguistics.

Min Xiao and Yuhong Guo, 2014. *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, chapter Distributed Word Representation Learning for Cross-Lingual Dependency Parsing, pages 119–129. Association for Computational Linguistics.

David Yarowsky and Grace Ngai. 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, NAACL '01, pages 1–8, Pittsburgh, Pennsylvania.

Wen-tau Yih and Vahed Qazvinian. 2012. Measuring word relatedness using heterogeneous vector space models. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 616–620, Stroudsburg, PA, USA. Association for Computational Linguistics.

3.1.6 EACL 2017 - if accept

Multilingual Training of Crosslingual Word Embeddings

Anonymous EACL submission

Abstract

Crosslingual word embeddings represent lexical items from different languages using the same vector space, enabling crosslingual transfer. Most prior work constructs embeddings for a pair of languages, with English on one side. We investigate methods for building high quality crosslingual word embeddings for many languages in a unified vector space. In this way, we can exploit and combine strength of many languages. We obtained high performance on bilingual lexicon induction, monolingual similarity and crosslingual document classification tasks.

1 Introduction

Monolingual word embeddings have facilitated advances in many natural language processing tasks, such as natural language understanding (Collobert and Weston, 2008), sentiment analysis (Socher et al., 2013), and dependency parsing (Dyer et al., 2015). Crosslingual word embeddings represent words from several languages in the same low dimensional space. They are helpful for multilingual tasks such as machine translation (Brown et al., 1993) and bilingual named entity recognition (Wang et al., 2013). Crosslingual word embeddings can also be used in transfer learning, where the source model is trained on one language and applied directly to another language; this is suitable for the low-resource scenario (Yarowsky and Ngai, 2001; Duong et al., 2015; Das and Petrov, 2011; Täckström et al., 2012).

Most prior work on building crosslingual word embeddings focuses on a pair of languages. English is usually on one side, thanks to the wealth of available English resources. However, it is highly desirable to have a crosslingual word embeddings for many languages so that different relations can be exploited.[1] For example, since Italian and Spanish are similar, they are excellent candidates for transfer learning. However, there are scarce parallel resources between Italian and Spanish for directly building bilingual word embeddings. Our multilingual word embeddings, on the other hand, map both Italian and Spanish to the same space without using any direct bilingual signal between them. Moreover, multilingual word embeddings are also crucial for multilingual applications such as multi-source machine translation (Zoph and Knight, 2016), multi- source transfer dependency parsing (McDonald et al., 2011).

We propose several algorithms to map bilingual word embeddings to the same vector space, either during training or during post-processing. We apply linear transformation to map the English side of each pretrained crosslingual word embedding to the same space. We also extend Duong et al. (2016), which used a dictionary to learn bilingual word embeddings. We modify the objective function to jointly build multilingual word embeddings during training. Unlike most prior work which focuses on downstream applications, we measure the quality of our multilingual word embeddings in three ways: bilingual lexicon induction, monolingual word similarity, and crosslingual document classification tasks. Relative to a benchmark of training on each language pair separately and various published multilingual word embeddings, we achieved high performance for all the tasks.

In this paper we make the following contributions: (a) novel algorithms for post-hoc combination of multiple bilingual word embeddings,

[1] From here on we refer to crosslingual word embeddings for a pair of languages and multiple languages as *bilingual word embeddings* and *multilingual word embeddings* respectively.

applicable to any pretrained bilingual model; (b) a method for jointly learning multilingual word embeddings, extending Duong et al. (2016), to jointly train over monolingual corpora in several languages; (c) Achieving competitive results in bilingual, monolingual and crosslingual transfer settings.

2 Related work

Crosslingual word embeddings are typically based on co-occurrence statistics from parallel text (Luong et al., 2015; Gouws et al., 2015; Chandar A P et al., 2014; Klementiev et al., 2012; Kočiský et al., 2014; Huang et al., 2015). Other work uses more widely available resources such as comparable data (Vulić and Moens, 2015) and shared Wikipedia entries (Søgaard et al., 2015). However, those approaches rely on data from Wikipedia, and it is non-trivial to extend them to languages that are not covered by Wikipedia. Dictionaries are another source of bilingual signal, with the advantage of high coverage. Multilingual lexical resources such as PanLex (Kamholz et al., 2014) and Wiktionary[2] cover thousands of languages, and have been used to construct high performance crosslingual word embeddings (Mikolov et al., 2013a; Xiao and Guo, 2014; Faruqui and Dyer, 2014).

Previous work mainly focuses on building word embeddings for a pair of languages, typically with English on one side, with the exception of Coulmance et al. (2015), Søgaard et al. (2015) and Ammar et al. (2016). Coulmance et al. (2015) extend the bilingual skipgram model from Luong et al. (2015), training jointly over many languages using the Europarl corpora. We also compare our models with an extension of Huang et al. (2015) adapted for multiple languages also using bilingual corpora. However, parallel data is an expensive resource and using parallel data seems to under-perform on the bilingual lexicon induction task (Vulić and Moens, 2015). While Coulmance et al. (2015) use English as the pivot language, Søgaard et al. (2015) learn multilingual word embeddings for many languages using Wikipedia entries which are the same for many languages. However, their approach is limited to languages covered in Wikipedia and seems to under-perform other methods. Ammar et al. (2016) propose two algorithms namely MultiCluster and MultiCCA for multilingual word embeddings using set of bilingual dictionaries. MultiCluster first builds the graph where nodes are lexicon and edges are translations. Each cluster in this graph is an anchor point for building multilingual word embeddings. MultiCCA is an extension of Faruqui and Dyer (2014), performing canonical correlation analysis (CCA) for multiple languages using English as the pivot language. A shortcoming of MultiCCA is that it ignores polysemous translations by retaining on only one-to-one dictionary pairs (Gouws et al., 2015), disregarding much information. As a simple solution, we propose a simple post-hoc method by mapping the English parts of each bilingual word embedding to each other. In this way, the mapping is always exact and one-to-one.

Duong et al. (2016) constructed bilingual word embeddings based on monolingual data and PanLex. In this way, their approach can be applied to more languages as PanLex covers more than a thousand languages. They solve the polysemy problem by integrating an EM algorithm for dictionary selection. Relative to many previous crosslingual word embeddings, their joint training algorithm achieved state-of-the-art performance for the bilingual lexicon induction task, performing significantly better on monolingual similarity and achieving a competitive result on cross lingual document classification. Here we also adopt their approach, and extend it to multilingual embeddings.

2.1 Base model for bilingual embeddings

We briefly describe the base model (Duong et al., 2016), an extension of the continuous bag-of-word (CBOW) model (Mikolov et al., 2013a) with negative sampling. The original objective function is

$$\sum_{i \in D} \left(\log \sigma(\mathbf{u}_{w_i}^\top \mathbf{h}_i) + \sum_{j=1}^{p} \log \sigma(-\mathbf{u}_{w_{ij}}^\top \mathbf{h}_i) \right), \quad (1)$$

where D is the training data, $\mathbf{h}_i = \frac{1}{2k} \sum_{j=-k;j\neq 0}^{k} \mathbf{v}_{w_{i+j}}$ is a vector encoding the context over a window of size k centred around position i, $\mathbf{V}$ and $\mathbf{U} \in \mathbb{R}^{|V_e| \times d}$ are learned matrices referred to as the context and centre word embeddings where V_e is the vocabulary and p is the number of negative examples randomly drawn from a noise distribution, $w_{ij} \sim P_n(w)$.

Duong et al. (2016) extend the CBOW model for application to two languages, using monolingual text in both languages and a bilingual dictio-

[2] wiktionary.org

nary. Their approach augments CBOW by generating not only the middle word, but also its translation in the other language. This is done by first selecting a translation $\bar{w}_i$ from dictionary for the middle word w_i, based on the cosine distance between the context h_i and the context embeddings $\mathbf{V}$ for each candidate foreign translation. In this way source monolingual training contexts must generate both source and target words, and similarly target monolingual training contexts also generate source and target words. Overall this results in compatible word embeddings across the two languages, and highly informative nearest neighbours across the two languages. This leads to the new objective function

$$\sum_{i \in D_s \cup D_t} \Big(\log \sigma(\mathbf{u}_{w_i}^\top \mathbf{h}_i) + \log \sigma(\mathbf{u}_{\bar{w}_i}^\top \mathbf{h}_i) + \sum_{j=1}^{p} \log \sigma(-\mathbf{u}_{w_{ij}}^\top \mathbf{h}_i) \Big) + \delta \sum_{w \in V_s \cup V_t} \|\mathbf{u}_w - \mathbf{v}_w\|_2^2 , \quad (2)$$

where D_s and D_t are source and target monolingual data, V_s and V_t are source and target vocabulary. Comparing with the CBOW objective function in Equation (1), this represents two additions: the translation cross entropy $\log \sigma(\mathbf{u}_{\bar{w}_i}^\top \mathbf{h}_i)$, and a regularisation term $\sum_{w \in V_s \cup V_t} \|\mathbf{u}_w - \mathbf{v}_w\|_2^2$ which penalises divergence between context and center word embedding vectors for each word type, which was shown to improve the embedding quality (Duong et al., 2016).

3 Post-hoc Unification of Embeddings

Our goal is to learn multilingual word embeddings over more than two languages. One simple way to do this is to take several learned bilingual word embeddings which share a common target language (here, English), and map these into a shared space (Mikolov et al., 2013a; Faruqui and Dyer, 2014). In this section we propose post-hoc methods, however in §4 we develop an integrated multilingual method using joint inference.

Formally, the input to the posthoc combination methods are a set of n pre-trained bilingual word embedding matrices, i.e., $C_i = \{(E_i, F_i)\}$ with $i \in \mathbf{F}$ is the set of foreign languages (not English), $E_i \in \mathbb{R}^{|V_{e_i}| \times d}$ are the English word embeddings and $F_i \in \mathbb{R}^{|V_{f_i}| \times d}$ are foreign language word embeddings for language i, with V_{e_i} and V_{f_i} being the English and foreign language vocabularies and

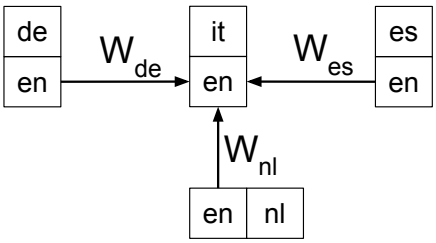


Figure 1: Examples of unifying four bilingual word embeddings between `en` and `it`, `de`, `es`, `nl` to the same space using post-hoc linear transformation.

d is the embedding dimension. These bilingual embeddings can be produced by any method, e.g., those discussed in §2.

Linear Transformation. The simplest method is to learn a linear transformation which maps the English part of each bilingual word embedding into the same space (inspired by Mikolov et al. (2013a)), as illustrated in Figure 1. One language pair is chosen as the pivot, `en-it` in this example, and the English side of the other language pairs, `en-de`, `en-es`, `en-nl`, are mapped to closely match the English side of the pivot, `en-it`. This is achieved through learning linear transformation matrices for each language, $W_{\texttt{de}}, W_{\texttt{es}}$ and $W_{\texttt{nl}}$, respectively, where each $W_i \in \mathbb{R}^{d \times d}$ is learned to minimize the objective function $\|E_i \times W_i - E_{pivot}\|_2^2$ where E_{pivot} is the English embedding of the pivot pair, `en-it`.

Each foreign language f_i is then mapped to the same space using the learned matrix W_i, i.e., $F_i' = F_i \times W_i$. These projected foreign embeddings are then used in evaluation, along with the English side of the language pair with largest English vocabulary coverage, i.e., biggest $|V_{e_i}|$. Together these embeddings allow for querying of monolingual and cross-lingual word similarity, and multilingual transfer of trained models.

The advantage of this approach is that it is very fast and simple to train, since the objective function is strictly convex and has closed form solution. Moreover, unlike Mikolov et al. (2013a) who learn the projection from source to a target language, we learn the projection from English to English, thus, do not require a dictionary, sidestepping the polysemy problem.[3]

[3] A possible criticism of this approach is that linear transformation is not powerful enough for the required mapping. We experimented with non-linear transformations but did not

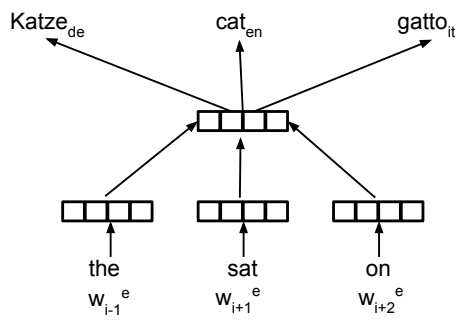


Figure 2: Examples of our multilingual joint training model without mapping for learning multilingual embeddings for three languages `en, it, de` using joint inference.

4 Multilingual Joint Training

Instead of combining bilingual word embeddings at the post-processing step, it might be more beneficial to do it during training, such that languages can interact with each other more freely. We extend the method in §2.1 to jointly learn the multilingual word embeddings during training. The input to the model is the combined monolingual data for each language and the set of dictionaries between any language pair.

We modify the base model (Duong et al., 2016) to accommodate more languages. For the first step, instead of just predicting the translation for a single target language, we predict the translation for all languages in the dictionary. That is, we compute $w_i^f = \text{argmax}_{w \in \text{dict}_e^f(w_i^e)} \cos(\mathbf{v}_w, \text{context})$, which is the best translation in language f of source word w_i^e in language e, given the bilingual dictionary dict_e^f and the context. For the second step, we jointly predict word w_i^e and all translations w_i^f in all foreign languages $f \in \mathbf{T}$ that we have dictionary dict_e^f as illustrated in Figure 2. The English word *cat* might have several translations in German {*Katze, Raupe, Typ*} and Italian {*gatto, gatta*}. In the first step, we select the closest translation given the context for each language, i.e. *Katze* and *gatto* for German and Italian respectively. In the second step, we jointly predict the English word *cat* together with selected translations *Katze* and *gatto* using the following modified objective function:

$$\mathcal{O} = \sum_{i \in D_{all}} \Bigg(\log \sigma(\mathbf{u}_{w_i^e}^\top \mathbf{h}_i) + \sum_{f \in \mathbf{T}} \log \sigma(\mathbf{u}_{w_i^f}^\top \mathbf{h}_i) + \sum_{j=1}^{p} \log \sigma(-\mathbf{u}_{w_{ij}}^\top \mathbf{h}_i) \Bigg) + \delta \sum_{w \in V_{all}} \|\mathbf{u}_w - \mathbf{v}_w\|_2^2 , \quad (3)$$

where D_{all} and V_{all} are the combined monolingual data and vocabulary for all languages. Each of the p negative samples, w_{ij}, are sampled from a unigram model over the combined vocabulary V_{all}.

Explicit mapping. As we keep adding more languages to the model, the hidden layer in our model – shared between all languages – might not be enough to accommodate all languages. However, we can combine the strength of the linear transformation proposed in §3 to our joint model as described in Equation (3). We explicitly learn the linear transformation jointly during training by adding the following regularization term to the objective function:

$$\mathcal{O}' = \mathcal{O} + \alpha \sum_{i \in D_e} \sum_{f \in \mathbf{F}} \|\mathbf{u}_{w_i^f} W_f - \mathbf{u}_{w_i^e}\|_2^2 , \quad (4)$$

where D_e is the English monolingual data (since we use English as the pivot language), $\mathbf{F}$ is the set of foreign languages (not English), $W_f \in \mathbb{R}^{d \times d}$ is the linear transformation matrix, and α controls the contribution of the regularization term and will be tuned in §6.[4] Thus, the set of learned parameters for the model are the word and context embeddings $\mathbf{U}, \mathbf{V}$ and $|\mathbf{F}|$ linear transformation matrices, $\{W_f\}_{f \in \mathbf{F}}$. After training is finished, we linearly transform the foreign language embeddings with the corresponding learned matrix W_f, such that all embeddings are in the same space.

5 Experiment Setup

Our experimental setup is based on that of Duong et al. (2016). We use the first 5 million sentences from tokenized monolingual data from the Wikipedia dump from Al-Rfou et al. (2013).[5] The dictionary is from PanLex which covers more

observe any improvements. Faruqui and Dyer (2014) extended Mikolov et al. (2013a) as they projected both source and target languages to the same space using canonical correlation analysis (CCA). We also adopted this approach for multilingual environment by applying multi-view CCA to map English part of each pre-trained bilingual word embeddings to the same space. However, we only observe minor improvement.

[4] For an efficient implementation, we apply this constraint to only 10% of English monolingual data.

[5] We will use the whole data if there are less than 5 million sentences.

	Model	it-en		es-en		nl-en		nl-es		Average	
		rec_1	rec_5	rec_1	rec_5	rec_1	rec_5	rec_1	rec_5	rec_1	rec_5
Baselines	MultiCluster	35.6	64.3	34.9	62.5	-	-	-	-	-	-
	MultiCCA	63.4	77.3	58.5	72.7	-	-	-	-	-	-
	MultiSkip	57.6	68.5	49.3	58.9	-	-	-	-	-	-
	MultiTrans	72.1	83.1	71.5	82.2	-	-	-	-	-	-
Ours	Linear	78.5	88.2	69.3	81.8	74.9	87.0	66.3	79.7	72.2	84.2
	Joint	79.4	89.7	73.6	84.6	76.6	89.6	69.4	82.0	74.7	86.5
	+ Mapping	**81.6**	**90.5**	**74.6**	**87.4**	**77.9**	**91.4**	**71.6**	**83.5**	**76.4**	**88.2**
	BiWE	80.8	90.4	74.7	85.4	79.1	90.5	71.7	80.7	76.6	86.7

Table 1: Bilingual lexicon induction performance for four pairs. Bilingual word embeddings (BiWE) is the state-of-the-art result from Duong et al. (2016) where each pair is trained separately. Our proposed methods including linear transformation (Linear), joint prediction as in Equation (3) (Joint) and joint prediction with explicit mapping as in Equation (4) (+mapping). We report recall at 1 and 5 with respect to four baseline multilingual word embeddings. The best scores for multilingual models are shown in bold.

than 1,000 language varieties. We build multilingual word embeddings for 5 languages (`en`, `it`, `es`, `nl`, `de`) jointly using the same parameters as Duong et al. (2016).[6] During training, for a fairer comparison, we only use dictionaries between English and each target language. However, it is straight-forward to incorporate any dictionary between any pair of languages into our model. The pre-trained bilingual word embeddings for the post-processing experiment in §3 are also from Duong et al. (2016). In the following sections, we evaluate the performance of our multilingual word embeddings in comparison with bilingual word embeddings and previous published multilingual word embeddings (MultiCluster, MultiCCA, MultiSkip and MultiTrans) for three tasks: bilingual lexicon induction (§6), monolingual similarity (§7) and crosslingual document classification (§8). MultiCluster and MultiCCA are the models proposed from Ammar et al. (2016) trained on monolingual data using bilingual dictionaries extracted from aligning Europarl corpus. MultiSkip is the reimplementation of the multilingual skipgram model from Coulmance et al. (2015). MultiTrans is the multilingual version of the translation invariance model from Huang et al. (2015). Both MultiSkip and MultiTrans are trained directly on parallel data from Europarl. All the previous work is trained with 512 dimensions on 12 languages acquired directly from Ammar et al. (2016).

6 Bilingual Lexicon Induction

In this section we evaluate our multilingual models on the bilingual lexicon induction (BLI) task, which tests the bilingual quality of the model. Given a word in the source language, the model must predict the translation in the target language. We report recall at 1 and 5 for the various models listed in Table 1. The evaluation data for `it-en`, `es-en`, `nl-en` was manually constructed (Vulić and Moens, 2015). We extend the evaluation for `nl-es` pair which do not involve English.[7]

The BiWE results for pairs involving English in Table 1 are from Duong et al. (2016) which is the state-of-the-art in this task. For the `nl-es` pair, we cannot build bilingual word embeddings, since we do not have dictionary between them. Instead, we use English as the pivot language. To get the `nl-es` translation, we use two bilingual embeddings of `nl-en` and `es-en` from Duong et al. (2016). We get the best English translation for the Dutch word, and get the top 5 Spanish translations with respect to the English word. This simple trick performs surprisingly well, probably

[6] Default learning rate of 0.025, negative sampling with 25 samples, subsampling rate of value $1e^{-4}$, embedding dimension $d = 200$, window size 48, run for 15 epochs and $\delta = 0.01$ for combining word and context embeddings.

[7] We build 1,000 translation pairs for `nl-es` pair with the source word from Vulić and Moens (2015) and ground truth candidates from Google Translate but manually verified.

because bilingual word embeddings involving English such as nl-en and es-en from Duong et al. (2016) are very accurate.

For the linear transformation, we use the first pair it-en as the pivot and learn to project es-en, de-en, nl-en pairs to this space as illustrated in Figure 1. We use English part ($E'_{biggest}$) from transformed de-en pair as the English output. Despite simplicity, linear transformation performs surprisingly well.

Our joint model to predict all target languages simultaneously as described in Equation (3) performs consistently better in contrast with linear transformation at all language pairs. The joint model with explicit mapping as described in Equation (4) can be understood as the combination of joint model and linear transformation. For this model, we need to tune α in Equation (4). We tested α with value in range $\{10^{-i}\}_{i=0}^{5}$ using es-en pair on BLI task. $\alpha = 0.1$ gives the best performance. To avoid over-fitting, we use the same value of α for all experiments and all other pairs. With this tuned value α, our joint model with mapping clearly outperforms other propose methods on all pairs. More importantly, this result is substantially better than all the baselines across four language pairs and two evaluation metrics. Comparing with the state-of-the-art (BiWE), our final model (joint + mapping) achieves relatively better result, especially for recall at 5.

7 Monolingual similarity

The multilingual word embeddings should preserve the monolingual property of the languages. We evaluate using the monolingual similarity task proposed in Luong et al. (2015). In this task, the model is asked to give the similarity score for a pair of words in the same language. This score is then measured against human judgment. Following Duong et al. (2016), we evaluate on three datasets, WordSim353 (WS-en), RareWord (RW-en), and the German version of WordSim353 (WS-de) (Finkelstein et al., 2001; Luong et al., 2013; Luong et al., 2015).

Table 2 shows the result of our multilingual word embeddings with respect to several baselines. The trend is similar to bilingual lexicon induction task. Linear transformation performs surprisingly well. Our joint model achieves similar result with linear transformation (better on WS-de but worse on WS-en and RW-en). Our joint model with explicit mapping regains the drop and performs slightly better than linear transformation. More importantly, this model is substantially better than all baselines, except for MultiTrans on RW-en dataset. This can probably be explained by the low coverage of MultiTrans on this dataset. Our final model (Joint + Mapping) is also close to the best bilingual word embeddings (BiWE) performance by Duong et al. (2016).

	Model	WS-de	WS-en	RW-en
Baselines	MultiCluster	51.0 [98.3]	53.9 [100]	38.1 [57.6]
	MultiCCA	60.2 [99.7]	66.3 [100]	43.1 [71.1]
	MultiSkip	48.4 [96.6]	51.2 [99.7]	33.9 [55.4]
	MultiTrans	56.4 [92.6]	61.1 [97.2]	**51.1** [23.1]
Ours	Linear	67.5 [99.4]	**74.7** [100]	45.4 [75.5]
	Joint	68.5 [99.4]	74.6 [100]	43.8 [75.5]
	Joint + Mapping	**70.4** [99.4]	74.4 [100]	45.1 [75.5]
	BiWE	71.1 [99.4]	76.2 [100]	44.0 [75.5]

Table 2: Spearman's rank correlation for monolingual similarity measurement for various models on 3 datasets WS-de (353 pairs), WS-en (353 pairs) and RW-en (2034 pairs). We compare against 4 baseline multilingual word embeddings. BiWE is the result from Duong et al. (2016) where each pair is trained separately which serves as the reference for the best bilingual word embeddings. The best results for multilingual word embeddings are shown in bold. Numbers in square brackets are the coverage percentage.

8 Crosslingual Document Classification

In the previous sections, we have shown that our methods for building multilingual word embeddings, either in the post-processing step or during training, preserved high quality bilingual and monolingual relations. In this section, we demonstrate the usefulness of multi-language crosslingual word embeddings through the crosslingual document classification (CLDC) task.

This task exploits transfer learning where the document classifier is trained on the source language and test on the target language. The source language classifier is transferred to the target language using crosslingual word embeddings as document is represented as sum of bag-of-word embeddings weighted by *tf.idf*. This setting is useful for target low-resource languages where the annotated data is insufficient.

The train and test data are from multilin-

		en→de	de→en	it→de	it→es	en→es	Avg
Baselines	MultiCluster	**92.9**	69.1	79.1	**81.0**	**63.1**	**77.0**
	MultiCCA	69.2	50.7	83.1	79.0	45.3	65.5
	MultiSkip	79.9	63.5	71.8	76.3	60.4	70.4
	MultiTrans	87.7	75.2	70.4	64.4	56.1	70.8
Ours	Linear	83.8	75.7	74.8	67.3	57.4	71.8
	Joint	86.2	75.7	82.3	70.7	56.0	74.2
	Joint + Mapping	89.5	**81.6**	**84.3**	74.1	53.9	76.7
Bilingual	Luong et al. (2015)	88.4	**80.3**	-	-	-	-
	Chandar A P et al. (2014)	**91.8**	74.2	-	-	-	-
	Duong et al. (2016)	86.3	76.8	-	-	53.8	-

Table 3: Crosslingual document classification accuracy for various model. Chandar A P et al. (2014) and Luong et al. (2015) achieved state-of-the-art result for en→de and de→en respectively, served as the reference. The best results for bilingual and multilingual word embeddings are bold.

gual RCV1/RCV2 corpus (Lewis et al., 2004) where each document is annotated with labels from 4 categories: CCAT (Corporate/Industrial), ECAT (Economics), GCAT (Government/Social) and MCAT (Markets). We extend the evaluation from Klementiev et al. (2012) to cover more language pairs. We use the same data split for en→de and de→en pairs but additionally construct the train and test data for it→de, it→es and en→es. For each pair, we use 1,000 documents in the source language as the training data and 5,000 documents in the target language as the testing data. The train data is randomly sampled, but the test data (for es) is evenly balanced among labels.

Table 3 shows the accuracy for the CLDC task for many pairs and models with respect to the baselines. For all bilingual models (Duong et al., 2016; Luong et al., 2015; Chandar A P et al., 2014), the bilingual word embeddings are constructed for each pair separately. In this way, they can only get the pairs involving English since there are much bilingual resource involving English in one side. For all our models including Linear, Joint and Joint + Mapping, the embedding space is available for multiple languages, that is why we can exploit different relation such as it→es. This is the motivation for this paper. Take es as an example, assuming that we want to build document classification for es but do not have any annotation. It is common to build en-es crosslingual word embeddings for transfer learning, however, it can only achieve 53.8 % accuracy. Nevertheless, if we use it as the source, we can get 81.0% accuracy. This is motivated by the fact that it and es are very similar.

The trend observed in Table 3 is consistent with previous observation. Linear transformation performs well. Joint training performs better especially for it→de pair. The joint model with explicit mapping is generally our best model, even better than the base bilingual model from Duong et al. (2016). The de→en result is even better than the state-of-the-art reported in Luong et al. (2015). Our final model (Joint + Mapping) achieved competitive results compared with four strong baseline multilingual word embeddings, achieving best results for two out of five pairs. Moreover, the best scores for each language pairs are all from multilingual training, emphasizing the advantages over bilingual training.

9 Analysis

Mikolov et al. (2013b) showed that monolingual word embeddings capture some analogy relations such as $\vec{\text{Paris}} - \vec{\text{France}} + \vec{\text{Italy}} \approx \vec{\text{Rome}}$. It seems that in our multilingual embeddings, these relations still hold. Table 4 shows some examples of such relations where each word in the analogy query is in different languages.

All our baselines (MultiCluster, MultiCCA, MultiSkip, MultiTrans) are trained using different datasets. While MultiSkip and MultiTrans are trained on parallel corpora, MultiCluster and MultiCCA use monolingual corpora and bilingual dictionaries which are similar with our proposed

chico$_{es}$ - bruder$_{de}$ + sorella$_{it}$ (boy - brother + sister)	ehemann$_{de}$ - padre$_{es}$ + madre$_{it}$ (husband - father + mother)	principe$_{it}$ - junge$_{de}$ + meisje$_{nl}$ (prince - boy + girl)
chica$_{es}$ (girl)	**echtgenote$_{nl}$** (wife)	**principessa$_{it}$** (princess)
ragazza$_{it}$ (girl)	**moglie$_{it}$** (wife)	**princess$_{en}$**
meisje$_{nl}$ (girl)	her$_{en}$	**princesa$_{es}$** (princess)
girl$_{en}$	marito$_{it}$ (husband)	príncipe$_{es}$ (prince)
mädchen$_{de}$ (girl)	haar$_{nl}$ (her)	**prinzessin$_{de}$** (princess)

Table 4: Top five closest words in our embeddings for multilingual word analogy. The transliteration is provided in parentheses. The correct output is bold.

	Tasks	MultiCluster	MultiCCA	Our model
Extrinsic	multilingual Dependency Parsing	61.0	58.7	**61.2**
	multilingual Document Classification	**92.1**	**92.1**	90.8
Intrinsic	monolingual word similarity	38.0	**43.0**	40.9
	multilingual word similarity	58.1	66.6	**69.8**
	word translation	43.7	35.7	**45.7**
	monolingual QVEC	10.3	10.7	**11.9**
	multilingual QVEC	**9.3**	8.7	8.6
	monolingual QVEC-CCA	62.4	**63.4**	46.4
	multilingual QVEC-CCA	**43.3**	41.5	31.0

Table 5: Performance of our model compared with MultiCluster and MultiCCA using extrinsic and intrinsic evaluation tasks on 12 languages proposed in Ammar et al. (2016), all models are trained on the same dataset. The best score for each task is bold.

methods. Therefore, for a strict comparison[8], we train our best model (Joint + Mapping) using the same monolingual data and set of bilingual dictionaries on the same 12 languages with MultiCluster and MultiCCA. Table 5 shows the performance on intrinsic and extrinsic tasks proposed in Ammar et al. (2016). Multilingual dependency parsing and document classification are trained on a set of source languages and test on a target language in the transfer learning setting. Monolingual word similarity task is similar with our monolingual similarity task described in §7, multilingual word similarity is an extension of monolingual word similarity task but tested for pair of words in different languages. Monolingual QVEC, multilingual QVEC test the linguistic content of word embeddings in monolingal and multilingual setting. Monolingual QVEC-CCA and multilingual QVEC-CCA are the extended versions of monolingual QVEC and multilingual QVEC also proposed in Ammar et al. (2016). Table 5 shows that our model achieved competitive results, best at 4 out of 9 evaluation tasks.

[8]also with respect to the word coverage since MultiSkip and MultiTrans usually have much lower word coverage, biasing the intrinsic evaluations.

10 Conclusion

In this paper, we introduced several methods to build unified multilingual word embeddings. This is superior since we can exploit more relations and combine strength from many languages. The input to our model is just a set of monolingual data and a set of bilingual dictionaries between any language pair. We automatically induce the bilingual relationship for all language pairs while keeping high quality monolingual relations. Our multilingual joint training model with explicit mapping consistently achieves better performance compared with linear transformation. We achieve new state-of-the-art performance on bilingual lexicon induction task for recall at 5, similar excellent results with the state-of-the-art bilingual word embeddings on monolingual similarity task (Duong et al., 2016). Moreover, our model is competitive at the crosslingual document classification task, achieving a new state-of-the-art for de→en and

it→de pair.

References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual nlp. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria, August. Association for Computational Linguistics.

Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A. Smith. 2016. Massively multilingual word embeddings. *CoRR*, abs/1602.01925.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Comput. Linguist.*, 19(2):263–311, June.

Sarath Chandar A P, Stanislas Lauly, Hugo Larochelle, Mitesh Khapra, Balaraman Ravindran, Vikas C Raykar, and Amrita Saha. 2014. An autoencoder approach to learning bilingual word representations. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1853–1861. Curran Associates, Inc.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA. ACM.

Jocelyn Coulmance, Jean-Marc Marty, Guillaume Wenzek, and Amine Benhalloum. 2015. Transgram, fast cross-lingual word-embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1109–1113, Lisbon, Portugal, September. Association for Computational Linguistics.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 600–609.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 845–850, Beijing, China. Association for Computational Linguistics.

Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. 2016. Learning Crosslingual Word Embeddings without Bilingual Corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, Austin, Texas, USA, November. Association for Computational Linguistics.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. 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)*, pages 334–343, Beijing, China, July. Association for Computational Linguistics.

Manaal Faruqui and Chris Dyer. 2014. 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, Gothenburg, Sweden, April. Association for Computational Linguistics.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 406–414, New York, NY, USA. ACM.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. Bilbowa: Fast bilingual distributed representations without word alignments. In David Blei and Francis Bach, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 748–756. JMLR Workshop and Conference Proceedings.

Kejun Huang, Matt Gardner, Evangelos Papalexakis, Christos Faloutsos, Nikos Sidiropoulos, Tom Mitchell, Partha P. Talukdar, and Xiao Fu. 2015. Translation invariant word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1084–1088, Lisbon, Portugal, September. Association for Computational Linguistics.

David Kamholz, Jonathan Pool, and Susan Colowick. 2014. Panlex: Building a resource for panlingual lexical translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3145–50, Reykjavik, Iceland. European Language Resources Association (ELRA).

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. In *Proceedings of COLING 2012*, pages 1459–1474, Mumbai, India, December. The COLING 2012 Organizing Committee.

Tomáš Kočiský, Karl Moritz Hermann, and Phil Blunsom. 2014. Learning bilingual word representations

by marginalizing alignments. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 224–229, Baltimore, Maryland, June. Association for Computational Linguistics.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397, December.

Thang Luong, Richard Socher, and Christopher D. Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning, CoNLL 2013, Sofia, Bulgaria, August 8-9, 2013*, pages 104–113.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Bilingual word representations with monolingual quality in mind. In *NAACL Workshop on Vector Space Modeling for NLP*, Denver, United States.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 62–72.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October. Association for Computational Linguistics.

Anders Søgaard, Željko Agić, Héctor Martínez Alonso, Barbara Plank, Bernd Bohnet, and Anders Johannsen. 2015. Inverted indexing for cross-lingual nlp. 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)*, pages 1713–1722, Beijing, China, July. Association for Computational Linguistics.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 477–487. Association for Computational Linguistics.

Ivan Vulić and Marie-Francine Moens. 2015. Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 719–725, Beijing, China, July. Association for Computational Linguistics.

Mengqiu Wang, Wanxiang Che, and Christopher D. Manning. 2013. Joint word alignment and bilingual named entity recognition using dual decomposition. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1082, Sofia, Bulgaria, August. Association for Computational Linguistics.

Min Xiao and Yuhong Guo, 2014. *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, chapter Distributed Word Representation Learning for Cross-Lingual Dependency Parsing, pages 119–129. Association for Computational Linguistics.

David Yarowsky and Grace Ngai. 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, NAACL '01, pages 1–8, Pittsburgh, Pennsylvania.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 30–34, San Diego, California, June. Association for Computational Linguistics.

3.1.7 NAACL 2016

An Attentional Model for Speech Translation Without Transcription

Long Duong,[12] **Antonios Anastasopoulos,**[3] **David Chiang,**[3] **Steven Bird**[14] and **Trevor Cohn**[1]
[1]Department of Computing and Information Systems, University of Melbourne
[2]National ICT Australia, Victoria Research Laboratory
[3]Department of Computer Science and Engineering, University of Notre Dame
[4]International Computer Science Institute, University of California Berkeley

Abstract

For many low-resource languages, spoken language resources are more likely to be annotated with translations than transcriptions. This bilingual speech data can be used for word-spotting, spoken document retrieval, and even for documentation of endangered languages. We experiment with the neural, attentional model applied to this data. On phone-to-word alignment and translation reranking tasks, we achieve large improvements relative to several baselines. On the more challenging speech-to-word alignment task, our model nearly matches GIZA++'s performance on gold transcriptions, but without recourse to transcriptions or to a lexicon.

1 Introduction

For many low-resource languages, spoken language resources are more likely to come with translations than with transcriptions. Most of the world's languages are not written, so there is no orthography for transcription. Phonetic transcription is possible but too costly to produce at scale. Even when a minority language has an official orthography, people are often only literate in the language of formal education, such as the national language. Nevertheless, it is relatively easy to provide written or spoken *translations* for audio sources. Subtitled or dubbed movies are a widespread example.

One application of models of bilingual speech data is documentation of endangered languages. Since most speakers are bilingual in a higher-resource language, they can listen to a source language recording sentence by sentence and provide a spoken translation (Bird, 2010; Bird et al., 2014). By aligning this data at the word level, we hope to automatically identify regions of data where further evidence is needed, leading to a substantial, interpretable record of the language that can be studied even if the language falls out of use (Abney and Bird, 2010; Bird and Chiang, 2012).

We experiment with extensions of the neural, attentional model of Bahdanau et al. (2015), working at the phone level or directly on the speech signal. We assume that the target language is a high-resource language such as English that can be automatically transcribed; therefore, in our experiments, the target side is text rather than the output of an automatic speech recognition (ASR) system.

In the first set of experiments, as a stepping stone to direct modeling of speech, we represent the source as a sequence of phones. For phone-to-word alignment, we obtain improvements of 9–24% absolute F1 over several baselines (Och and Ney, 2000; Neubig et al., 2011; Stahlberg et al., 2012). For phone-to-word translation, we use our model to rerank n-best lists from Moses (Koehn et al., 2007) and observe improvements in BLEU of 0.9–1.7.

In the second set of experiments, we operate directly on the speech signal, represented as a sequence of Perceptual Linear Prediction (PLP) vectors (Hermansky, 1990). Without using transcriptions or a lexicon, the model is able to align the source-language speech to its English translations nearly as well as GIZA++ using gold transcriptions.

Our main contributions are: (i) proposing a new task, alignment of speech with text translations, including a dataset extending the Spanish

Fisher and CALLHOME datasets; (ii) extending the neural, attentional model to outperform existing models at both alignment and translation reranking when working on source-language phones; and (iii) demonstrating the feasibility of alignment directly on source-language speech.

2 Background

To our knowledge, there has been relatively little research on models that operate directly on parallel speech. Typically, speech is transcribed into a word sequence or lattice using ASR, or at least a phone sequence or lattice using a phone recognizer. This normally requires manually transcribed data and a pronunciation lexicon, which can be costly to create. Recent work has introduced models that do not require pronunciation lexicons, but train only on speech with text transcriptions (Lee et al., 2013; Maas et al., 2015; Graves et al., 2006). Here, we bypass phonetic transcriptions completely, and rely only on translations.

Such data can be found, for example, in subtitled or dubbed movies. Some specific examples of corpora of parallel speech are the European Parliament Plenary Sessions Corpus (Van den Heuvel et al., 2006), which includes parliamentary speeches in the 21 official EU languages, as well as their interpretation into all the other languages; and the TED Talks Corpus (Cettolo et al., 2012), which provides speech in one language (usually English) together with translations into other languages.

As mentioned in the introduction, a stepping-stone to model parallel speech is to assume a recognizer that can produce a phonetic transcription of the source language, then to model the transformation from transcription to translation. We compare against three previous models that can operate on sequences of phones. The first is simply to run GIZA++ (IBM Model 4) on a phonetic transcription (without word boundaries) of the source side. Stahlberg et al. (2012) present a modification of IBM Model 3, named Model 3P, designed specifically for phone-to-word alignment. Finally, pialign (Neubig et al., 2011), an unsupervised model for joint phrase alignment and extraction, has been shown to work well at the character level (Neubig et al., 2012) and extends naturally to work on phones.

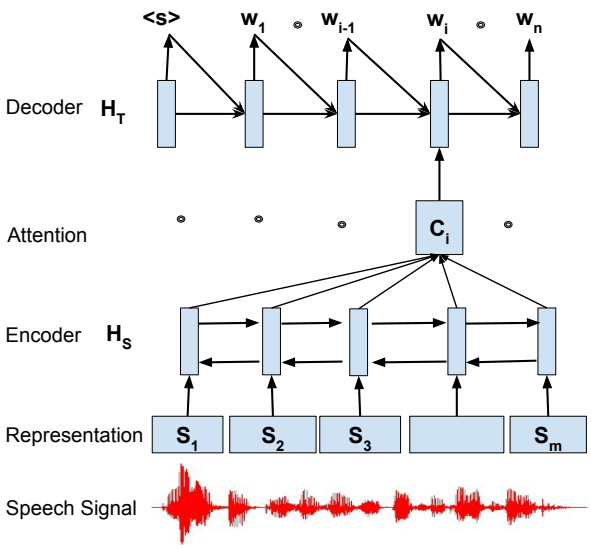


Figure 1: The attentional model as applied to our tasks. We consider two types of input: discrete phone input, or continuous audio, represented as PLP vectors at 10ms intervals

3 Model

We base our approach on the attentional translation model of Cohn et al. (2016), an extension of Bahdanau et al. (2015) which incorporates more fine grained components of the attention mechanism to mimic the structural biases in standard word based translation models. The attentional model encodes a source as a sequence of vectors, then decodes it to generate the output. At each step, it "attends" to different parts of the encoded sequence. This model has been used for translation, image caption generation, and speech recognition (Luong et al., 2015; Xu et al., 2015; Chorowski et al., 2014; Chorowski et al., 2015). Here, we briefly describe the basic attentional model, following Bahdanau et al. (2015), review the extensions for encoding structural biases (Cohn et al., 2016), and then present our novel means for adapting the approach handle parallel speech.

3.1 Base attentional model

The model is shown in Figure 1. The speech signal is represented as a sequence of vectors $S_1, S_2, \ldots, S_m$. For the first set of experiments, each S_i is a 128-dimensional vector-space embedding of a phone. For the second set of experiments, each S_i is the

39-dimensional PLP vector of a single frame of the speech signal. Our model has two main parts: an encoder and a decoder. For the encoder, we used a bidirectional recurrent neural network (RNN) with Long Short-Term Memory (LSTM) units (Hochreiter and Schmidhuber, 1997); we also tried Gated Recurrent Units (Pezeshki, 2015), with similar results. The source speech signal is encoded as sequence of vectors $H_S = (H_S^1, H_S^2, \ldots, H_S^m)$ where each vector H_S^j $(1 \leq j \leq m)$ is the concatenation of the hidden states of the forward and backward LSTMs at time j.

The attention mechanism is added to the model through an alignment matrix $\alpha \in \mathbb{R}^{n \times m}$, where n is the number of target words. We add <s> and </s> to mark the start and end of the target sentence. The row $\alpha_i \in \mathbb{R}^m$ shows where the model should attend to when generating target word w_i. Note that $\sum_{j=1}^m \alpha_{ij} = 1$. The "glimpse" vector c_i of the source when generating w_i is $c_i = \sum_j \alpha_{ij} H_S^j$.

The decoder is another RNN with LSTM units. At each time step, the decoder LSTM receives c_i in addition to the previously-output word. Thus, the hidden state[1] at time i of the decoder is defined as $H_T^i = \text{LSTM}(H_T^{i-1}, c_i, w_{i-1})$, which is used to predict word w_i:

$$p(w_i \mid w_1 \cdots w_{i-1}, H_S) = \text{softmax}(g(H_T^i)), \qquad (1)$$

where g is an affine transformation. We use 128 dimensions for the hidden states and memory cells in both the source and target LSTMs.

We train this model using stochastic gradient descent (SGD) on the negative log-likelihood for 100 epochs. The gradients are rescaled if their L2 norm is greater than 5. We tried Adagrad (Duchi et al., 2011), AdaDelta (Zeiler, 2012), and SGD with momentum (Attoh-Okine, 1999), but found that simple SGD performs best. We implemented dropout (Srivastava et al., 2014) and the local attentional model (Luong et al., 2015), but did not observe any significant improvements.

3.2 Structural bias components

As we are primarily interested in learning accurate alignments (roughly, attention), we include the modelling extensions of Cohn et al. (2016) for incorporating structural biases from word-based translation models into the neural attentional model. As shown later, we observe that including these components result in a substantial improvement in measured alignment quality. We now give a brief overview of these components.

Previous attention. In the basic attentional model, the alignment is calculated based on the source encoding H_S and the previous hidden state H_T^{i-1} of the target, $\alpha_i = \text{Attend}(H_T^{i-1}, H_S)$, where Attend is a function that outputs m attention coefficients. This attention mechanism is overly simplistic, in that it is incapable of capturing patterns in the attention over different positions i. Recognising and exploiting these kinds of patterns has proven critical in traditional word based models of translation (Brown et al., 1993; Vogel et al., 1996; Dyer et al., 2013). For this reason Cohn et al. (2016) include explicit features encoding structural biases from word based models, namely absolute and relative position, Markov conditioning and fertility:

1. previous alignment, α_{i-1}
2. sum of previous alignments, $\sum_{j=1}^{i-1} \alpha_j$
3. source index vector, $(1, 2, 3, \ldots, m)$; and
4. target index vector $(i, i, i, \ldots, i)$.

These features are concatenated to form a feature matrix $\beta \in \mathbb{R}^{4 \times m}$, which are added to the alignment calculation, i.e., $\alpha_i = \text{Attend}(H_T^{i-1}, H_S, \beta)$.

Coverage penalty. The sum over previous alignments feature, described above provides a basic fertility mechanism, however as it operates locally it is only partially effective. To address this, Cohn et al. (2016) propose a global regularisation method for implementing fertility.

Recall that the alignment matrix $\alpha \in \mathbb{R}^{n \times m}$, each α_i is normalized, such that $\sum_j \alpha_{ij} = 1$. However, nothing in the model requires that every source element gets used. This is remedied by encouraging the columns of the alignment matrix to also sum to one, that is, $\sum_i \alpha_{ij} = 1$. To do so, we add a regularization penalty, $\lambda \sum_{j=1}^m \left\| \sum_{i=1}^n \alpha_{ij} - 1 \right\|_2^2$ to the objective function where λ controls the regularization strength. We tune λ on the development set and found that $\lambda = 0.05$ gives the best performance.

[1] The LSTM also carries a memory cell, along with the hidden state; we exclude this from the presentation for clarity of notation.

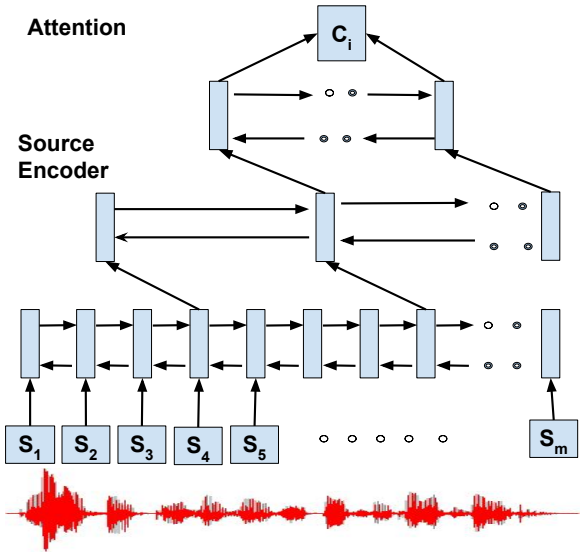


Figure 2: Stacking three layers of LSTM to the source side as in the second set of experiments

4 Extensions for Speech

We can easily apply the attentional model to parallel data, where the source side is represented as a sequence of phones. In cases where no annotated data or lexicon are available, we expect it is difficult to obtain phonetic transcriptions. Instead, we would like to work directly with the speech signal. However, dealing with the speech signal is significantly different than the phone representation, and so we need to modify the base attentional model.

4.1 Stacked and pyramidal RNNs

Both the encoder and decoder can be made more powerful by stacking several layers of LSTMs (Sutskever et al., 2014). For the first set of experiments below, we stack 4 layers of LSTMs on the target side; further layers did not improve performance on the development set.

For the second set of experiments, we work directly with the speech signal as a sequence of PLP vectors, one per frame. Since the frames begin at 10 millisecond intervals, the sequence can be very long. This makes the model slow to train; in our experiments, it seems not to converge at all. Following Chan et al. (2016), we use RNNs stacked into a pyramidal structure to reduce the size of the source speech representation. As illustrated in Figure 2, we stack 3 layers of bidirectional LSTMs. The first layer is the same as the encoder H_S described in Figure 1. The second layer uses every fourth output of the first layer as its input. The third layer selects every other output of the second layer as its input. The attention mechanism is applied only to the top layer. This reduces the size of the alignment matrix by a factor of eight, giving rise to vectors at the top layer representing 80ms intervals, which roughly correspond in duration to input phones.

4.2 Alignment smoothing

In most bitexts, source and target sentences have roughly the same length. However, for our task of aligning text and speech where the speech is represented as a sequence of phones or PLP vectors, the source can easily be several times larger than the target. Therefore we expect that a target word will commonly align to a run of several source elements. We want to encourage this behavior by smoothing the alignment matrix.

The easiest way to do this is by post-processing the alignment matrix. We train the model as usual, and then modify the learned alignment matrix α by averaging each cell over a window, $\alpha'_{ij} := \frac{1}{3}(\alpha_{i,j-1} + \alpha_{ij} + \alpha_{i,j+1})$. The modified alignment matrix, α', is only used for generating hard alignments in our alignment evaluation experiments. We can smooth further by changing the computation of α_{ij} during training. We flatten the softmax by adding a temperature factor, $T \geq 1$:

$$\alpha_{ij} = \frac{\exp(e_{ij}/T)}{\sum_k \exp(e_{ik}/T)}$$

Note that when $T = 1$ we recover the standard softmax function; we set $T = 10$ in both experiments.

5 Experimental Setup

We work on the Spanish CALLHOME Corpus (LDC96S35), which consists of telephone conversations between Spanish native speakers based in the US and their relatives abroad. While Spanish is not a low-resource language, we pretend that it is by not using any Spanish ASR or resources like transcribed speech or pronunciation lexicons (except in the construction of the "silver" standard for evaluation, described below). We also use the English translations produced by Post et al. (2013).

We treat the Spanish speech as a sequence of 39-dimensional PLP vectors (order 12 with energy and first and second order delta) encoding the power spectrum of the speech signal. We do not have gold standard alignments between the Spanish speech and English words for evaluation, so we produced "silver" standard alignments. We used a forced aligner (Gorman et al., 2011) to align the speech to its transcription, and GIZA++ with the `gdfa` symmetrization heuristic (Och and Ney, 2000) to align the Spanish transcription to the English translation. We then combined the two alignments to produce "silver" standard alignments between the Spanish speech and the English words.

Cleaning and splitting the data based on dialogue turns, resulted in a set of 17,532 Spanish utterances from which we selected 250 for development and 500 testing. For each utterance we have the corresponding English translation, and for each word in the translation we have the corresponding span of Spanish speech.

The forced aligner produces the phonetic sequences that correspond to each utterance, which we use later in our first set of experiments as an intermediate representation for the Spanish speech.

In order to evaluate an automatic alignment between the Spanish speech and English translation against the "silver" standard alignment, we compute alignment precision, recall, and F1-score as usual, but on links between Spanish PLP vectors and English words.

6 Phone-to-Word Experiments

In our first set of experiments, we represent the source Spanish speech as a sequence of phones. This sets an upper bound for our later experiments working directly on speech.

6.1 Alignment

We compare our model against three baselines: GIZA++, Model 3P, and pialign. For pialign, in order to better accommodate the different phrase lengths of the two alignment sides, we modified the model to allow different parameters for the Poisson distributions for the average phrase length, as well as different null alignment probabilities for each side.[2] We used the settings `-maxsentlen 200 -maxphraselen 20 -avgphraselenF 10 -nullprobF 0.001`, improving performance by 1.6% compared with the default setting. For Model 3P, we used the settings `-maxFertility 15 -maxWordLength 20`, unrestricted `max[Src/Trg]SenLen` and 10 `Model3Iterations`. We chose the iteration with the highest score to report as the baseline.

Model	F-score	Δ
GIZA++	29.7	−13.0
Model 3P	31.2	−11.5
Pialign (default)	42.4	−0.3
Pialign (modified)	44.0	+1.3
Base model	42.7	+0
+ alignment features	46.2	+3.5
+ coverage penalty	48.6	+5.9
+ stacking	46.3	+3.6
+ alignment smoothing	47.3	+4.6
+ alignment/softmax smoothing	48.2	+5.5
All modifications	53.6	+10.9

Table 1: On the alignment task, the base model performs much better than GIZA++ and Model 3P, and at roughly the same level as pialign; modifications to the model produce further large improvements. The Δ column shows the score difference compared with the base model.

The attentional model produces a soft alignment matrix, whose entries α_{ij} indicate $p(s_j \mid w_i)$ of aligning source phone s_j to target word w_i. For evaluation, we need to convert this to a hard alignment that we can compare against the "silver" standard. Since each word is likely to align with several phones, we choose a simple decoding algorithm: for each phone s_j, pick the word w_i that maximizes $p(w_i \mid s_j)$, where this probability is calculated from alignment matrix α using Bayes' Rule.

Table 1 shows the results of the alignment experiment. The base attentional model achieved an F-score of 42.7%, which is much better than GIZA++ and Model 3P (by 13% and 11.5% absolute, respectively) and at roughly the same level as pialign. Adding our various modifications one at a time

[2] Our modifications have been submitted to the pialign project.

aligner	decoder	reranker none	reranker AM
AM (all mods)		14.6	
GIZA++	Moses	18.2	19.9
pialign	Moses	18.9	19.8
pialign (mod)	Moses	20.2	21.1
Word-based Reference		34.1	

Table 2: BLEU score on the translation task. Using the attentional model (AM) alone (first row) significantly underperformed Moses. However, using the AM as a reranker yielded improvements across several settings. The word-based reference translation provides the upper bound for our phoneme-based systems.

yields improvements ranging from 3.5% to 5.9%. Combining all of them yields a net improvement of 10.9% over the base model, which is 9.4% better than the modified pialign, 22.4% better than Model 3P, and 23.9% better than GIZA++.

6.2 Translation

In this section, we evaluate our model on the translation task. We compare the model against the Moses phrase-based translation system (Koehn et al., 2007), applied to phoneme sequences. We also provide baseline results for Moses applied to word sequences, to serve as an upper bound. Since Moses requires word alignments as input, we used various alignment models: GIZA++, pialign, and pialign with our modifications. Table 2 shows that translation performance roughly correlates with alignment quality.

For the attentional model, we used all of the modifications described above except alignment smoothing. We also used more dimensions (256) for hidden states and memory cells in both encoder and decoder. The decoding algorithm starts with the symbol <s> and uses beam search to generate the next word. The generation process stops when we reach the symbol </s>. We use a beam size of 5, as larger beam sizes make the decoder slower without substantial performance benefits.

As shown in Table 2, the attentional model achieved a BLEU score of 14.6 on the test data, whereas the Moses baselines achieve much better BLEU scores, from 18.2 to 20.2. We think this is because the attentional model is powerful, but we don't have enough data to train it fully given that the output space is the size of the vocabulary. Moreover, this attentional model has been configured to optimize the alignment quality rather than translation quality.

We then tried using the attentional model to rerank 100-best lists output by Moses. The model gives a score for generating the next word $p(w_i|w_1 \cdots w_{i-1}, H_S)$ as in equation (1). We simply compute the score of a hypothesis by averaging the negative log probabilities of the output words,

$$\text{score}(w_1 \cdots w_n) = -\frac{1}{n}\sum_{i=1}^{n} \log(p(w_i|w_1 \cdots w_{i-1}, H_S)),$$

and then choosing the best scoring hypothesis. Table 2 shows the result using the attentional model as the reranker on top of Moses, giving improvements of 0.9 to 1.7 BLEU over their corresponding baselines. These consistent improvements suggest that the probability estimation part of the attentional model is good, but perhaps the search is not adequate. Further research is needed to improve the attentional model's translation quality. Another possibility, which we leave for future work, is to include the attentional model score as a feature in Moses.

Table 3 shows some example translations comparing different models. In all examples, it appears that using pialign produced better translations than GIZA++. Using the attentional model as a reranker for pialign further corrects some errors. Using the attentional model alone seems to perform the worst, which is evident in the third example where the attentional model simply repeats a text fragment (although all models do poorly here). Despite the often incoherent output, the attentional model still captures the main keywords used in the translation.

We test this hypothesis by applying the attentional model for a cross-lingual keyword spotting task where the input is the English keyword and the outputs are all Spanish sentences (represented as phones) containing a likely translation of the keyword. From the training data we select the top 200 terms as the keyword based on tf.idf. The relevance judgment is based on exact word matching. The attentional model achieved 35.8% precision, 43.3%

recall and 36.0% F-score on average on 200 queries. Table 4 shows the English translations of retrieved Spanish sentences. In the first example, the attentional model identifies *mañana* as the translation of *tomorrow*. In the second example, it does reasonably well by retrieving 2 correct sentences out of 3, correctly identifying *dejamos* and *salgo* as the translation of *leave*.

7 Speech-to-Word Experiments

In this section, we represent the source Spanish speech as a sequence of 39 dimensional PLP vectors. The frame length is 25ms, and overlapping frames are computed every 10ms. As mentioned in Section 4.1, we used a pyramidal RNN to reduce the speech representation size. Other than that, the model used here is identical to the first set of experiments.

Using this model directly for translation from speech does not yield useful output, as is to be expected from the small training data, noisy speech data, and an out-of-domain language model. However, we are able to produce useful results for the ASR and alignment tasks, as presented below.

	PER (%)
Our model	24.3
Our model + monotonic	22.3
Chorowski et al. (2014)	18.6
Graves et al. (2013)	17.7

Table 5: Phone-error-rate (PER) for various models evaluated on TIMIT

7.1 ASR Evaluation

To illustrate the utility of our approach to modelling speech input, first, we evaluate on the more common ASR task of phone recognition. This can be considered as a sub-problem of translation, and moreover, this allows us to benchmark our approach against the state-of-the-art in phone recognition. We experimented on the TIMIT dataset. Following convention, we removed all the SA sentences, evaluated on the 24 speaker core test set and used the 50 auxiliary speaker development set for early stopping. The model was trained to recognize 48 phonemes and was mapped to 39 phonemes for testing. We extracted 39 dimensional PLP features from the TIMIT dataset and trained the same model without any modification. Table 5 shows the performance of our model. It performs reasonably well compared with the state-of-the-art (Graves et al., 2013), considering that we didn't tune any hyper-parameters or feature representations for the task. Moreover, our model is not designed for the monotonic constraints inherent to the ASR problem, which process the input without reordering. By simply adding a masking function (equation 2 from Chorowski et al. (2014)) to encourage the monotonic constraint in the alignment function, we observe a 2% PER improvement. This is close to the performance reported by Chorowski et al. (2014) (Table 5), despite the fact that they employed user-adapted speech features.

7.2 Alignment Evaluation

We use alignment as a second evaluation, training and testing on parallel data comprising paired Spanish speech input with its English translations (as described in §5), and using the speech-based modelling techniques (see §4.) We compare to a naive baseline where we assume that each English letter (not including spaces) corresponds to an equal number of Spanish frames. The results of our attentional model and the baseline are summarized in Table 6. The attentional model is substantially lower than the scores in Table 1, because the PLP vector representation is much less informative than the gold phonetic transcription. Here, we have to identify phones and their boundaries in addition to phone-word alignment. However, the naive baseline does surprisingly well, presumably because our (unrealistic) choice of Spanish-English does not have very much reordering.

Figure 3 presents some examples of Spanish speech and English text, showing a heat map of the alignment matrix α (before smoothing). Due to the pyramidal structure of the encoder, each column roughly corresponds to 80ms. In the example on the left, the model is confident at aligning *a little* with columns 1–5, which corresponds roughly to their correct Spanish translation *algo*. We misalign the word *of* with columns 8–10, when the correct alignment should be columns 5–6, corresponding

Phones	sil e m e d i h o k e t e i B a a y a m a r sp a y e r o a n t e a y e r sp sil
Transcription	eh , me dijo que te iba a llamar , ayer , o anteayer
AM	eh , he told me that she was going to call , yesterday before yesterday
Giza	oh , he told me that you called yesterday or before yesterday .
Mod. Pialign	eh , she told me that I was going to call yesterday or before yesterday .
Mod. Pialign + AM	eh , he told me that I was going to call , yesterday or before yesterday .
Reference	eh , he told me that he was going to call you , yesterday , or the day before yesterday .
Phones	sil i t u k o m o a s e s t a D o h w a n i t o e s t a s t r a B a h a n d o k e sp e s t a s a s y e n d o sil
Transcription	y tú , cómo has estado , juanito , estás trabajando , qué estás haciendo.
AM	and how have you been working , are working ?
GIZA	and how are you Juanito , are you job , what are you doing ?
Mod. pialign	and how have you been Juanito are you working , what are you doing ?
Mod. pialign + AM	and how have you been Juan , are you working , what are you doing ?
Reference	and how have you been , Juanita , are you working , what are you doing .
Phones	sil t e n g o k e a s e r l e e l a s e o a s i k o m o a u n h a r D i n i n f a n t i l sp sil
Transcription	tengo que hacerle el aseo así como a un jardín infantil –
AM	I have to have to him like to like that to ⟨unkA⟩
GIZA	I have to do the , the how a vegetable information in the .
Mod. pialign	I have to do the that like to a and it was , didn't you don't have the .
Mod. pialign + AM	I have to make the or like to a and it was , didn't you don't have the –
Reference	I have to clean it like a kindergarten

Table 3: Translation examples for various models: the attentional model (AM), the standard Moses with GIZA++ aligner (giza), with modified Pialign aligner (Mod. pialign) and using the attentional model as reranker on top of pialign.

Keyword : **tomorrow**
El va **mañana** para Caracas. A qué va a Caracas él.
Y **mañana** , y **mañana** o pasado te voy a poner un paquete.
Oh , no , Julio no sé a dónde está y va **mañana** a Caracas , está con Richard.
Oye , qué bueno , entonces nos vamos tempranito en la **mañana**
No , aquí la gente se acuesta a las dos de la **mañana**.
Keyword : **leave**
Todo , organizar completo todo , desde los alquileres , la comida , mozo , cantina , todo lo pongo yo aquí
Y entonces dónde lo **dejamos** pagando estacionamiento y pagando seguro
Sí , el veintiuno. yo **salgo** de para aquí el dieciséis para florida , y el veintiuno llego a Caracas.

Table 4: Examples of cross-lingual keyword spotting using the attentional model. The bolded terms in the retrieved text are based on manual inspection.

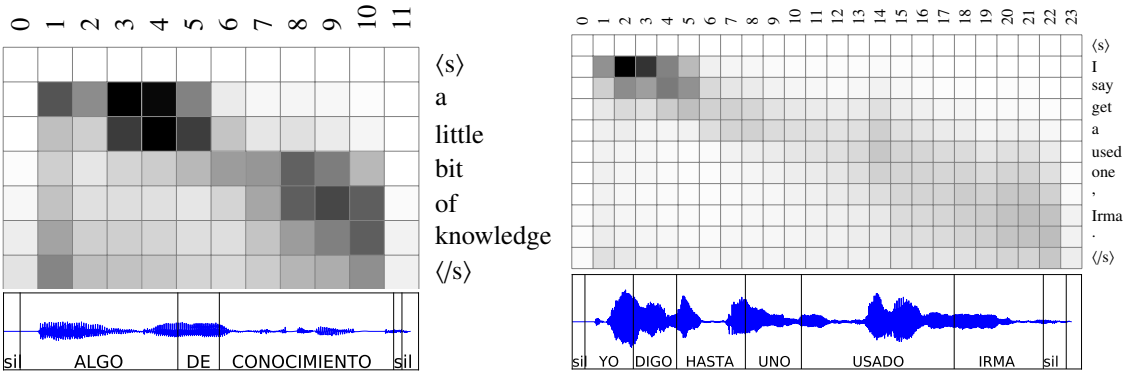


Figure 3: PLP-word alignment examples. The heat maps shows the alignment matrix which is time-aligned with the speech signals and their transcriptions.

ASR	aligner	F1
none	Naive baseline	31.7
none	AM (all mods)	26.4
cz	AM (all mods)	28.0
hu	AM (all mods)	27.9
ru	AM (all mods)	27.4
es	GIZA++	29.7

Table 6: Alignment of Spanish speech to English translations. In the first two rows, no gold or automatic transcriptions of any sort are used. In the next three rows, non-Spanish phone recognizers (cz, hu, ru) are used on the Spanish speech and the attentional model is run on the noisy transcription; this does better than no transcriptions. The last row is an unfair comparison because it uses gold Spanish (es) phonetic transcriptions; nevertheless, our model performs nearly as well.

to Spanish translation *de*. The word *knowledge* is aligned quite well with columns 7–10, corresponding to Spanish *conocimiento*. The example on the right is for a longer sentence. The model is less confident about this example, mostly because there are words that appear infrequently, such as the personal name *Irma*. However, we are still observing diagonal-like alignments that are roughly correct. In both examples, the model correctly leaves silence (`sil`) unaligned.

As a middle ground between assuming gold phonetic transcriptions (cf. Section 6) and no transcriptions at all, we use noisy transcriptions by running speech recognizers for other languages on the Spanish speech: Russian (ru), Hungarian (hu) and Czech (cz) (Vasquez et al., 2012). These distantly related languages were chosen to be a better approximation to the low-resource scenario. All three models perform better than operating directly on the speech signal (Table 6), and notably, the Russian result is nearly as good as GIZA++'s performance on gold phonetic transcriptions.

8 Conclusion

This paper reports our work to train models directly on parallel speech, i.e. source-language speech with English text translations that, in the low-resource setting, would have originated from spoken translations. To our knowledge, it is the first exploration of this type. We augmented the Spanish Fisher and CALLHOME datasets and extended the alignment F1 evaluation metric for this setting. We extended the attentional model of Bahdanau et al. to work on parallel speech and observed improvements relative to all baselines on phone-to-word alignment. On speech-to-word alignment, our model, without using any knowledge of Spanish, performs almost as well as GIZA++ using gold Spanish transcriptions.

Language pairs with word-order divergences and other divergences will of course be more challenging than Spanish-English. This work provides a proof-of-concept that we hope will spur future work towards solving this important problem in a true low-resource language.

Acknowledgments

This work was partly conducted during Duong's internship at ICSI, UC Berkeley and partially supported by the University of Melbourne and National ICT Australia (NICTA). We are grateful for support from NSF Award 1464553 and the DARPA LORELEI Program. Cohn is the recipient of an Australian Research Council Future Fellowship FT130101105.

References

Steven Abney and Steven Bird. 2010. The Human Language Project: Building a universal corpus of the world's languages. In *Proceedings of ACL*, pages 88–97.

Nii O. Attoh-Okine. 1999. Analysis of learning rate and momentum term in backpropagation neural network algorithm trained to predict pavement performance. *Advances in Engineering Software*, 30(4):291–302.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*.

Steven Bird and David Chiang. 2012. Machine translation for language preservation. In *Proceedings of COLING*, pages 125–134, Mumbai, India.

Steven Bird, Lauren Gawne, Katie Gelbart, and Isaac McAlister. 2014. Collecting bilingual audio in remote indigenous communities. In *Proceedings of COLING*, pages 1015–1024.

Steven Bird. 2010. A scalable method for preserving oral literature from small languages. In *The Role of Digital Libraries in a Time of Global Change: 12th Inter-*

national Conference on Asia-Pacific Digital Libraries, pages 5–14, Berlin, Heidelberg. Springer-Verlag.

Peter E. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2).

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT^3: Web inventory of transcribed and translated talks. In *Proceedings of EAMT*, pages 261–268.

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Proceedings of ICASSP*.

Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. End-to-end continuous speech recognition using attention-based recurrent NN: first results. In *Proceedings of NIPS Workshop on Deep Learning and Representation Learning*.

Jan K. Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Proceedings of NIPS*, pages 577–585.

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. In *Proceedings of NAACL HLT*.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM Model 2. In *Proceedings of NAACL HLT*, pages 644–648.

Kyle Gorman, Jonathan Howell, and Michael Wagner. 2011. Prosodylab-Aligner: A tool for forced alignment of laboratory speech. *Canadian Acoustics*, 39(3):192–193.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of ICML*, pages 369–376. ACM.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Proceedings of ICASSP*, pages 6645–6649.

Hynek Hermansky. 1990. Perceptual linear predictive (PLP) analysis for speech. *Acoustical Society of America*, pages 1738–1752.

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

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Christ Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL (Interactive Poster and Demonstration Sessions)*, pages 177–180.

Chia-ying Lee, Yu Zhang, and James Glass. 2013. Joint learning of phonetic units and word pronunciations for ASR. In *Proceedings of EMNLP*, pages 182–192.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of EMNLP*, pages 1412–1421.

Andrew L. Maas, Ziang Xie, Dan Jurafsky, and Andrew Y. Ng. 2015. Lexicon-free conversational speech recognition with neural networks. In *Proceedings of NAACL HLT*, pages 345–354.

Graham Neubig, Taro Watanabe, Eiichiro Sumita, Shinsuke Mori, and Tatsuya Kawahara. 2011. An unsupervised model for joint phrase alignment and extraction. In *Proceedings of NAACL HLT*, pages 632–641.

Graham Neubig, Taro Watanabe, Shinsuke Mori, and Tatsuya Kawahara. 2012. Machine translation without words through substring alignment. In *Proceedings of ACL*, pages 165–174.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of ACL*, pages 440–447.

Mohammad Pezeshki. 2015. Sequence modeling using gated recurrent neural networks. *arXiv preprint arXiv:1501.00299*.

Matt Post, Gaurav Kumar, Adam Lopez, Damianos Karakos, Chris Callison-Burch, and Sanjeev Khudanpur. 2013. Improved speech-to-text translation with the Fisher and Callhome Spanish–English speech translation corpus. In *Proceedings of IWSLT*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Felix Stahlberg, Tim Schlippe, Sue Vogel, and Tanja Schultz. 2012. Word segmentation through cross-lingual word-to-phoneme alignment. In *Proceedings of the IEEE Spoken Language Technology Workshop (SLT)*, pages 85–90.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of NIPS*, pages 3104–3112.

Henk Van den Heuvel, Khalid Choukri, Chr Gollan, Asuncion Moreno, and Djamel Mostefa. 2006. Tc-

star: New language resources for ASR and SLT purposes. In *Proceedings of LREC*, pages 2570–2573.

Daniel Vasquez, Rainer Gruhn, and Wolfgang Minker. 2012. *Hierarchical Neural Network Structures for Phoneme Recognition*. Springer.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proceedings of COLING*, pages 836–841.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of ICML*, pages 2048–2057.

Matthew D. Zeiler. 2012. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

3.2 Evaluation of Contribution

Have the task and paper ...

3.2.1 Research question revisited

3.3 Future Work

Real case for low-resource language (cost for annotation, cost for adapting to different language...)

3.4 Conclusion

Bibliography

ADAMS, OLIVER, GRAHAM NEUBIG, TREVOR COHN, STEVEN BIRD, QUOC TRUONG DO, and SATOSHI NAKAMURA. 2016. Learning a lexicon and translation model from phoneme lattices. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2377–2382, Austin, Texas. Association for Computational Linguistics.

AGARWAL, APOORV, BOYI XIE, ILIA VOVSHA, OWEN RAMBOW, and REBECCA PASSONNEAU. 2011. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, 30–38, Stroudsburg, PA, USA. Association for Computational Linguistics.

AMMAR, WALEED, GEORGE MULCAIRE, YULIA TSVETKOV, GUILLAUME LAMPLE, CHRIS DYER, and NOAH A. SMITH. 2016. Massively multilingual word embeddings. *CoRR* abs/1602.01925.

ANDREAS, JACOB, and DAN KLEIN. 2014. How much do word embeddings encode about syntax? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 822–827.

BAHDANAU, DZMITRY, KYUNGHYUN CHO, and YOSHUA BENGIO. 2014. Neural machine translation by jointly learning to align and translate. *CoRR* abs/1409.0473.

BANKO, MICHELE, and ROBERT C. MOORE. 2004. Part of speech tagging in context. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04.

BANSAL, MOHIT, KEVIN GIMPEL, and KAREN LIVESCU. 2014. Tailoring continuous word representations for dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 809–815.

BARONI, MARCO, GEORGIANA DINU, and GERMÁN KRUSZEWSKI. 2014. 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)*, 238–247, Baltimore, Maryland. Association for Computational Linguistics.

BAUMANN, PETER, and JANET PIERREHUMBERT. 2014. Using resource-rich languages to improve morphological analysis of under-resourced languages. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 3355–3359, Reykjavik, Iceland.

BENGIO, YOSHUA, RÉJEAN DUCHARME, PASCAL VINCENT, and CHRISTIAN JANVIN. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.* 3.1137–1155.

BERG-KIRKPATRICK, TAYLOR, ALEXANDRE BOUCHARD-CÔTÉ, JOHN DENERO, and DAN KLEIN. 2010. Painless unsupervised learning with features. In *Proceeding of Human Language Technology - North Americal Association for Computational Linguistics*, 582–590.

BERMENT, VINCENT, 2004. *Methods to computerize "little equipped" languages and groups of languages*. Université Joseph-Fourier - Grenoble I dissertation.

BERNDT, DONALD J., and JAMES CLIFFORD. 1994. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, AAAIWS'94, 359–370. AAAI Press.

BESACIER, LAURENT, ETIENNE BARNARD, ALEXEY KARPOV, and TANJA SCHULTZ. 2014. Automatic speech recognition for under-resourced languages: A survey. *Speech Commun.* 56.85–100.

BIEMANN, CHRIS. 2006a. Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, TextGraphs-1, 73–80. Association for Computational Linguistics.

——. 2006b. Unsupervised part-of-speech tagging employing efficient graph clustering. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, COLING ACL '06, 7–12. Association for Computational Linguistics.

BIRD, STEVEN, LAUREN GAWNE, KATIE GELBART, and ISAAC MCALISTER. 2014a. Collecting bilingual audio in remote indigenous communities. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, 1015–1024, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

——, FLORIAN R. HANKE, OLIVER ADAMS, and HAEJOONG LEE. 2014b. Aikuma: A mobile app for collaborative language documentation. In *Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages*, 1–5.

BLACHON, DAVID, ELODIE GAUTHIER, LAURENT BESACIER, GUY-NOL KOUARATA, MARTINE ADDA-DECKER, and ANNIE RIALLAND. 2016. Parallel speech collection for under-resourced language studies using the lig-aikuma mobile device app. *Procedia Computer Science* 81.61 – 66. SLTU-2016 5th Workshop on Spoken Language Technologies for Under-resourced languages 09-12 May 2016 Yogyakarta, Indonesia.

BLACOE, WILLIAM, and MIRELLA LAPATA. 2012. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 546–556, Jeju Island, Korea. Association for Computational Linguistics.

BLUNSOM, PHIL, and TREVOR COHN. 2011. A hierarchical pitman-yor process hmm for unsupervised part of speech induction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, 865–874.

BOCCHIERI, E., and G. DODDINGTON. 1986. Speaker independent digit recognition with reference frame-specific distance measures. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86.*, volume 11, 2699–2702.

BÖHMOVÁ, ALENA, JAN HAJIČ, EVA HAJIČOVÁ, and BARBORA HLADKÁ. 2001. The Prague Dependency Treebank: A Three-Level Annotation Scenario. In *Treebanks: Building and Using Syntactically Annotated Corpora*, ed. by Anne Abeillé, 103–127. Kluwer Academic Publishers.

BRANTS, THORSTEN. 2000. TnT: A statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing (ANLP '00)*, 224–231, Seattle, Washington, USA.

BRILL, ERIC. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics* 21.543–565.

BROWN, PETER F., PETER V. DESOUZA, ROBERT L. MERCER, VINCENT J. DELLA PIETRA, and JENIFER C. LAI. 1992. Class-based n-gram models of natural language. *Computational Linguistics* 18.467–479.

BRUNI, ELIA, GEMMA BOLEDA, MARCO BARONI, and NAM-KHANH TRAN. 2012. Distributional semantics in technicolor. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers*, 136–145.

CAMACHO-COLLADOS, JOSÉ, MOHAMMAD TAHER PILEHVAR, and ROBERTO NAVIGLI. 2015. A framework for the construction of monolingual and cross-lingual word similarity datasets. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 1–7, Beijing, China. Association for Computational Linguistics.

CHAN, WILLIAM, NAVDEEP JAITLY, QUOC V. LE, and ORIOL VINYALS. 2015. Listen, attend and spell. *CoRR* abs/1508.01211.

CHANDAR A P, SARATH, STANISLAS LAULY, HUGO LAROCHELLE, MITESH KHAPRA, BALARAMAN RAVINDRAN, VIKAS C RAYKAR, and AMRITA SAHA. 2014. An autoencoder approach

to learning bilingual word representations. In *Advances in Neural Information Processing Systems 27*, ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, 1853–1861. Curran Associates, Inc.

CHEN, DANQI, and CHRISTOPHER MANNING. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 740–750, Doha, Qatar. Association for Computational Linguistics.

CHEN, WENLIANG, YUE ZHANG, and MIN ZHANG. 2014a. Feature embedding for dependency parsing. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, 816–826, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

CHEN, XINXIONG, ZHIYUAN LIU, and MAOSONG SUN. 2014b. A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1025–1035, Doha, Qatar. Association for Computational Linguistics.

CHOROWSKI, JAN, DZMITRY BAHDANAU, KYUNGHYUN CHO, and YOSHUA BENGIO. 2014. End-to-end continuous speech recognition using attention-based recurrent NN: first results. *CoRR* abs/1412.1602.

CHRISTODOULOPOULOS, CHRISTOS, SHARON GOLDWATER, and MARK STEEDMAN. 2010. Two decades of unsupervised POS induction: How far have we come? In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, 575–584.

CHU, Y. J., and T. H. LIU. 1965. On the shortest arborescence of a directed graph.

COLLOBERT, RONAN, and JASON WESTON. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, 160–167, New York, NY, USA. ACM.

COULMANCE, JOCELYN, JEAN-MARC MARTY, GUILLAUME WENZEK, and AMINE BENHALLOUM. 2015. Trans-gram, fast cross-lingual word-embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1109–1113, Lisbon, Portugal. Association for Computational Linguistics.

CRYSTAL, D. 2002. *Language Death*. Canto Refresh Your Series. Cambridge University Press.

CUCU, H., L. BESACIER, C. BURILEANU, and A. BUZO. 2012. Asr domain adaptation methods for low-resourced languages: Application to romanian language. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, 1648–1652.

DARWISH, KAREEM. 2013. Named entity recognition using cross-lingual resources: Arabic as an example. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1558–1567, Sofia, Bulgaria. Association for Computational Linguistics.

DAS, DIPANJAN, and SLAV PETROV. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, 600–609.

DE MARNEFFE, MARIE-CATHERINE, TIMOTHY DOZAT, NATALIA SILVEIRA, KATRI HAVERINEN, FILIP GINTER, JOAKIM NIVRE, and CHRISTOPHER D. MANNING. 2014. Universal stanford dependencies: A cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014.*, 4585–4592.

DE MARNEFFE, MARIE-CATHERINE, BILL MACCARTNEY, and CHRISTOPHER D. MANNING. 2006. Generating typed dependency parses from phrase structure parses. In *IN PROC. International Conference on Language Resources and Evaluation (LREC*, 449–454.

——, and CHRISTOPHER D. MANNING. 2008. The stanford typed dependencies representation. In *Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, CrossParser '08, 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

DRYER, MATTHEW S., and MARTIN HASPELMATH (eds.) 2013. *WALS Online*. Leipzig: Max Planck Institute for Evolutionary Anthropology.

DUONG, LONG, PAUL COOK, STEVEN BIRD, and PAVEL PECINA. 2013a. Increasing the quality and quantity of source language data for Unsupervised Cross-Lingual POS tagging. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 1243–1249, Nagoya, Japan.

——, ——, ——, and ——. 2013b. Simpler unsupervised POS tagging with bilingual projections. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 634–639, Sofia, Bulgaria.

DYER, CHRIS, MIGUEL BALLESTEROS, WANG LING, AUSTIN MATTHEWS, and NOAH A. SMITH. 2015. Transition-based dependency parsing with stack long short-term memory. 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)*, 334–343, Beijing, China. Association for Computational Linguistics.

EISNER, JASON, and JOHN BLATZ. 2007. Program transformations for optimization of parsing algorithms and other weighted logic programs. In *Proceedings of FG 2006: The 11th Conference on Formal Grammar*, ed. by Shuly Wintner, 45–85. CSLI Publications.

FANG, ZHENG, ZHANG GUOLIANG, and SONG ZHANJIANG. 2001. Comparison of different implementations of mfcc. *J. Comput. Sci. Technol.* 16.582–589.

FARUQUI, MANAAL, and CHRIS DYER. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 462–471, Gothenburg, Sweden. Association for Computational Linguistics.

FELDMAN, ANNA, JIRKA HANA, and CHRIS BREW. 2006. A cross-language approach to rapid creation of new morpho-syntactically annotated resources. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'06)*, 549–554, Genoa, Italy.

FINKELSTEIN, LEV, EVGENIY GABRILOVICH, YOSSI MATIAS, EHUD RIVLIN, ZACH SOLAN, GADI WOLFMAN, and EYTAN RUPPIN. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, 406–414, New York, NY, USA. ACM.

GANCHEV, KUZMAN, JENNIFER GILLENWATER, and BEN TASKAR. 2009. Dependency grammar induction via bitext projection constraints. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, 369–377, Stroudsburg, PA, USA. Association for Computational Linguistics.

GAROFOLO, J. S., L. F. LAMEL, W. M. FISHER, J. G. FISCUS, D. S. PALLETT, and N. L. DAHLGREN, 1993. DARPA TIMIT acoustic phonetic continuous speech corpus CDROM.

GARRETTE, DAN, and JASON BALDRIDGE. 2013. Learning a part-of-speech tagger from two hours of annotation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-13)*, 138–147, Atlanta, GA.

——, JASON MIELENS, and JASON BALDRIDGE. 2013. Real-world semi-supervised learning of pos-taggers for low-resource languages. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL-2013)*, 583–592, Sofia, Bulgaria.

GELLING, DOUWE, TREVOR COHN, PHIL BLUNSOM, and JOÃO GRAÇA. 2012. The pascal challenge on grammar induction. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, WILS '12, 64–80, Stroudsburg, PA, USA. Association for Computational Linguistics.

GEORGI, RYAN, FEI XIA, and WILLIAM D. LEWIS. 2013. Enhanced and portable dependency projection algorithms using interlinear glossed text. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 306–311, Sofia, Bulgaria. Association for Computational Linguistics.

GHITZA, ODED. 2011. Linking speech perception and neurophysiology: speech decoding guided by cascaded oscillators locked to the input rhythm. *Frontiers in psychology* 2.130.

GIMPEL, KEVIN, NATHAN SCHNEIDER, BRENDAN O'CONNOR, DIPANJAN DAS, DANIEL MILLS, JACOB EISENSTEIN, MICHAEL HEILMAN, DANI YOGATAMA, JEFFREY FLANIGAN, and NOAH A. SMITH. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 42–47, Portland, Oregon, USA. Association for Computational Linguistics.

GOLDBERG, YOAV, MENI ADLER, and MICHAEL ELHADAD. 2008. Em can find pretty good hmm pos-taggers (when given a good start. In *In Proc. ACL*, 746–754.

——, and JOAKIM NIVRE. 2012. A dynamic oracle for arc-eager dependency parsing. In *Proceedings of COLING 2012*, 959–976. The COLING 2012 Organizing Committee.

GOLDHAHN, DIRK, THOMAS ECKART, and UWE QUASTHOFF. 2012. Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, ed. by Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, 759–765, Istanbul, Turkey. European Language Resources Association (ELRA). ACL Anthology Identifier: L12-1154.

GOUWS, STEPHAN, YOSHUA BENGIO, and GREG CORRADO. 2015. Bilbowa: Fast bilingual distributed representations without word alignments. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, ed. by David Blei and Francis Bach, 748–756. JMLR Workshop and Conference Proceedings.

——, and ANDERS SØGAARD. 2015. Simple task-specific bilingual word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1386–1390, Denver, Colorado. Association for Computational Linguistics.

GRAVES, ALEX, SANTIAGO FERNNDEZ, and FAUSTINO GOMEZ. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceeding of International Conference in Machine Learning (ICML)*, 369–376.

——, ABDEL-RAHMAN MOHAMED, and GEOFFREY E. HINTON. 2013. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, 6645–6649.

GREZL, F., M. KARAFIAT, S. KONTAR, and J. CERNOCKY. 2007. Probabilistic and bottle-neck features for lvcsr of meetings. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, volume 4, IV–757–IV–760.

HAN, BO, and TIMOTHY BALDWIN. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 368–378, Portland, Oregon, USA. Association for Computational Linguistics.

HANA, JIRI, ANNA FELDMAN, and CHRIS BREW. 2004. A resource-light approach to Russian morphology: Tagging Russian using Czech resources. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP '04)*, 222–229, Barcelona, Spain.

HERMANN, KARL MORITZ, and PHIL BLUNSOM. 2014. Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 58–68, Baltimore, Maryland. Association for Computational Linguistics.

HERMANSKY, H., D. P. W. ELLIS, and S. SHARMA. 2000. Tandem connectionist feature extraction for conventional hmm systems. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on*, volume 3, 1635–1638 vol.3.

HERMANSKY, HYNEK. 1990. Perceptual linear predictive (plp) analysis for speech. *Acoustical Society of America* 1738–1752.

HILL, FELIX, ROI REICHART, and ANNA KORHONEN. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics* 41.665–695.

HUANG, ERIC H., RICHARD SOCHER, CHRISTOPHER D. MANNING, and ANDREW Y. NG. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, 873–882, Stroudsburg, PA, USA. Association for Computational Linguistics.

HUANG, KEJUN, MATT GARDNER, EVANGELOS PAPALEXAKIS, CHRISTOS FALOUTSOS, NIKOS SIDIROPOULOS, TOM MITCHELL, PARTHA P. TALUKDAR, and XIAO FU. 2015. Translation invariant word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1084–1088, Lisbon, Portugal. Association for Computational Linguistics.

HWA, REBECCA, PHILIP RESNIK, AMY WEINBERG, CLARA CABEZAS, and OKAN KOLAK. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering* 11.311–325.

IACOBACCI, IGNACIO, MOHAMMAD TAHER PILEHVAR, and ROBERTO NAVIGLI. 2015. Sensembed: Learning sense embeddings for word and relational similarity. 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), 95–105, Beijing, China. Association for Computational Linguistics.

JANSEN, A., and B. VAN DURME. 2011. Efficient spoken term discovery using randomized algorithms. In *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, 401–406.

JOHNSON, MARK. 2007. Transforming projective bilexical dependency grammars into efficiently-parsable cfgs with unfold-fold. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 168–175. Association for Computational Linguistics.

——, THOMAS L. GRIFFITHS, and SHARON GOLDWATER. 2006. Adaptor grammars: A framework for specifying compositional nonparametric bayesian models. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, 641–648.

KALCHBRENNER, NAL, EDWARD GREFENSTETTE, and PHIL BLUNSOM. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 655–665, Baltimore, Maryland. Association for Computational Linguistics.

KAMHOLZ, DAVID, JONATHAN POOL, and SUSAN COLOWICK. 2014. Panlex: Building a resource for panlingual lexical translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 3145–50, Reykjavik, Iceland. European Language Resources Association (ELRA).

KAMPER, H., M. ELSNER, A. JANSEN, and S. GOLDWATER. 2015a. Unsupervised neural network based feature extraction using weak top-down constraints. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5818–5822.

KAMPER, HERMAN, AREN JANSEN, and SHARON GOLDWATER. 2015b. *Fully Unsupervised Small-Vocabulary Speech Recognition Using a Segmental Bayesian Model*.

——, ——, and ——. 2015c. Fully unsupervised small-vocabulary speech recognition using a segmental bayesian model. In *INTERSPEECH*.

——, ——, and ——. 2016a. A segmental framework for fully-unsupervised large-vocabulary speech recognition. *CoRR* abs/1606.06950.

——, ——, and ——. 2016b. Unsupervised word segmentation and lexicon discovery using acoustic word embeddings. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.* 24.669–679.

KARANASOU, PANAGIOTA, and LORI LAMEL. 2010. Comparing smt methods for automatic generation of pronunciation variants. In *Proceedings of the 7th International Conference on Advances in Natural Language Processing*, IceTAL'10, 167–178, Berlin, Heidelberg. Springer-Verlag.

KHANAGHA, VAHID, KHALID DAOUDI, ORIOL PONT, and HUSSEIN YAHIA. 2014. Phonetic segmentation of speech signal using local singularity analysis. *Digit. Signal Process.* 35.86–94.

KIROS, RYAN, YUKUN ZHU, RUSLAN SALAKHUTDINOV, RICHARD S. ZEMEL, ANTONIO TORRALBA, RAQUEL URTASUN, and SANJA FIDLER. 2015. Skip-thought vectors. *CoRR* abs/1506.06726.

KLEMENTIEV, ALEXANDRE, IVAN TITOV, and BINOD BHATTARAI. 2012. Inducing crosslingual distributed representations of words. In *Proceedings of COLING 2012*, 1459–1474, Mumbai, India. The COLING 2012 Organizing Committee.

KOEHN, PHILIPP. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*, 79–86, Phuket, Thailand.

KONG, LINGPENG, NATHAN SCHNEIDER, SWABHA SWAYAMDIPTA, ARCHNA BHATIA, CHRIS DYER, and NOAH A. SMITH. 2014. A dependency parser for tweets. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1001–1012, Doha, Qatar. Association for Computational Linguistics.

KUPIEC, JULIAN. 1992. Robust part-of-speech tagging using a hidden markov model. *Computer Speech & Language* 6.225 – 242.

LE, QUOC V., and TOMAS MIKOLOV. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, 1188–1196.

LE, V. B., and L. BESACIER. 2009. Automatic speech recognition for under-resourced languages: Application to vietnamese language. *IEEE Transactions on Audio, Speech, and Language Processing* 17.1471–1482.

LEE, CHIAYING, TIMOTHY O'DONNELL, and JAMES GLASS. 2015. Unsupervised lexicon discovery from acoustic input. *Transactions of the Association for Computational Linguistics* 3.389–403.

LEVY, OMER, YOAV GOLDBERG, and IDO DAGAN. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics* 3.211–225.

LEWIS, P. 2009. *Ethnologue: Languages of the World*. SIL International.

LI, SHEN, JOÃO V. GRAÇA, and BEN TASKAR. 2012. Wiki-ly supervised part-of-speech tagging. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, 1389–1398, Jeju Island, Korea.

LUONG, MINH-THANG, HIEU PHAM, and CHRISTOPHER D. MANNING. 2015. Bilingual word representations with monolingual quality in mind. In *NAACL Workshop on Vector Space Modeling for NLP*, Denver, United States.

LYZINSKI, VINCE, GREGORY SELL, and AREN JANSEN. 2015. An evaluation of graph clustering methods for unsupervised term discovery. In *INTERSPEECH*.

MA, XUEZHE, and FEI XIA. 2014. Unsupervised dependency parsing with transferring distribution via parallel guidance and entropy regularization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1337–1348.

MAAS, ANDREW, ZIANG XIE, DAN JURAFSKY, and ANDREW NG. 2015. Lexicon-free conversational speech recognition with neural networks. In *Proceeding of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies . NAACL HLT*, 345–354.

MARCUS, MITCHELL P., BEATRICE SANTORINI, and MARY ANN MARCINKIEWICZ. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* 19.313–330.

MCDONALD, RYAN, JOAKIM NIVRE, YVONNE QUIRMBACH-BRUNDAGE, YOAV GOLDBERG, DIPANJAN DAS, KUZMAN GANCHEV, KEITH HALL, SLAV PETROV, HAO ZHANG, OSCAR TÄCKSTRÖM, CLAUDIA BEDINI, NÚRIA BERTOMEU CASTELLÓ, and JUNGMEE LEE. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 92–97.

——, FERNANDO PEREIRA, KIRIL RIBAROV, and JAN HAJIČ. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, 523–530. Association for Computational Linguistics.

——, SLAV PETROV, and KEITH HALL. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, 62–72.

MERIALDO, BERNARD. 1994. Tagging english text with a probabilistic model. *Computational Linguistics* 20.155–171.

MIKOLOV, TOMAS, KAI CHEN, GREG CORRADO, and JEFFREY DEAN. 2013a. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

——, QUOC V. LE, and ILYA SUTSKEVER. 2013b. Exploiting similarities among languages for machine translation. *CoRR* abs/1309.4168.

——, WEN-TAU YIH, and GEOFFREY ZWEIG. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 746–751, Atlanta, Georgia. Association for Computational Linguistics.

MILLER, GEORGE A. 1995. Wordnet: A lexical database for english. *Commun. ACM* 38.39–41.

MNIH, ANDRIY, and KORAY KAVUKCUOGLU. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems 26*, ed. by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, 2265–2273. Curran Associates, Inc.

NASEEM, TAHIRA, REGINA BARZILAY, and AMIR GLOBERSON. 2012. Selective sharing for multilingual dependency parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, 629–637, Stroudsburg, PA, USA. Association for Computational Linguistics.

NIVRE, JOAKIM, 2002. Two models of stochastic dependency grammar.

——. 2008. Algorithms for deterministic incremental dependency parsing. *Comput. Linguist.* 34.513–553.

——. 2009. Non-projective dependency parsing in expected linear time. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, 351–359, Stroudsburg, PA, USA. Association for Computational Linguistics.

——, ŽELJKO AGIĆ, LARS AHRENBERG, MARIA JESUS ARANZABE, MASAYUKI ASAHARA, AITZIBER ATUTXA, MIGUEL BALLESTEROS, JOHN BAUER, KEPA BENGOETXEA, YEVGENI BERZAK, RIYAZ AHMAD BHAT, CRISTINA BOSCO, GOSSE BOUMA, SAM BOWMAN, GÜLŞEN CEBIROLU ERYIIT, GIUSEPPE G. A. CELANO, ÇAR ÇÖLTEKIN, MIRIAM CONNOR, MARIE-CATHERINE DE MARNEFFE, ARANTZA DIAZ DE ILARRAZA, KAJA DOBROVOLJC, TIMOTHY DOZAT, KIRA DROGANOVA, TOMAŽ ERJAVEC, RICHÁRD FARKAS, JENNIFER FOSTER, DANIEL GALBRAITH, SEBASTIAN GARZA, FILIP GINTER, IAKES GOENAGA, KOLDO GOJENOLA, MEMDUH GOKIRMAK, YOAV GOLDBERG, XAVIER GÓMEZ GUINOVART, BERTA GONZÁLES SAAVEDRA, NORMUNDS GRŪZĪTIS, BRUNO GUILLAUME, JAN HAJIČ, DAG HAUG, BARBORA HLADKÁ, RADU ION, ELENA IRIMIA, ANDERS JOHANNSEN, HÜNER KAŞKARA, HIROSHI KANAYAMA, JENNA KANERVA, BORIS KATZ, JESSICA KENNEY, SIMON KREK, VERONIKA LAIPPALA, LUCIA LAM, ALESSANDRO LENCI, NIKOLA LJUBEŠIĆ, OLGA LYASHEVSKAYA, TERESA LYNN, AIBEK MAKAZHANOV, CHRISTOPHER MANNING, CĂTĂLINA MĂRĂNDUC, DAVID MAREČEK, HÉCTOR MARTÍNEZ ALONSO, JAN MAŠEK, YUJI MATSUMOTO, RYAN MCDONALD, ANNA MISSILÄ, VERGINICA MITITELU, YUSUKE MIYAO, SIMONETTA MONTEMAGNI, KEIKO SOPHIE MORI, SHUNSUKE MORI, KADRI MUISCHNEK, NINA MUSTAFINA, KAILI MÜÜRISEP, VITALY NIKOLAEV, HANNA NURMI, PETYA OSENOVA, LILJA ØVRELID, ELENA PASCUAL, MARCO PASSAROTTI, CENEL-AUGUSTO PEREZ, SLAV PETROV, JUSSI PIITULAINEN, BARBARA PLANK, MARTIN POPEL, LAUMA PRETKALNIA,

PROKOPIS PROKOPIDIS, TIINA PUOLAKAINEN, SAMPO PYYSALO, LOGANATHAN RAMASAMY, LAURA RITUMA, RUDOLF ROSA, SHADI SALEH, BAIBA SAULĪTE, SEBASTIAN SCHUSTER, WOLFGANG SEEKER, MOJGAN SERAJI, LENA SHAKUROVA, MO SHEN, NATALIA SILVEIRA, MARIA SIMI, RADU SIMIONESCU, KATALIN SIMKÓ, KIRIL SIMOV, AARON SMITH, CAROLYN SPADINE, ALANE SUHR, UMUT SULUBACAK, ZSOLT SZÁNTÓ, TAKAAKI TANAKA, REUT TSARFATY, FRANCIS TYERS, SUMIRE UEMATSU, LARRAITZ URIA, GERTJAN VAN NOORD, VIKTOR VARGA, VERONIKA VINCZE, JING XIAN WANG, JONATHAN NORTH WASHINGTON, ZDENĚK ŽABOKRTSKÝ, DANIEL ZEMAN, and HANZHI ZHU, 2016. Universal dependencies 1.3. LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague.

——, and RYAN MCDONALD. 2008. Integrating graphbased and transition-based dependency parsers. In *In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT*, 950–958.

NOTHMAN, JOEL, NICKY RINGLAND, WILL RADFORD, TARA MURPHY, and JAMES R. CURRAN. 2013. Learning multilingual named entity recognition from wikipedia. *Artif. Intell.* 194.151–175.

PARK, ALEX SEUNGRYONG, 2007. *Unsupervised Pattern Discovery in Speech: Applications to Word Acquisition and Speaker Segmentation*. Cambridge, MA, USA: . AAI0818385.

PENNINGTON, JEFFREY, RICHARD SOCHER, and CHRISTOPHER D. MANNING. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

PETROV, SLAV, DIPANJAN DAS, and RYAN MCDONALD. 2012. A universal part-of-speech tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey.

PLANK, BARBARA, DIRK HOVY, and ANDERS SØGAARD. 2014. Learning part-of-speech taggers with inter-annotator agreement loss. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL'14)*, 742–751, Gothenburg, Sweden.

PRYS, DELYTH. The blark matrix and its relation to the language resources situation for the celtic languages. *Strategies for developing machine translation for minority languages* p. 31.

RÄSÄNEN, OKKO. 2012. Computational modeling of phonetic and lexical learning in early language acquisition: Existing models and future directions. *Speech Commun.* 54.975–997.

RÄSÄNEN, OKKO, GABRIEL DOYLE, and MICHAEL C. FRANK. 2015. Unsupervised word discovery from speech using automatic segmentation into syllable-like units. In *INTERSPEECH 2015,*

16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015, 3204–3208.

REDDY, SIVA, and SERGE SHAROFF. 2011. Cross language POS taggers (and other tools) for Indian languages: An experiment with Kannada using Telugu resources. In *Proceedings of IJCNLP workshop on Cross Lingual Information Access: Computational Linguistics and the Information Need of Multilingual Societies. (CLIA 2011 at IJNCLP 2011)*, Chiang Mai, Thailand.

RESNIK, PHILIP, and NOAH A. SMITH. 2003. The web as a parallel corpus. *Comput. Linguist.* 29.349–380.

ROSA, RUDOLF, JAN MAEK, DAVID MAREEK, MARTIN POPEL, DANIEL ZEMAN, and ZDENK ABOKRTSK. 2014. Hamledt 2.0: Thirty dependency treebanks stanfordized. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, ed. by Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, Reykjavik, Iceland. European Language Resources Association (ELRA).

ROTHE, SASCHA, and HINRICH SCHÜTZE. 2015. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. 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)*, 1793–1803, Beijing, China. Association for Computational Linguistics.

SCANNELL, KEVIN P., 2007. The crúbadán project: Corpus building for under-resourced languages.

SCHLIPPE, TIM, SEBASTIAN OCHS, and TANJA SCHULTZ. 2014. Web-based tools and methods for rapid pronunciation dictionary creation. *Speech Communication* 56.101 – 118.

SHRAWANKAR, URMILA, and VILAS M. THAKARE. 2013. Techniques for feature extraction in speech recognition system : A comparative study. *CoRR* abs/1305.1145.

SINISCALCHI, SABATO MARCO, JEREMY REED, TORBJRN SVENDSEN, and CHIN-HUI LEE". 2013. "universal attribute characterization of spoken languages for automatic spoken language recognition ". *"Computer Speech and Language "* 209 – 227.

SOCHER, RICHARD, ALEX PERELYGIN, JEAN WU, JASON CHUANG, CHRISTOPHER D. MANNING, ANDREW NG, and CHRISTOPHER POTTS. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

SØGAARD, ANDERS. 2011. Data point selection for cross-language adaptation of dependency parsers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, 682–686. Association for Computational Linguistics.

——, ŽELJKO AGIĆ, HÉCTOR MARTÍNEZ ALONSO, BARBARA PLANK, BERND BOHNET, and ANDERS JOHANNSEN. 2015. Inverted indexing for cross-lingual nlp. 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)*, 1713–1722, Beijing, China. Association for Computational Linguistics.

STAHLBERG, FELIX, TIM SCHLIPPE, SUE VOGEL, and TANJA SCHULTZ. 2012. Word segmentation through cross-lingual word-to-phoneme alignment. In *IEEE Spoken Language Technology Workshop (SLT)*, 85–90.

STOLCKE, A., F. GREZL, MEI-YUH HWANG, XIN LEI, N. MORGAN, and D. VERGYRI. 2006. Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, I–I.

STUKER, S., T. SCHULTZ, F. METZE, and A. WAIBEL. 2003. Multilingual articulatory features. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, volume 1, I–144–I–147 vol.1.

STÜKER, SEBASTIAN. 2008. Integrating thai grapheme based acoustic models into the ml-mix framework - for language independent and cross-language asr. In *SLTU*.

SUN, LIANG, JASON MIELENS, and JASON BALDRIDGE. 2014. Parsing low-resource languages using gibbs sampling for pcfgs with latent annotations. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 290–300, Doha, Qatar. Association for Computational Linguistics.

TÄCKSTRÖM, OSCAR, DIPANJAN DAS, SLAV PETROV, RYAN MCDONALD, and JOAKIM NIVRE. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics* 1.1–12.

TÄCKSTRÖM, OSCAR, RYAN MCDONALD, and JOAKIM NIVRE. 2013. Target language adaptation of discriminative transfer parsers. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1061–1071, Atlanta, Georgia. Association for Computational Linguistics.

——, RYAN MCDONALD, and JAKOB USZKOREIT. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 Conference of the North American Chap-*

ter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT '12, 477–487. Association for Computational Linguistics.

TAI, KAI SHENG, RICHARD SOCHER, and CHRISTOPHER D. MANNING. 2015. Improved semantic representations from tree-structured long short-term memory networks. *CoRR* abs/1503.00075.

THOMAS, S., S. GANAPATHY, and H. HERMANSKY. 2012. Multilingual mlp features for low-resource lvcsr systems. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4269–4272.

TIAN, FEI, HANJUN DAI, JIANG BIAN, BIN GAO, RUI ZHANG, ENHONG CHEN, and TIE-YAN LIU. 2014. A probabilistic model for learning multi-prototype word embeddings. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 151–160, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

TIEDEMANN, JÖRG. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, ed. by Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA). ACL Anthology Identifier: L12-1246.

TOUTANOVA, KRISTINA, DAN KLEIN, CHRISTOPHER D. MANNING, and YORAM SINGER. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1 (NAACL '03)*, 173–180, Edmonton, Canada.

TSAI, CHEN-TSE, STEPHEN MAYHEW, and DAN ROTH. 2016. Cross-lingual named entity recognition via wikification. In *CoNLL*.

TURIAN, JOSEPH, LEV RATINOV, and YOSHUA BENGIO. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, 384–394, Stroudsburg, PA, USA. Association for Computational Linguistics.

UPADHYAY, SHYAM, MANAAL FARUQUI, CHRIS DYER, and DAN ROTH. 2016. Cross-lingual models of word embeddings: An empirical comparison. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1661–1670, Berlin, Germany. Association for Computational Linguistics.

VERSTEEGH, MAARTEN, XAVIER ANGUERA, AREN JANSEN, and EMMANUEL DUPOUX. 2016. The zero resource speech challenge 2015: Proposed approaches and results. *Procedia Computer*

Science 81.67 – 72. SLTU-2016 5th Workshop on Spoken Language Technologies for Under-resourced languages 09-12 May 2016 Yogyakarta, Indonesia.

VESEL, K., M. KARAFIT, F. GRZL, M. JANDA, and E. EGOROVA. 2012. The language-independent bottleneck features. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*, 336–341.

VILLING, RUDI, JOSEPH TIMONEY, and TOMAS WARD. 2004. Automatic blind syllable segmentation for continuous speech. In *Irish Signals and Systems Conference 2004*.

——, TOMAS WARD, and JOSEPH TIMONEY. 2006. Performance limits for envelope based automatic syllable segmentation. In *Irish Signals and Systems Conference, 2006. IET*, 521–526. IET.

VRIES, NIC, MARELIE H. DAVEL, JACO BADENHORST, WILLEM D. BASSON, FEBE DE WET, ETIENNE BARNARD, and ALTA DE WAAL. 2014. A smartphone-based {ASR} data collection tool for under-resourced languages. *Speech Communication* 56.119 – 131.

VULIĆ, IVAN, and MARIE-FRANCINE MOENS. 2015. Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 719–725, Beijing, China. Association for Computational Linguistics.

WANG, MENGQIU, WANXIANG CHE, and CHRISTOPHER D. MANNING. 2013. Joint word alignment and bilingual named entity recognition using dual decomposition. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1073–1082, Sofia, Bulgaria. Association for Computational Linguistics.

——, and CHRISTOPHER D. MANNING. 2013. Cross-lingual pseudo-projected expectation regularization for weakly supervised learning. *CoRR* abs/1310.1597.

XIA, FEI, and WILLIAM LEWIS. 2007. Multilingual structural projection across interlinear text. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, 452–459, Rochester, New York. Association for Computational Linguistics.

XIAO, MIN, and YUHONG GUO. 2014. *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, chapter Distributed Word Representation Learning for Cross-Lingual Dependency Parsing, 119–129. Association for Computational Linguistics.

YAMADA, HIROYASU, and YUJI MATSUMOTO. 2003. Statistical dependency analysis with support vector machines. In *In Proceedings of IWPT*, 195–206.

YANG, YI, and JACOB EISENSTEIN. 2016. Part-of-speech tagging for historical english. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies, 1318–1328, San Diego, California. Association for Computational Linguistics.

YAROWSKY, DAVID, and GRACE NGAI. 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, NAACL '01, 1–8, Pittsburgh, Pennsylvania.

YOUNGER, DANIEL H. 1967. Recognition and parsing of context-free languages in time n3. *Information and Control* 10.189 – 208.

ZEMAN, DANIEL, UNIVERZITA KARLOVA, and PHILIP RESNIK. 2008. Cross-language parser adaptation between related languages. In *In IJCNLP-08 Workshop on NLP for Less Privileged Languages*, 35–42.

——, DAVID MARECEK, MARTIN POPEL, LOGANATHAN RAMASAMY, JAN STEPÁNEK, ZDENEK ZABOKRTSKÝ, and JAN HAJIC. 2012. Hamledt: To parse or not to parse? In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, 2735–2741.

ZHANG, Y., R. SALAKHUTDINOV, H. A. CHANG, and J. GLASS. 2012. Resource configurable spoken query detection using deep boltzmann machines. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5161–5164.

ZHANG, YUAN, and REGINA BARZILAY. 2015. Hierarchical low-rank tensors for multilingual transfer parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1857–1867, Lisbon, Portugal. Association for Computational Linguistics.

ZHANG, YUE, and STEPHEN CLARK. 2008. A tale of two parsers: Investigating and combining graph-based and transition-based dependency parsing using beam-search. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, 562–571, Stroudsburg, PA, USA. Association for Computational Linguistics.

ZOPH, BARRET, and KEVIN KNIGHT. 2016. Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 30–34, San Diego, California. Association for Computational Linguistics.

ZOU, WILL Y., RICHARD SOCHER, DANIEL CER, and CHRISTOPHER D. MANNING. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1393–1398, Seattle, Washington, USA. Association for Computational Linguistics.

Appendix A

Other Papers

this is other chapter