Transfer Learning for Low-resource Natural Language Analysis

by

Yuan Zhang

B.S., Tsinghua University (2011)
M.S., Massachusetts Institute of Technology (2013)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2017

Author .....
Department of Electrical Engineering and Computer Science
January 31, 2017

Certified by .....
Regina Barzilay
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair of the Committee on Graduate Students

Transfer Learning for Low-resource Natural Language Analysis

by

Yuan Zhang

Submitted to the Department of Electrical Engineering and Computer Science
on January 31, 2017, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Expressive machine learning models such as deep neural networks are highly effective when they can be trained with large amounts of in-domain labeled training data. While such annotations may not be readily available for the target task, it is often possible to find labeled data for another related task. The goal of this thesis is to develop novel transfer learning techniques that can effectively leverage annotations in source tasks to improve performance of the target low-resource task. In particular, we focus on two transfer learning scenarios: (1) transfer across languages and (2) transfer across tasks or domains in the same language.

In multilingual transfer, we tackle challenges from two perspectives. First, we show that linguistic prior knowledge can be utilized to guide syntactic parsing with little human intervention, by using a hierarchical low-rank tensor method. In both unsupervised and semi-supervised transfer scenarios, this method consistently outperforms state-of-the-art multilingual transfer parsers and the traditional tensor model across more than ten languages. Second, we study lexical-level multilingual transfer in low-resource settings. We demonstrate that only a few (e.g., ten) word translation pairs suffice for an accurate transfer for part-of-speech (POS) tagging. Averaged across six languages, our approach achieves a 37.5% improvement over the monolingual top-performing method when using a comparable amount of supervision.

In the second monolingual transfer scenario, we propose an aspect-augmented adversarial network that allows aspect transfer over the same domain. We use this method to transfer across different aspects in the same pathology reports, where traditional domain adaptation approaches commonly fail. Experimental results demonstrate that our approach outperforms different baselines and model variants, yielding a 24% gain on this pathology dataset.

Thesis Supervisor: Regina Barzilay
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

First and foremost, I'm forever thankful to my advisor Regina Barzilay for her never-ending guidance and support throughout the six years of my PhD. When I came to MIT six years ago, I knew nothing about Natural Language Processing (NLP). Parsing, grounding, morphology – I heard those words for the first time in my life. Regina shows her brilliant and unique ways of leading me to plot my research path on the blank paper. She always had patience with all my faulty ideas, and shaped them with her insights into innovative research directions. Her high standards always help me produce solid and impactful research. This thesis would not have been possible without her – she was the best advisor that I could have ever asked for.

I am also thankful to my fantastic thesis committee, Tommi Jaakkola and Jim Glass. They gave me invaluable advice and comments on my thesis throughout the whole process. Beyond the committee, I have been very fortunate to have a chance to collaborate with Tommi for the last three years of my graduate studies. What I will miss most, are those conversations with him that start with a chaos and incomplete idea but end up with an elegant math solution.

I would also like to give my special thanks to Amir Globerson of Tel Aviv University. I was fortunate enough to have collaboration with him during my early years as a graduate student. I really learned a lot from him about classical machine learning and optimization algorithms that have been super useful in my research life. He is also such a nice guy that I have really enjoyed chatting with, even remotely.

Many thanks must also go to my collaborators, my friends and all the wonderful people in my group. I have benefited tremendously from conversations and assistance from Yonatan Belinkov, Yevgeni Berzak, S.R.K. Branavan, Benson Chen, Desai Chen, Kareem Darwish, Abdi Dirie, David Gaddy, Youyang Gu, Zach Hynes, Wengong Jin, Hrishikesh Joshi, Nate Kushman, Yoong Keok Lee, Tao Lei, Eduardo De Leon, Chengtao Li, Nick Locascio, Fan Long, Jiaming Luo, David Alvarez Melis, Lluís Màrquez, Alessandro Moschitti, Karthik Narasimhan, Tahira Naseem, Roi Reichart, Christy Sauper, Elena Sergeeva, Darsh Shah, Tianxiao Shen, Yu Xin, Adam Yala, Yu

Zhang. I will never forget all those discussions (about any topics) with you at the fourth floor of the Stata Center, and will always miss those outdoor activities with the group. I also want to give special thanks to Marcia Davidson for her help with all administrative things.

A big thank you to my wife Xin. Getting married with her is the most fortunate thing ever in my life. I started gaining weight after she coming and cooking the world's best food for me. I am forever indebted to her for her supports, loves, and being my best friend.

Finally, I would like to thank my parents Haiyan and Haizhou. Many thanks for their supports and patience to me through the entire PhD process. They always believed in me and encouraged me when I lost confidence in myself. I dedicate this thesis to my wonderful family.

Bibliographic Notes

The main ideas in this thesis have been previously published in two peer-reviewed conferences while one is currently under peer-review. The list of publications by chapter is as follows:

- **Chapter 2: Hierarchical Low-Rank Tensors for Multilingual Transfer Parsing** *In proceedings of Empirical Methods in Natural Language Processing (EMNLP), 2015.*
- **Chapter 3: Ten Pairs to Tag – Multilingual POS Tagging via Coarse Mapping between Embeddings** *In proceedings of North American Chapter of the ACL (NAACL), 2016.*
- **Chapter 4: Aspect-augmented Adversarial Networks for Domain Adaptation** *submitted to Transactions of the Association for Computational Linguistics (TACL), currently under peer-review*

The code for the work presented in this thesis are publicly available at `https://github.com/yuanzh`.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Today, formulating and learning expressive neural models has become a paradigm of choice across a wide range of natural language processing (NLP) tasks. While flexible continuous representations afforded by these models often lead to well-performing systems, learning them requires significant amounts of well-annotated data. To achieve top performance, these models typically need to be trained on more than millions of tokens annotated for each specific task, such as syntactic parsing [3], question answering [105, 45] and machine translation [107, 4].

While such large amounts of annotations exist in some cases, they are not readily available in other scenarios. For example, the Universal Dependency Treebank [77] consists of syntactic treebanks for more than 20 languages. However, these treebanks only cover a small fraction of the thousands of existing world languages. Many of these languages, such as Kinyarwanda and Malagasy, have insufficient annotated parse trees, and thus are beyond the scope of state-of-the-art supervised parsers. Even in a resource-rich language like English, the annotation sparsity issue also exists when we are confronted with different domains. A common situation is that large amounts of labeled data are available in one domain, such as news articles, but we truly desire the model to perform well in another low-resource domain, say medical reports. However, a state-of-the-art model trained on news articles will have inferior performance in the medical domain because vocabulary and writing style vary so widely across domains.

Despite the fact that the lack of annotated resources has become significant ob-

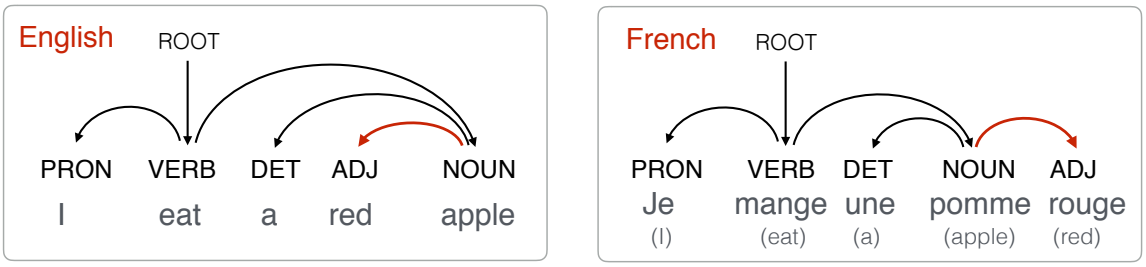

Figure 1-1: An English sentence and its translation in French annotated with universal POS tags and dependency trees. Universal POS tags are used as non-lexical features to alleviate the vocabulary discrepancy between languages.

stacle to achieving high performance, developing such resources in large quantities is sometimes an extremely laborious task. The development process can be prohibitively expensive because of the level of linguistic expertise required to produce target annotations, such as syntactic trees. This unforgiving obstacle motivates the ultimate goal of my research which is to minimize the amount of annotated data required to achieve the state-of-the-art performance. As a step towards this goal, this thesis explores the use of transfer learning techniques to address the challenges in low-resource natural language analysis.

The key idea of transfer learning is to capitalize on rich annotations in a *source* field, while our *target* field is resource-poor [78]. The goal of learning is to automatically and robustly handle the discrepancy between the two fields and enable information transfer from *source* to *target* to improve model performance. In this thesis, I tackle the challenges in transfer learning in the context of two scenarios: (1) transfer across languages and (2) transfer across tasks or domains in the same language.

Transfer Learning across Languages The first transfer learning scenario is to parse and analyze a *target* language by utilizing annotations available in other *source* languages. In this scenario, a model that depends on lexical features is not directly applicable because of the vocabulary discrepancy between two languages. As a remedy, recent methods have been relying solely on non-lexical features that are available

in both the source and the target languages, such as universal part-of-speech (POS) tags [68, 75, 97]. Figure 1-1 shows an example sentence in English (left) and its translation in French (right) annotated with universal POS tags. While the words are totally different, their POS sequences look similar to each other. Thus, given these universal POS annotations, a parser trained on an English treebank can also correctly parse sentences in French.

However, in many NLP tasks those non-lexical features alone are never sufficient for models to reach respectable performance. For example, a state-of-the-art supervised POS tagger primarily relies on lexical features such as context words, prefixes and suffixes. It is next to impossible to achieve the same level of performance with only non-lexical features. Therefore, lexical-level transfer is necessary for multilingual POS tagging. In order to achieve lexical-level transfer, prior methods typically make use of significant parallel resources such as parallel translations or bilingual dictionaries [26, 96, 92]. These resources act as substitutes for explicit annotations available in the target language for supervised methods. Such parallel resources, however, are not always available in sufficient amount in real life. It is less clear what can be done without such extensive parallel resources. Indeed, one motivation of this thesis comes from trying to understand how little parallel data is necessary for effective multilingual transfer.

Another challenge in multilingual transfer is to handle linguistic differences across languages, such as linguistic typology related to word ordering. As shown in Figure 1-1, an adjective typically comes before a noun in English while this order is reversed in French. Therefore, such syntactic property should not be transferred between these two languages. While such linguistic knowledge is publicly available online for many languages [25], encoding this knowledge in models is non-trivial. Previous work have implemented this selective transfer idea by heavy feature engineering [97], which is time consuming and hard to scale. As an alternative to this manual process, it is necessary to design an automatic and systematic approach of inducing feature representations tailored for target tasks (e.g. syntactic parsing) as well as incorporating prior knowledge on linguistic typology.

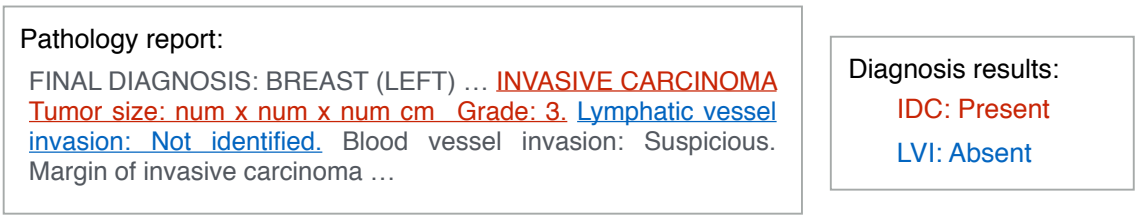

Figure 1-2: A snippet of a breast pathology report with diagnosis results for two types of disease. Evidence for both results is in red and blue, respectively.

Transfer Learning across Domains In the second scenario, we study how to learn a model from annotations in a *source* domain or task that generalizes well to another related *target* domain or task. In this thesis we are primarily interested in transfer between two classification tasks over the same domain, i.e., over the same set of examples. We call this *aspect transfer* as the two classification tasks can be thought to pertain to different aspects of the same examples. As an example, consider Figure 1-2 that shows a pathology report with diagnosis results (presence or absence) on two types of breast disease: lymph invasion (LVI) and invasive carcinoma (IDC). The target goal may be to classify the pathology report for the presence of LVI but the available training data involve only annotations for IDC in the same report. Traditional domain adaptation approaches are likely to fail on this task because they only predict the category based on an overall feature representation of the reports. In this task, however, both aspects have to be predicted from the same document, so in both cases the classifier operates over the same representation from the very beginning and cannot distinguish between different aspects. To address this challenge, the model should be capable to learn to extract information that is relevant to a particular aspect while ignoring the rest.

Recently, the employment of deep learning methods brings new challenges in domain adaptation problem. While complex neural models have achieved top performance in the supervised setting, they typically have little transparency in their inner mapping process from texts to corresponding vector representations. Essentially, this hidden mapping process makes it intractable to exactly measure the distribution dissimilarity across domains, thus opening a new research question on how to explicitly

encourage the learning of domain-invariant representations for adaptation, which is another focus of this thesis.

In this thesis, we consider transfer learning in low-resource settings. We are primarily interested in tackling challenges under the following three low-resource setups:

- **Little Annotations on Target Tasks:** First, as in standard low-resource scenarios, we assume little or no label annotations on the target side. We only have raw data for the target task.
- **Little Parallel Data:** Much previous work relies on large amounts of parallel data to enable transfer. In contrast, we assume no parallel data is available, or we use only a few (e.g., ten) word translation pairs.
- **Low Level of Human Effort:** Our methods require low level of human intervention on model design, reflected in two perspectives. First, our method requires little manual feature engineering to encode prior knowledge for transfer in the model. Second, we guide the aspect transfer by using only a small number of manual keyword rules.

We illustrate our techniques to address the above challenges in the context of three transfer learning tasks: (1) multilingual dependency parsing with no parallel data and little manual engineering, (2) multilingual POS tagging with only a few word translation pairs, and (3) aspect transfer with only a few keyword rules. Next I will briefly describe the challenges in each task and our approaches to solving these challenges.

1.1 Multilingual Parsing without Feature Engineering

Task and Challenge Our first task is multilingual dependency parsing where the goal is to train a dependency parser for a resource-poor language by using annotations in other resource-rich languages. Accurate multilingual transfer parsing typically relies on careful feature engineering by hand. Figure 1-3 shows examples of manually crafted features, including ones that are selectively shared and ones that are universal across languages. However, this manual feature engineering is time consuming. An appealing alternative to this process is tensor-based scoring models. These models automatically consider all possible combinations of atomic features, and address the parameter explosion problem via a low-rank assumption. Figure 1-3 depicts a four-way tensor that captures all combinations of head POS, modifier POS, arc direction and typology values. One trait of traditional tensor-based scoring is that no prior knowledge about feature interactions is assumed. In the multilingual transfer setting, however, we have some prior knowledge about legitimate feature combinations. As shown in the bottom part of Figure 1-3, the feature combination in blue is valid, but the one in red is invalid. This is because the preference of noun-adjective order should be specifically associated to a noun-adjective arc, not a verb-noun arc. However, the traditional tensor technique still considers these unobserved feature combinations, and assigns them non-zero weights.

Our Approach To address this issue, we introduce a hierarchical tensor model that enables us to incorporate prior knowledge about desired feature interactions, eliminating invalid feature combinations. The hierarchical structure uses intermediate embeddings to capture desired feature combinations. Algebraically, this hierarchical tensor is equivalent to the sum of traditional tensors with shared components, and thus can be effectively trained with standard online learning algorithms. Empirically, we demonstrate that our hierarchical tensor consistently improves parsing accuracy over baselines across more than ten different languages.

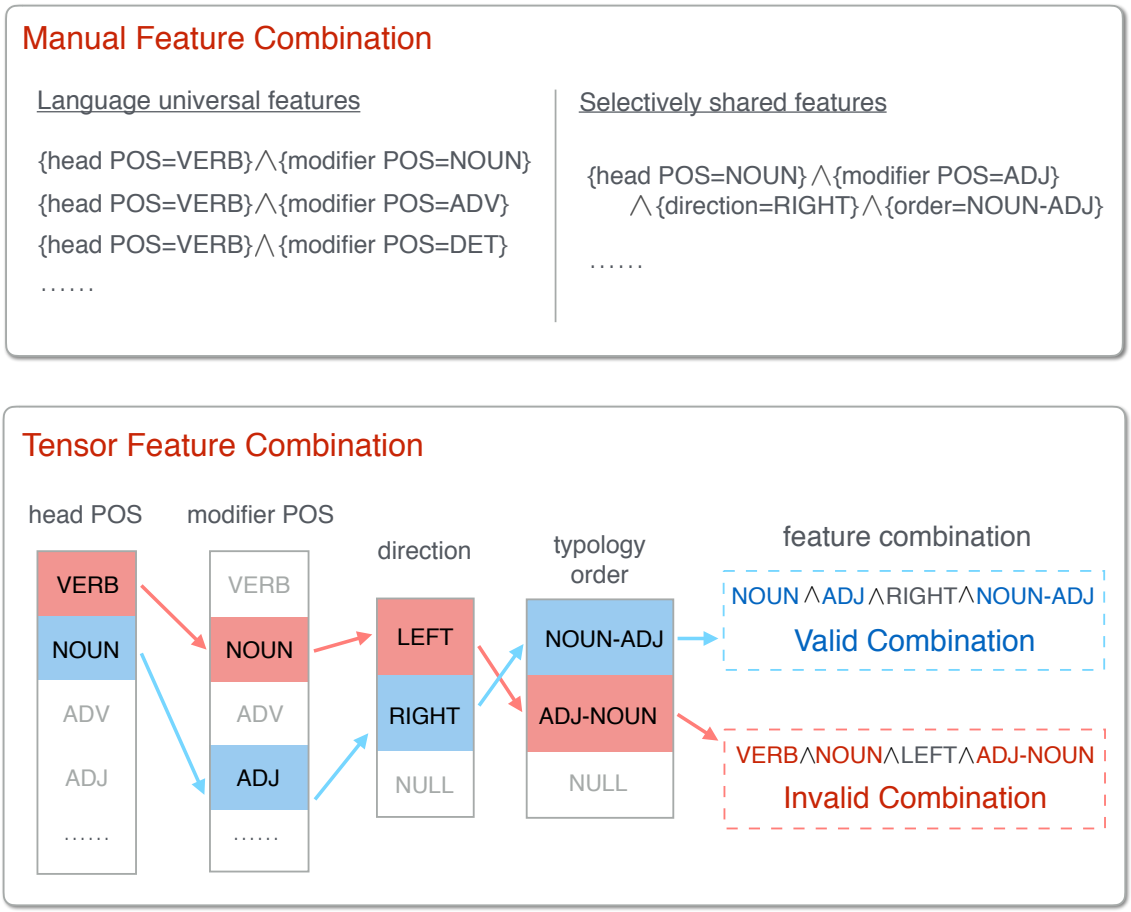

Figure 1-3: Examples of manual (top) and tensor features combinations (bottom). Previous state-of-the-art multilingual transfer parsers require heavy feature engineering on the design of language universal features and selective shared features. In contrast, the example tensor automatically captures all feature combinations over head POS, modifier POS, direction and typology order of noun-adjective. Note that, however, tensor methods also capture invalid feature combinations.

1.2 Multilingual POS Tagging with Ten Translation Pairs

Task and Challenge In our second multilingual transfer application, we choose POS tagging as the underlying task. Unlike multilingual parsing, lexical-level transfer is essential in this task. A common solution is to utilize multilingual word embeddings as a universal representation of words to facilitate the lexical-level transfer. In particular, monolingual embeddings are aligned between languages such that translation word pairs have the same representations. This enables the supervised source model expressed in terms of embeddings to be "directly transferred" on the target language. However, learning such multilingual embeddings typically requires large amounts of parallel data that are not available in many cases.

Our Approach In this work, we demonstrate that only ten word translation pairs suffice for effective multilingual transfer of POS tagging. The key idea is still to align monolingual embeddings to transfer annotations. However, a full fine-grained alignment is not possible with only ten translation pairs due to differences between the languages and variations across raw corpora from which the embeddings are derived. Instead, we restrict the coarse mapping to be linear and isometric (orthonormal) so as to leave lengths and angles between the word vectors invariant. One advantage is that this preserves cosine similarity between vectors, which is viewed as a proxy for syntactic/semantic similarity [70, 80, 44]. Figure 1-4 shows an example of the coarse mapping process using three translation pairs.

We further refine the model in an unsupervised manner by initializing and regularizing it to be close to the direct transfer model. While unsupervised methods are fragile and challenging to estimate in general, they can be helpful if initialized and regularized properly. The refined model is able to capture language-specific syntactic properties and to fit the target language better. Averaged across six languages, our model yields a 37.5% absolute improvement over the monolingual prototype-driven method [42] when using a comparable amount of supervision.

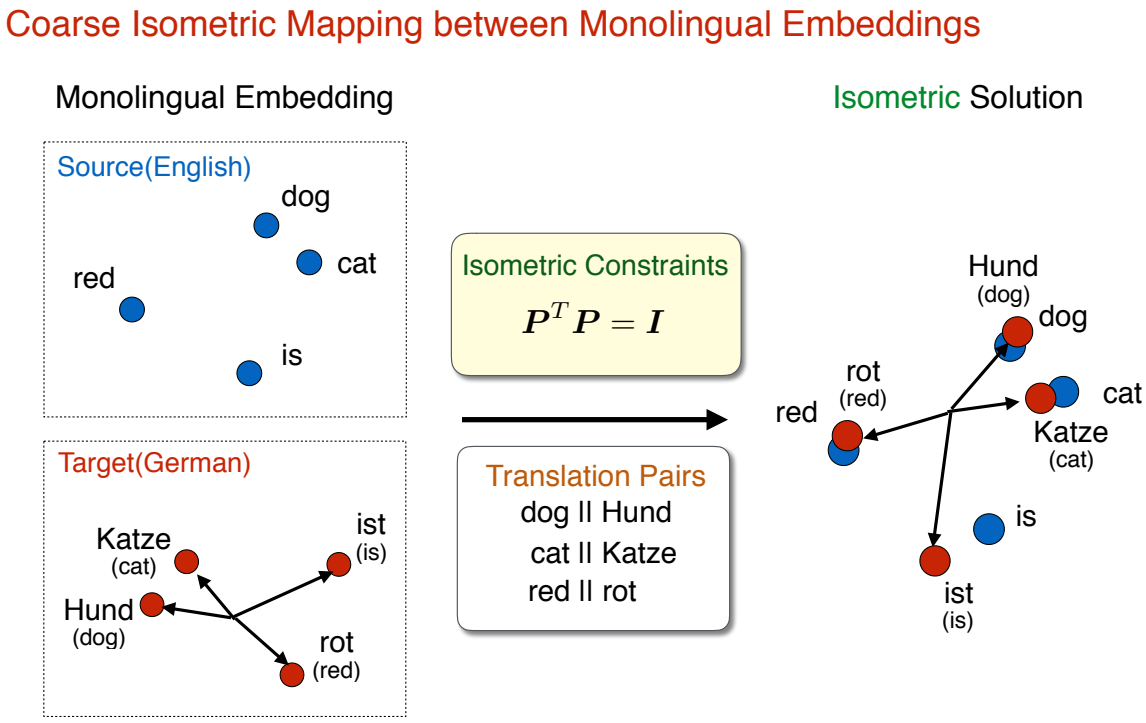

Figure 1-4: An example of coarse mapping between monolingual embeddings with isometric constraints. Isometric transformation preserves angles and lengths (cosine similarity) of word vectors, thus preserving semantic relations. Embeddings are aligned using three translation pairs in this example.

1.3 Aspect Transfer with Few Keyword Rules

Task and Challenge We formulate our final task as an aspect transfer problem where the goal is to predict labels pertaining to a particular aspect in texts. As illustrated in Figure 1-2, if the model aims to predict labels for lymph invasion (LVI), it should extract features only from the second sentence (in blue) while ignoring the rest. To solve this task, the model must learn to properly relate the source and target aspects. Moreover, unlike both multilingual transfer tasks above, we have no linguistic prior knowledge to guide transfer in this case. The model needs to learn domain-invariant feature representations in an unsupervised manner.

Our Approach To enable aspect transfer, we propose an aspect-driven document encoder that can selectively extract features from aspect-relevant fragments while ignoring the rest. Specifically, we equip the encoder with a sentence-level relevance scorer that allows the model to select aspect-relevant sentences from the document. Instead of target labels, we assume a small set of keywords pertaining to each aspect as a form of weak supervision for learning the scorer. Such annotations can be easily provided by domain experts, such as extracting from medical literature such as codex rules in pathology [79]. On the pathology dataset, we show that this relevance scorer brings 24% absolute gain in prediction accuracy.

Our relevance-driven encoder returns the aspect transfer problem closer to the realm of standard domain adaptation. To support generalization across aspects, we employ an adversarial neural network to learn aspect-invariant representations. Figure 1-5 illustrates the idea of adversarial training. We train a domain/aspect classifier that naturally provides an effective measurement on cross-aspect dissimilarities. In other words, the classifier will perform poorly if features are aspect-invariant. We therefore jointly optimize the encoder to counteract the classifier as an adversary. Experimental results show that our model significantly outperforms the no-adversarial-training baseline, achieving a 12.8% gain on the pathology dataset.

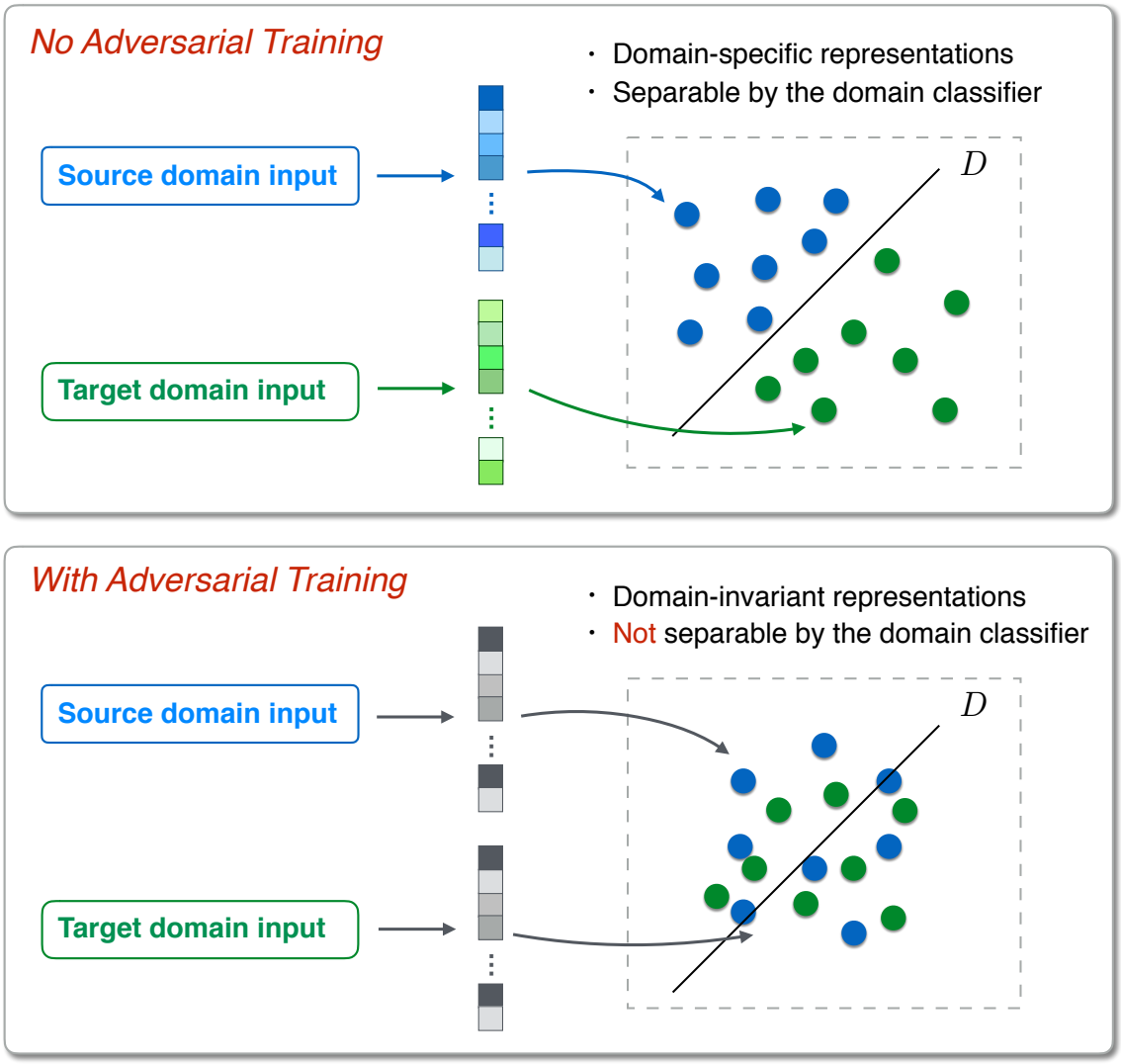

Figure 1-5: An illustration of the idea of using adversarial training for domain adaptation. When no adversarial training is used (top), learned representations consist of domain-specific features and are separable by a discriminative domain classifier. In contrast, adversarial training encourages the emergence of domain-invariant features that are not separable by the domain classifier (bottom).

1.4 Contributions

The primary contribution of this thesis is four-fold:

- **Incorporating linguistic knowledge in tensor-based models** We propose a novel hierarchical tensor based-model for multilingual transfer parsing. This approach enables us to constrain learned representation based on desired feature interactions. We demonstrate that our model outperforms state-of-the-art multilingual transfer parsers and traditional tensors.

- **Guiding lexical-level transfer with only ten translation pairs** We demonstrate that ten translation pairs suffice for an effective multilingual transfer of POS tagging. The effectiveness of our approach suggests its potential application to a broader range of NLP tasks that require lexical-level multilingual transfer, such as machine translation and entity recognition.

- **Handling Aspect-driven transfer using relevance scoring** We show that the aspect-relevance scoring enables aspect transfer. We demonstrate that this technique is particularly important in scenarios where we transfer across different aspects in the same document, such as parsing pathology reports.

- **Learning domain-invariant representations using adversarial training** We present an adversarial network for domain adaption. As shown by our results, we demonstrate the effectiveness of adversarial training to guide the learning of domain-invariant representations. We also provide extensive analysis on the behavior of adversarial training. The success of our work provides insights into using adversarial training in many other NLP tasks.

1.5 Outline

The remainder of this thesis is organized as follows:

- **Chapter 2** proposes a hierarchical low-rank tensor method for multilingual transfer parsing. This method uses linguistic typology knowledge to constrain the feature construction process in the tensor structure.
- **Chapter 3** presents our coarse embedding mapping method that uses just ten word translation pairs to achieve effective POS transfer.
- **Chapter 4** describes our aspect-augmented adversarial network for transfer learning between two aspects over the same domain.
- **Chapter 5** summarizes the thesis and directions for future work.

Chapter 2

Hierarchical Low-Rank Tensors for Multilingual Transfer Parsing

In this chapter, we consider the task of multilingual transfer for dependency parsing. By using a hierarchical tensor-based approach, we demonstrate that linguistic typology knowledge can be encoded in a tensor-based model by explicitly modeling the feature combination process in the tensor structure. In both unsupervised and semi-supervised settings, we show that our method consistently outperforms previous state-of-the-art approaches on a broad range of transfer scenarios.

2.1 Introduction

The goal of multilingual syntactic transfer is to parse a resource-lean target language utilizing annotations in other resource-rich source languages. Recent approaches have demonstrated that such transfer is possible, even in the absence of parallel data. As a main source of guidance, these methods rely on the commonalities in dependency structures across languages. In particular, these commonalities manifest themselves as a broad and diverse set of indicator features, ranging from standard non-lexical arc features used in monolingual parsers to linguistic typological properties needed to guide cross-lingual sharing (e.g., verb-subject ordering preference). Table 2.1 shows two examples of typological features on word ordering preferences and the correspond-

ID	Description	English	Spanish	French
82A	Order of Subject and Verb	SV	SV	SV
87A	Order of Adjective and Noun	Adj-Noun	Noun-Adj	Noun-Adj

Table 2.1: Example linguistic typological features from WALS. 82A and 87A denote the feature codes for verb-subject and noun-adjective ordering preferences, respectively. All three languages have the same preference on verb-subject ordering, but their noun-adjective ordering preferences are different.

ing feature values of three languages. All three languages have the same preference on verb-subject ordering, but different values on noun-adjective typology. Therefore, the parameters pertaining to the direction of a noun-adjective arc should be selectively shared between French and Spanish, not from English.

Prior state-of-the-art multilingual transfer parsers implement the idea of selective sharing by careful feature engineering [97]. Consider the example feature in Table 2.2 that combines a standard arc feature with a noun-adjective typological feature value. This feature only fires when the parser operates on a sentence in Spanish or French due to the presence of the typological features. When testing on French sentences, the parser thus selectively transfers the corresponding syntactic pattern from Spanish, not English. As shown in [97], such selectively shared features play a crucial role in cross-lingual transfer parsing. However, constructing these features also requires a high level of expertise and significant amount of human effort.

Tensor-based models are an appealing alternative to this manual feature design process. These models represent the high-dimensional feature vectors as tensor-products of multiple smaller vectors that encode atomic features, such as the head POS. The associated parameters are viewed as a tensor of low rank. By factorizing the tensor represented in a low-rank form, the model automatically induces a compact low-dimensional feature representation that are specifically tailored for parsing accuracy. These tensor-based methods assume no prior knowledge about feature interactions. As a result, the model considers all possible combinations of these atomic

Verb-subject: {head POS=VERB} ∧ {modifier POS=NOUN} ∧ {label=subj} ∧ {direction =LEFT} ∧ {82A=SV}
Noun-adjective: {head POS=NOUN} ∧ {modifier POS=ADJ} ∧ {direction=RIGHT} ∧ {87A= Noun-Adj}

Table 2.2: Example verb-subject and noun-adjective typological features. 82A and 87A denote the WALS [25] feature codes for verb-subject and noun-adjective ordering preferences.

features, and addresses the parameter explosion problem via a low-rank assumption.

In the multilingual transfer setting, however, we have some prior knowledge about legitimate feature combinations. For instance, consider the linguistic typological feature that encodes verb-subject preferences, as shown in Table 2.2. This features is expressed as a conjunction of five atomic features. Ideally, we would like to treat this composition as a single non-decomposable feature. However, the traditional tensor model decomposes this feature into five dimensions, and considers various combinations of these features as well as their individual interactions with other features. Specifically, the model captures all possible combinations of atomic features over head POS, modifier POS, label, direction and typological feature values. In practice, we want to avoid invalid combinations that conjoin the typological feature with unrelated atomic features. For instance, there is no point to constructing features of the form {head POS=ADJ} ∧ {head POS=VERB} ∧ ⋯ ∧ {82A=SV} as the head POS can only take a single value. However, the traditional tensor technique still considers these unobserved feature combinations, and assigns them non-zero weights (see Section 2.7). This inconsistency between prior knowledge and the low-rank assumption results in a sub-optimal parameter estimation.

To address this issue, we introduce a hierarchical tensor model that constrains parameter representation. The model encodes prior knowledge by explicitly excluding undesired feature combinations over the same atomic features. At the bottom level

of the hierarchy, the model constructs combinations of atomic features, generating intermediate embeddings that represent the legitimate feature groupings. For instance, these groupings will not combine the verb-subject ordering feature and the POS head feature. At higher levels of the hierarchy, the model combines these embeddings as well as the expert-defined typological features over the same atomic features. The hierarchical tensor is thereby able to capture the interaction between features at various subsets of atomic features. Algebraically, the hierarchical tensor is equivalent to the sum of traditional tensors with shared components. Thus, we can use standard online algorithms for optimizing the low-rank hierarchical tensor.

We evaluate our model on labeled dependency transfer parsing using several universal dependency treebanks. We first use the multilingual universal dependency treebank v2.0 [77, 69]. We compare our model against the state-of-the-art multilingual transfer dependency parser [97] and the direct transfer model [68]. All the parsers utilize the same training resources but with different feature representations. When trained on source languages alone, our model outperforms the baselines for seven out of ten languages on both unlabeled attachment score (UAS) and labeled attachment score (LAS). On average, it achieves 1.1% UAS improvement over [97]'s model and 4.8% UAS over the direct transfer. We also consider a semi-supervised setting where multilingual data is augmented with 50 annotated sentences in the target language. In this case, our model achieves improvement of 1.7% UAS over [97]'s model and 4.5% UAS over the direct transfer. Moreover, we demonstrate that our model closes the performance gap between training on the 50 sentences and on the full training set by about 30% on UAS and LAS.

In addition, we evaluate our model on the multilingual universal dependency treebank v1.0 [77, 69]. We use 3,000 annotated tokens in the target language as well as all annotations from other source languages. We compare our model against a recent neural network-based multilingual transfer parser [29]. Both our method and this baseline parser use the same amount of annotations as supervision. Out of nine languages, our model achieves better LAS on seven languages. On average, our model outperforms the baseline by 2.5% on LAS. We also compare against the state-of-the-

art supervised parser. By utilizing source languages, our model achieves a 0.9% gain on LAS.

The remainder of this chapter is organized as follows. We first describe prior related work on multilingual parsing and tensor-based models in Section 2.2. In Section 2.3, we introduce background of tensor-based methods for parsing followed by detail description of our proposed hierarchical tensor approach for multilingual transfer parsing. The following two sections, 2.4 and 2.5 describe details on model learning and features, respectively. Finally, we present applications and experimental results of our model in Section 2.6 and 2.7 before concluding in Section 2.8.

2.2 Related Work

2.2.1 Multilingual Parsing

The lack of annotated parsing resources for the vast majority of world languages has kindled significant interest in multi-source parsing transfer [46, 30, 118, 116, 19, 86, 41]. Recent research has focused on the non-parallel setting, where transfer is driven by cross-lingual commonalities in syntactic structure [74, 97, 5, 18, 29].

Our work is closely related to the selective-sharing approaches [75, 97]. The core of these methods is the assumption that head-modifier attachment preferences are universal across different languages. However, the sharing of arc direction is selective and is based on linguistic typological features. While this selective sharing idea was first realized in the generative model [75], higher performance was achieved in a discriminative arc-factored model [97]. These gains were obtained by a careful construction of features templates that combine standard dependency parsing features and typological features. However, manually constructing these features require a high level of expertise and significant human effort. In contrast, we propose an automated, tensor-based approach that can effectively capture the interaction between these features, yielding a richer representation for cross-lingual transfer. Moreover, our model handles labeled dependency parsing while previous work only focused on the unlabeled dependency parsing task.

2.2.2 Tensor-based Models

Our approach also relates to prior work on tensor-based modeling. Lei et al. [53] employ three-way tensors to obtain a low-dimensional input representation optimized for parsing performance. Later, Lei et al. [54] apply the same idea and introduce a four-way tensor-based approach to semantic role labeling (SRL). Srikumar et al. [95] learn a multi-class label embedding tailored for document classification and POS tagging in the tensor framework. Yu et al. [115] and Fried et al. [34] apply low-rank tensor decompositions to learn task-specific word and phrase embeddings. Other applications

of tensor framework include low-rank regularization [83, 84, 90] and neural tensor networks [93, 114]. While these methods can automatically combine atomic features into a compact composite representation, they cannot take into account constraints on feature combination. In contrast, our method can capture features at different composition levels, and more generally can incorporate structural constraints based on prior knowledge. As our experiments show, this approach delivers higher transfer accuracy.

2.3 Hierarchical Low-rank Scoring for Transfer Parsing

2.3.1 Background

Desired Feature Combination We start by briefly reviewing the desired feature combinations that has been shown to play a crucial role in multilingual transfer parsing [97]. Figure 2-1 shows examples of desired feature templates in three major categories: (1) language universal features, (2) selectively shared features and (3) typology group features. The first category aims at capturing language universal syntactic properties such as head-modifier attachment preferences. The second category of templates selectively transfers syntactic properties of specific types of arcs based on typological features. For instance, we design a feature by conjoining the WALS typological feature 87A with the standard directional features that fires only for noun-adjective arcs (as shown in Figure 2-1). This feature will be shared only between languages that have the same noun-adjective ordering preference. Thirdly, the typology group feature templates conjoin standard arc features with all typological features. Thus languages with exactly the same typological properties will share the same set of features.

Tensor Scoring for Parsing Next we briefly introduce the traditional three-way tensor scoring function for parsing [53]. The three-way tensor characterizes each arc $h \rightarrow m$ using the *tensor-product* over three feature vectors: the head vector ($\phi_h \in \mathbb{R}^n$), the modifier vector ($\phi_m \in \mathbb{R}^n$) and the arc vector ($\phi_{h \rightarrow m} \in \mathbb{R}^l$). ϕ_h captures atomic features associated with the head, such as its POS tag and its word form. Similarly, ϕ_m and $\phi_{h \rightarrow m}$ capture atomic features associated with the modifier and the arc respectively. The tensor-product of these three vectors is a rank-1 tensor:

$$\phi_h \otimes \phi_m \otimes \phi_{h \rightarrow m} \in \mathbb{R}^{n \times n \times l}$$

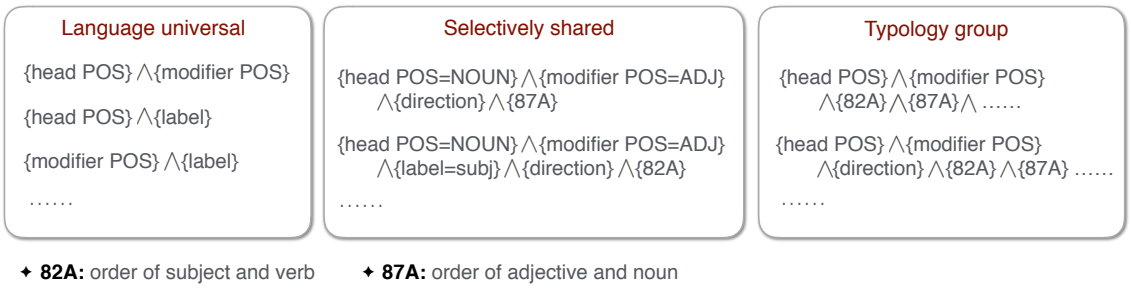

Figure 2-1: Examples of feature templates desired for multilingual transfer parsing. Language universal features capture universal syntactic properties, such as a verb typically taking a noun or an adverb as a dependent. Selectively shared features capture the arc direction preference selectively transfer information based on typological features. Typology group features are shared between languages with exactly the same typological properties.

This rank-1 tensor captures all possible combinations of the atomic features in each vector, and therefore significantly expands the feature set. The tensor score is the inner product between a three-way parameter tensor $A \in \mathbb{R}^{n \times n \times l}$ and this rank-1 feature tensor:

$$vec(A) \cdot vec(\phi_h \otimes \phi_m \otimes \phi_{h \to m})$$

where $vec(\cdot)$ denotes the vector representation of a tensor. This tensor scoring method avoids the parameter explosion and overfitting problem by assuming a low-rank factorization of the parameters A. Specifically, A is decomposed into the sum of r rank-1 components:

$$A = \sum_{i=1}^{r} U(i) \otimes V(i) \otimes W(i)$$

where r is the rank of the tensor, $U, V \in \mathbb{R}^{r \times n}$ and $W \in \mathbb{R}^{r \times l}$ are parameter matrices. $U(i)$ denotes the i-th row of matrix U and similarly for $V(i)$ and $W(i)$. With this factorization, the model effectively alleviates the feature explosion problem by projecting sparse feature vectors into dense r-dimensional embeddings via U, V and W. Subsequently, the score is computed as follows:

$$S_{tensor}(h \to m) = \sum_{i=1}^{r} [U\phi_h]_i [V\phi_m]_i [W\phi_{h \to m}]_i$$

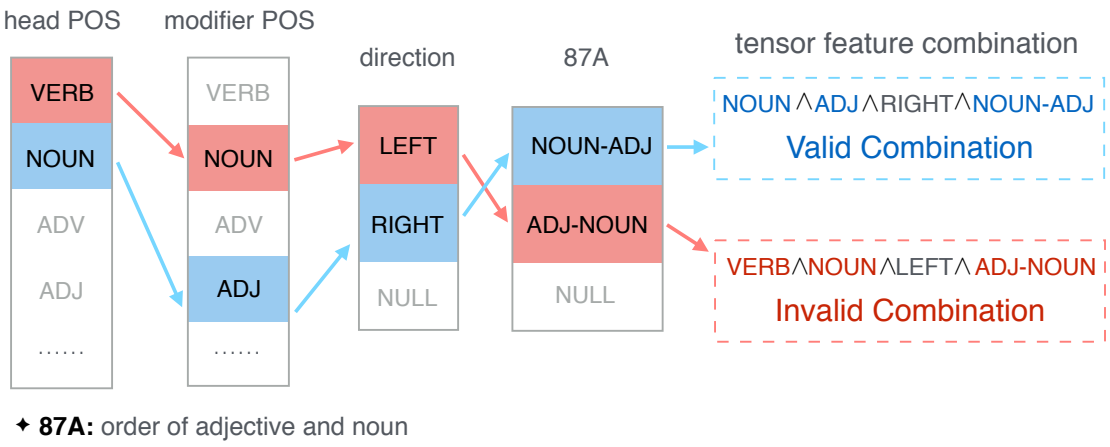

Figure 2-2: An illustration of feature combination by a four-way tensor. The tensor captures all possible concatenations of atomic features over head POS, modifier POS, direction and typology, including combinations that should never trigger.

where $[\cdot]_i$ denotes the i-th element of the matrix.

In multilingual transfer, however, we want to incorporate typological features without introducing feature combinations that are never observed. For example, if we add the noun-adjective ordering preference into $\phi_{h\to m}$, the tensor will represent the concatenation of this preference with a verb-noun arc, even though this feature should never trigger. Figure 2-2 illustrates this issue with a four-way tensor that captures both valid and invalid combinations.

2.3.2 Hierarchical Low-rank Tensor

To address this issue, we propose the hierarchical factorization of tensor parameters.[1] The key idea is to generate intermediate embeddings that capture the interaction of the same set of atomic features as other expert-defined features (e.g. typological features). As a motivating example, consider again the example four-way tensor in Figure 2-2. The typological feature 87A already specifies the head and the modifier POS as noun and adjective. By taking the tensor-product over the typological feature vector and the head POS feature vector, the tensor generates conflict values for

[1]In this section we focus on delexicalized transfer, and describe the lexicalization process in Section 2.3.3.

Notation	Description
H, ϕ_h M, ϕ_m	Head/modifier POS tag
D, ϕ_d	Arc length and direction
L, ϕ_l	Arc label
T_u, ϕ_{t_u}	Typological features that depend on head/modifier POS but not arc label
T_l, ϕ_{t_l}	Typological features that depend on arc label
H_c, ϕ_{h_c} M_c, ϕ_{m_c}	POS tags of head/modifier neighboring words

Table 2.3: Notations and descriptions of parameter matrices and feature vectors in our hierarchical tensor model. $\phi_\cdot$ denote the feature vectors and capital letters denote the corresponding parameter matrices.

head POS. Therefore, we apply an additive operation rather than the tensor-product between the two feature vectors.

We will illustrate this idea in the context of multilingual parsing. Table 2.3 summarizes the notations of the feature vectors and the corresponding parameters. $\phi_h \in \mathbb{R}^{n_h}$ is the feature vector for head POS with size n_h. $H \in \mathbb{R}^{r \times n_h}$ is the parameter matrix for head POS and r is the rank of the tensor. We consider eight different atomic feature vectors in total, ranging from head and modifier POS, arc properties, typological features and neighboring word POS. In general, we use $\phi_\cdot$ to denote the feature vectors and use capital letters to denote the corresponding parameter matrices. Traditional tensor-based models characterizes each arc by directly taking the tensor-product over all eight feature vectors. Figure 2-3 illustrates the visual structure of traditional multiway tensors which assumes no prior knowledge on legitimate feature combinations.

Now, we explain our hierarchical tensor scoring for transfer parsing. Figure 2-4 shows the visual representation of the hierarchical structure. This design enables the model to handle expert-defined features (e.g. typological features) over various

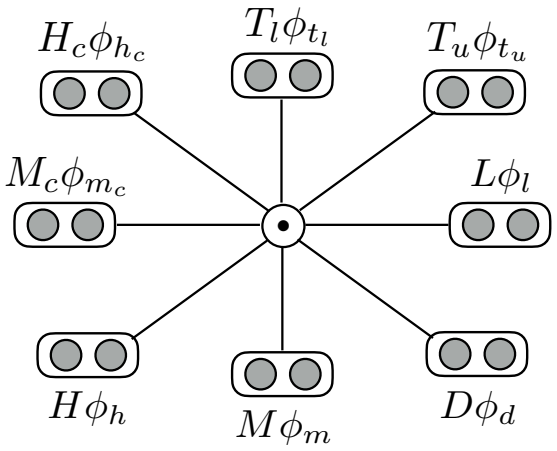

Figure 2-3: Visual representation for traditional multiway tensor. The tensor characterizes each arc using the tensor-product over the eight feature vectors. The arc representation is computed as the element-wise product over the eight r-dimensional feature embeddings pertaining to head POS ($H\phi_h$), modifier POS ($M\phi_m$) etc. The final arc score is computed as the sum of elements in the arc representation.

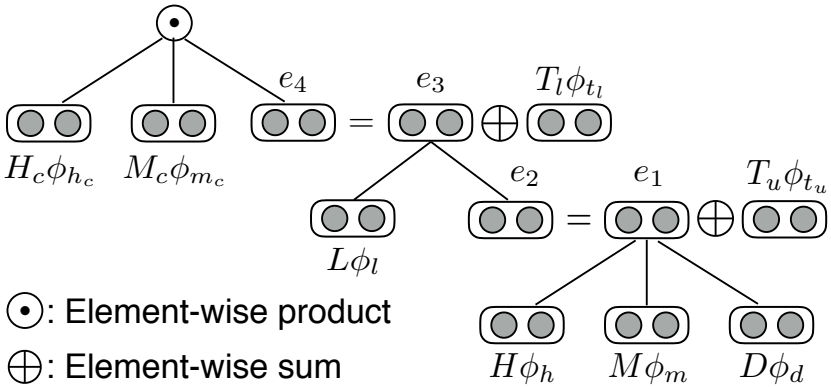

Figure 2-4: Visual representation for hierarchical tensor, represented as a tree structure. The tensor first captures the low-level interaction ($H\phi_h$, $M\phi_m$ and $D\phi_d$) by an element-wise product, and then combines the intermediate embedding with other components higher in the hierarchy, e.g. e_2 and $L\phi_l$. The equations show that we composite two representations by an element-wise sum.

subsets of the atomic features. Specifically, for each arc $h \rightarrow m$ with label l, we first compute the intermediate feature embedding e_1 that captures the interaction between the head ϕ_h, the modifier ϕ_m and the arc direction and length ϕ_d, by an element-wise product.

$$[e_1]_i = [H\phi_h]_i[M\phi_m]_i[D\phi_d]_i \tag{2.1}$$

where $[\cdot]_i$ denotes the i-th value of the feature embedding, and H, M and D are the parameter matrices as in Table 2.3. The embedding e_1 captures the unconstrained interaction over the *head*, the *modifier* and the *arc*. Note that ϕ_{t_u} includes expert-defined typological features that rely on the specific values of the head POS, the modifier POS and the arc direction, such as the example noun-adjective feature in Table 2.2. Therefore, the embedding $T_u\phi_{t_u}$ captures an expert-defined interaction over the *head*, the *modifier* and the *arc*. Thus e_1 and $T_u\phi_{t_u}$ provide two different representations of the same set of atomic features (e.g. the *head*) and our prior knowledge motivates us to exclude the interaction between them. Otherwise, the model will capture invalid conjoining between typological features (e.g. verb-subject) and Thus, we combine e_1 and $T_u\phi_{t_u}$ as e_2 using an element-wise sum

$$[e_2]_i = [e_1]_i + [T_u\phi_{t_u}]_i \tag{2.2}$$

and thereby avoid such combinations. As Figure 2-4 shows, e_2 in turn is used to capture the higher level interaction with arc label features ϕ_l,

$$[e_3]_i = [L\phi_l]_i[e_2]_i \tag{2.3}$$

Now e_3 captures the interaction between head, modifier, arc direction, length and label. It is over the same set of atomic features as the typological features that depend on arc labels ϕ_{t_l}, such as the example verb-subject ordering feature in Table 2.2. Therefore, we sum over these embeddings as

$$[e_4]_i = [e_3]_i + [T_l\phi_{t_l}]_i \tag{2.4}$$

Finally, we capture the interaction between e_4 and context feature embeddings $H_c\phi_{h_c}$ and $M_c\phi_{m_c}$ and compute the tensor score as

$$S_{tensor}(h \xrightarrow{l} m) = \sum_{i=1}^{r} [H_c\phi_{h_c}]_i [M_c\phi_{m_c}][e_4]_i \tag{2.5}$$

By combining Equation 2.1 to 2.5, we observe that our hierarchical tensor score decomposes into three multiway tensor scoring functions.

$$\begin{aligned} S_{tensor}(h \xrightarrow{l} m) = & \sum_{i=1}^{r} [H_c\phi_{h_c}]_i [M_c\phi_{m_c}]_i \\ & \Big\{ [T_l\phi_{t_l}]_i + [L\phi_l]_i \Big([T_u\phi_{t_u}]_i + [H\phi_h]_i [M\phi_m]_i [D\phi_d]_i \Big) \Big\} \\ = & \sum_{i=1}^{r} \Big\{ [H_c\phi_{h_c}]_i [M_c\phi_{m_c}]_i [T_l\phi_{t_l}]_i + [H_c\phi_{h_c}]_i [M_c\phi_{m_c}]_i [L\phi_l]_i [T_u\phi_{t_u}]_i \\ & + [H_c\phi_{h_c}]_i [M_c\phi_{m_c}]_i [L\phi_l]_i [H\phi_h]_i [M\phi_m]_i [D\phi_d]_i \Big\} \end{aligned} \tag{2.6}$$

This decomposition provides another view of our tensor model. That is, our hierarchical tensor is algebraically equivalent to the sum of three multiway tensors, where H_c, M_c and L are shared.[2] From this perspective, we can see that our tensor model effectively captures the following three sets of combinations over atomic features, as shown in Table 2.4. The last set of features f_3 captures the interaction across standard atomic features. The other two sets of features f_1 and f_2 focus on combining atomic typological features with atomic label and context features. Consequently, we explicitly assign zero weights for invalid assignments, by excluding the combination of ϕ_{t_u} with ϕ_h and ϕ_m.

2.3.3 Lexicalization Components

In order to encode lexical information in our tensor-based model, we add two additional components, $H_w\phi_{h_w}$ and $M_w\phi_{m_w}$, for head and modifier lexicalization respectively. We compute the final score as the interaction between the delexicalized feature

[2]We could also associate each multiway tensor with a different weight. In our work, we keep them weighted equally.

• f_1: $\phi_{h_c} \otimes \phi_{m_c} \otimes \phi_{t_l}$
• f_2: $\phi_{h_c} \otimes \phi_{m_c} \otimes \phi_l \otimes \phi_{t_u}$
• f_3: $\phi_{h_c} \otimes \phi_{m_c} \otimes \phi_l \otimes \phi_h \otimes \phi_m \otimes \phi_d$

Table 2.4: Three feature groups captured by our hierarchical tensor model. Descriptions of each feature vector $\phi_\cdot$ are summarized in Table 2.3.

embedding in Equation 2.5 and the lexical components. Specifically:

$$
\begin{aligned}
[e_5]_i &= [H_c\phi_{h_c}]_i[M_c\phi_{m_c}]_i[e_4]_i \\
S_{tensor}(h \xrightarrow{l} m) &= \sum_{i=1}^{r}[H_w\phi_{h_w}]_i[M_w\phi_{m_w}]_i[e_5]_i
\end{aligned}
\tag{2.7}
$$

where e_5 is the embedding that represents the delexicalized transfer results. We describe the features in ϕ_{h_w} and ϕ_{m_w} in Section 2.5.

2.3.4 Combined Scoring

Similar to previous work on low-rank tensor scoring models [53, 54], we combine the traditional scoring and the low-rank tensor scoring. More formally, for a sentence $\mathbf{x}$ and a dependency tree $\mathbf{y}$, our final scoring function has the form

$$
S(\mathbf{x}, \mathbf{y}) = \gamma \sum_{h \xrightarrow{l} m \in \mathbf{y}} \mathbf{w} \cdot \phi(h \xrightarrow{l} m) + (1 - \gamma) \sum_{h \xrightarrow{l} m \in \mathbf{y}} S_{tensor}(h \xrightarrow{l} m) \tag{2.8}
$$

where $\phi(h \xrightarrow{l} m)$ is the traditional features for arc $h \rightarrow m$ with label l and $\mathbf{w}$ is the corresponding parameter vector. $\gamma \in [0, 1]$ is the balancing hyper-parameter and we tune the value on the development set. The parameters in our model are $\theta = (\mathbf{w}, H, M, D, L, T_u, T_l, H_c, M_c)$, and our goal is to optimize all parameters given the training set.

2.4 Learning

In this section, we describe our learning method.[3] Following standard practice, we optimize the parameters $\theta = (\mathbf{w}, H, M, D, L, T_u, T_l, H_c, M_c)$ in a maximum soft-margin framework, using online passive-aggressive (PA) updates [20].

For tensor parameter update, we employ the joint update method originally used by Lei et al. [54] in the context of four-way tensors. While our tensor has a very high order (8 components for the delexicalized parser and 10 for the lexicalized parser) and is hierarchical, the gradient computation is nevertheless similar to that of traditional tensors. As described in Section 2.3.2, we can view our hierarchical tensor as the combination of three multiway tensors with parameter sharing. Therefore, we can compute the gradient of each multiway tensor and take the sum accordingly (see Appendix A.1 for more details). For example, the gradient of the label component is

$$\begin{aligned}\partial L = & \sum_{h \xrightarrow{l} m \in \hat{\mathbf{y}}} \Big((H_c \phi_{h_c}) \odot (M_c \phi_{m_c}) \odot [(T_u \phi_{t_u}) + (H \phi_h) \odot (M \phi_m) \odot (D \phi_d)] \Big) \otimes \phi_l \\ & - \sum_{h \xrightarrow{l} m \in \tilde{\mathbf{y}}} \Big((H_c \phi_{h_c}) \odot (M_c \phi_{m_c}) \odot [(T_u \phi_{t_u}) + (H \phi_h) \odot (M \phi_m) \odot (D \phi_d)] \Big) \otimes \phi_l \end{aligned} \tag{2.9}$$

where $\odot$ is the element-wise product and $+$ denotes the element-wise addition. $\hat{\mathbf{y}}$ and $\tilde{\mathbf{y}}$ are the gold tree and the maximum violated tree respectively. For each sentence $\mathbf{x}$, we find $\tilde{\mathbf{y}}$ via cost-augmented decoding.

Tensor Initialization Given the high tensor order, initialization has a significant impact on the learning quality. We extend the previous power method for high-order tensor initialization [54] to the hierarchical structure using the algebraic view as in computing the gradient.

Figure 2-5 shows the framework of the tensor initialization algorithm. First, note that the traditional manually constructed features $\phi(h \xrightarrow{l} m)$ is an expressive and

[3] Our description focuses on delexicalized transfer, and we can easily extend the method to the lexicalized case.

Input: sparse parameter vector $\mathbf{w}$ from the pre-trained model
tensor rank r
Output: initial tensor parameter matrices $H, M, D, L, T_u, T_l, H_c, M_c$

1: create sparse tensors T_1, T_2, T_3 for feature groups f_1, f_2, f_3 (see Table 2.4) by putting each weight in $\mathbf{w}$ into its corresponding entry in tensors.
2: **for** $i := 1 \dots r$ **do**
3: Randomly initialize unit vectors $h, m, d, l, t_u, t_l, h_c, h_m$
4: $T_1' = T_1 - \sum_{j=1}^{i-1} H_c(j) \otimes M_c(j) \otimes T_l(j)$
5: $T_2' = T_2 - \sum_{j=1}^{i-1} H_c(j) \otimes M_c(j) \otimes L(j) \otimes T_u(j)$
6: $T_3' = T_3 - \sum_{j=1}^{i-1} H_c(j) \otimes M_c(j) \otimes L(j) \otimes H(j) \otimes M(j) \otimes D(j)$
7: **repeat**
8: $h = \langle T_3', h_c, m_c, l, -, m, d\rangle$ and normalize it
9: $m = \langle T_3', h_c, m_c, l, h, -, d\rangle$ and normalize it
10: $d = \langle T_3', h_c, m_c, l, h, m, -\rangle$ and normalize it
11: $t_u = \langle T_2', h_c, m_c, l, -\rangle$ and normalize it
12: $t_l = \langle T_1', h_c, m_c, -\rangle$ and normalize it
13: $l = \langle T_3', h_c, m_c, -, h, m, d\rangle + \langle T_2', h_c, m_c, -, t_u\rangle$ and normalize it
14: $h_c = \langle T_3', -, m_c, l, h, m, d\rangle + \langle T_2', -, m_c, l, t_u\rangle + \langle T_1', -, m_c, t_l\rangle$ and normalize it
15: $m_c = \langle T_3', h_c, -, l, h, m, d\rangle + \langle T_2', h_c, -, l, t_u\rangle + \langle T_1', h_c, -, t_l\rangle$
16: **until** $||m_c||_2$ converges
17: $H(i) = h,\ M(i) = m,\ D(i) = d,\ L(i) = l$
$T_u(i) = t_u,\ T_l(i) = t_l,\ H_c(i) = h_c, M_c(i) = m_c$
18: **end for**
19: **return** $H, M, D, L, T_u, T_l, H_c, M_c$

Figure 2-5: The iterative power method for hierarchical tensor initialization. This method finds a low-rank approximation to the sparse tensors created from the parameter vector $\mathbf{w}$ of a pre-trained model on manual constructed features. $H(i)$ denotes the i-th row of the parameter matrix H. The operator $t_l = \langle T_1', h_c, m_c, -\rangle$ returns a vector in which the k-th element is computed as $\sum_{ij} T_1'(i, j, k) h_c(i) m_c(j)$.

informative subset of the huge feature expansion covered in the tensor. We pre-train our model using only the manual features and then use the corresponding feature parameter vector $\mathbf{w}$ to initialize the tensor. Specifically, we create sparse tensors T_1, T_2, T_3 for feature groups f_1, f_2, f_3 (as shown in Table 2.4) from $\mathbf{w}$ (line 1 in Figure 2-5). We then find a low-rank approximation of the sparse tensors using the power method. Briefly, the power method incrementally computes the most important

rank-1 component for $H(i)$, $M(i)$ etc, for $i = 1 \dots r$. In each iteration, the algorithm updates each component by taking the multiplication between the tensor T and the rest of the components (line 8-15). For instance, when we update the label component l, we do the multiplication for different multiway tensors and then take the sum.

$$l = \langle T'_3, h_c, m_c, -, h, m, d \rangle + \langle T'_2, h_c, m_c, -, t_u \rangle$$

where the operator $\langle T'_2, h_c, m_c, -, t_u \rangle$ returns a vector in which the k-th element is computed as $\sum_{ijl} T'_2(i, j, k, l) h_c(i) m_c(j) t_u(l)$. The algorithm updates other components in a similar fashion until convergence.

2.5 Features

Linear Scoring Features Our traditional linear scoring features in $\phi(h \xrightarrow{l} m)$ are mainly drawn from previous work [97]. Table 2.5 lists the typological features from "The World Atlas of Language Structure (WALS)" [25] used to build the feature templates in our work. We use 82A and 83A for verb-subject and verb-object order respectively because we can distinguish between these two relations based on dependency labels. Table 2.6 summarizes the typological feature templates we use. In addition, we expand features with dependency labels to enable labeled dependency parsing.

Tensor Scoring Features For our tensor model, feature vectors listed in Table 2.3 capture the five types of atomic features as follows:

(a) ϕ_h, ϕ_m: POS tags of the head or the modifier.

(b) ϕ_{h_c}, ϕ_{m_c}: POS tags of the left/right neighboring words.

(c) ϕ_l: dependency labels.

(d) ϕ_d: dependency length conjoined with direction.

(e) ϕ_{t_u}, ϕ_{t_l}: selectively shared typological features, as described in Table 2.6.

We further conjoin atomic features (b) and (d) with the family and the typology class of the language, because the arc direction and the word order distribution depends on the typological property of languages [97]. We also add a bias term into each feature vector.

Partial Lexicalization We utilize multilingual word embeddings to incorporate partial lexical information in our model. We use the CCA method [33] to generate multilingual word embeddings. Specifically, we project word vectors in each non-English language to the English embedding space. To reduce the noise from the automatic projection process, we only incorporate lexical information for the top-100

ID	Feature Description	Possible Values
82A	Order of Subject and Verb	SV, VS, No dominant order
83A	Order of Object and Verb	VO, OV, No dominant order
85A	Order of Adposition and Noun	Prepositions, Postpositions
86A	Order of Genitive and Noun	Genitive-Noun, Noun-Genitive
87A	Order of Adjective and Noun	Adjective-Noun, Noun-adjective

Table 2.5: Typological features from WALS [25] used to build the feature templates in our work, inspired by Naseem et al. [75]. Unlike previous work [75, 97], we use 82A and 83A instead of 81A (order of subject, object and verb) because we can distinguish between subject and object relations based on dependency labels.

Vector	Feature Templates
ϕ_{t_l}	$dir \cdot 82\text{A} \cdot \delta(hp{=}\texttt{VERB} \wedge mp{=}\texttt{NOUN} \wedge \texttt{subj} \in l)$
	$dir \cdot 82\text{A} \cdot \delta(hp{=}\texttt{VERB} \wedge mp{=}\texttt{PRON} \wedge \texttt{subj} \in l)$
	$dir \cdot 83\text{A} \cdot \delta(hp{=}\texttt{VERB} \wedge mp{=}\texttt{NOUN} \wedge \texttt{obj} \in l)$
	$dir \cdot 83\text{A} \cdot \delta(hp{=}\texttt{VERB} \wedge mp{=}\texttt{PRON} \wedge \texttt{obj} \in l)$
ϕ_{t_u}	$dir \cdot 85\text{A} \cdot \delta(hp{=}\texttt{ADP} \wedge mp{=}\texttt{NOUN})$
	$dir \cdot 85\text{A} \cdot \delta(hp{=}\texttt{ADP} \wedge mp{=}\texttt{PRON})$
	$dir \cdot 86\text{A} \cdot \delta(hp{=}\texttt{NOUN} \wedge mp{=}\texttt{NOUN})$
	$dir \cdot 87\text{A} \cdot \delta(hp{=}\texttt{NOUN} \wedge mp{=}\texttt{ADJ})$

Table 2.6: Typological feature templates used in our work. ϕ_{t_l} and ϕ_{t_u} are typological feature vectors as described in Table 2.3. hp/mp are POS tags of the head/modifier. $dir \in \{\texttt{LEFT}, \texttt{RIGHT}\}$ denotes the arc direction. 82A-87A denote the WALS typological feature value. $\delta(\cdot)$ is the indicator function. $\texttt{subj} \in l$ denotes that the arc label l indicates a subject relation, and similarly for $\texttt{obj} \in l$.

most frequent words in the following closed classes: pronoun, determiner, adposition, conjunction, particle and punctuation mark. Therefore, we call this feature extension partial lexicalization.[4]

We follow previous work [53] for adding embedding features. For the linear scoring model, we simply append the head and the modifier word embeddings after the feature vector. For the tensor-based model, we add each entry of the word embedding as a feature value into ϕ_{h_w} and ϕ_{m_w}. In addition, we add indicator features for the English translation of words because this improves performance in preliminary experiments. For example, for the German word *und*, we add the word *and* as a feature.

[4]In our preliminary experiments, we observe that our lexicalized model usually outperforms the unlexicalized counterparts by about 2% (see Section 2.7).

2.6 Experimental Setup

2.6.1 Dataset

Universal Dependency Treebank v2.0 We evaluate our model on the newly released multilingual universal dependency (UD) treebank v2.0 [69] that consists of ten languages: English (EN), French (FR), German (DE), Indonesian (ID), Italian (IT), Japanese (JA), Korean (KO), Brazilian-Portuguese (PT), Spanish (ES) and Swedish (SV). This multilingual treebank is annotated with a universal POS tagset and a universal dependency label set. Therefore, this dataset is an excellent benchmark for cross-lingual transfer evaluation. For POS tags, the gold universal annotation used the coarse tagset [82] that consists of twelve tags: noun, verb, adjective, adverb, pronoun, determiner, adposition, numeral, conjunction, particle, punctuation mark, and a catch-all tag X. For dependency labels, the universal annotation developed the Stanford dependencies [23] into a rich set of 40 labels. This universal annotation enables labeled dependency parsing in cross-lingual transfer.

Universal Dependency Treebank v1.0 We also test on the prior version (v1.0) of the UD treebank in order to make a direct comparison against previous neural network-based multilingual transfer parser [29]. This treebank version consists of nine languages: Czech (CS), Finnish (FI), French (FR), German (DE), Hungarian (HU), Irish (GA), Italian (IT), Spanish (ES), Swedish (SV). It is annotated with the same universal POS tagset and the universal dependency label set as the new version (v2.0) of the UD treebank described above.

2.6.2 Evaluation Scenarios

We first consider the unsupervised transfer scenario, in which we assume no target language annotations are available. Following the standard setup, for each target language evaluated, we train our model on the concatenation of the training data in all other source languages.

In addition, we consider the semi-supervised transfer scenario. On the UD treebank v2.0, we assume 50 sentences in the target language are available with annotation. However, we observe that random sentence selection of the supervised sample results in a big performance variance. Instead, we select sentences that contain patterns that are absent or rare in source language treebanks. To this end, each time we greedily select the sentence that minimizes the KL divergence between the trigram distribution of the target language and the trigram distribution of the training data after adding this sentence. The training data includes both the target and the source languages. The trigrams are based on universal POS tags. Note that our method does not require any dependency annotations. To incorporate the new supervision, we simply add the new sentences into the original training set, weighing their impact by a factor of ten. On the UD treebank v1.0, we follow the setting in previous work [28] and use 3,000 token annotations in the target language.

2.6.3 Baselines

We compare against different variants of our model.

- **Direct**: a direct transfer baseline [68] that uses only delexicalized features in the MSTParser [67]. No typological feature is used in this model.

- **NT-Select**: our model without the tensor component. This baseline corresponds to the prior feature-based transfer method [97] with extensions to labeled parsing, lexicalization and semi-supervised parsing.[5]

- **Multiway**: tensor-based model where typological features are added as an additional component and parameters are factorized in the multiway structure similarly as in Figure 2-3.

- **Sup50**: our model trained only on the 50 sentences in the target language in the semi-supervised scenario.

[5]We use this as a re-implementation of Täckström et al. [97]'s model because their code is not publicly available.

In all the experiments we incorporate partial lexicalization for all variants of our model and we focus on labeled dependency parsing.

2.6.4 Supervised Upper Bound

As a performance upper bound, we train the RBGParser [53], the state-of-the-art tensor-based parser, on the full target language training set. We train the first-order model[6] with default parameter settings, using the current version of the code.[7]

2.6.5 Evaluation Measures

Following standard practices, we report unlabeled attachment score (UAS) and labeled attachment score (LAS), excluding punctuation. For all experiments, we report results on the test set.

2.6.6 Experimental Details

For all experiments, we use the arc-factored model and use Eisner's algorithm [32] to infer the projective Viterbi parse. We train our model and the baselines for 10 epochs. We set a strong regularization $C = 0.001$ during learning because cross-lingual transfer contains noise and the models can easily overfit. Other hyper-parameters are set as $\gamma = 0.3$ and $r = 200$ (rank of the tensor). For partial lexicalization, we set the embedding dimension to 50.

[6] All multilingual transfer models in our work and in Täckström et al. [97]'s work are first-order. Therefore, we train first-order RBGParser for consistency.

[7] `https://github.com/taolei87/RBGParser`

2.7 Results

In this section we present the experimental results on the universal dependency (UD) treebank v2.0 and v1.0, respectively. In addition, we provide analysis on our model properties.

2.7.1 Universal Dependency Treebank v2.0

Table 2.7 and 2.9 summarize the results for the unsupervised and the semi-supervised scenarios on the UD treebank v2.0. Averaged across languages, our model outperforms all the baselines in both cases. Moreover, it achieves best UAS and LAS on 7 out of 10 languages. The difference is more pronounced in the semi-supervised case. Below, we summarize our findings when comparing the model with the baselines.

Impact of Hierarchical Tensors We first analyze the impact of using a hierarchical tensor by comparing against the Multiway baseline that implements traditional tensor model. As Table 2.8 shows, this model learns non-zero weights even for invalid feature combinations.

This disregard to known constraints impacts the resulting performance. In the unsupervised scenario, our hierarchical tensor achieves an average improvement of 0.5% on UAS and 1.3% on LAS. Moreover, our model obtains better UAS on all languages and better LAS on 9 out of 10 languages. This observation shows that the multilingual transfer consistently benefits more from a hierarchical tensor structure. In addition, we observe a similar gain over this baseline in the semi-supervised scenario.

Impact of Tensor Models To evaluate the effectiveness of tensor modeling in multilingual transfer, we compare our model against the NT-Select baseline. In the unsupervised scenario, our tensor model yields a 1.1% gain on UAS and a 1.5% on LAS. In the semi-supervised scenario, the improvement is more pronounced, reaching 1.7% on UAS and 1.9% on LAS. The relative error reduction almost doubles, e.g. 7.1% vs. 3.8% on UAS.

Language	Direct		NT-Select		Multiway		Ours	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
EN	65.7	56.7	67.6	55.3	69.8	56.3	**70.5**	**59.8**
FR	77.9	67.4	**79.1**	**68.9**	78.4	68.3	78.9	68.8
DE	62.1	53.1	62.1	53.3	62.1	54.0	**62.5**	**54.1**
ID	46.8	39.3	57.4	37.1	59.5	38.9	**61.0**	**43.5**
IT	77.9	67.9	**79.4**	**69.4**	79.0	69.0	79.3	**69.4**
JA	57.8	16.8	69.2	20.8	69.9	20.4	**71.7**	**21.3**
KO	59.9	34.3	70.4	29.1	70.5	28.1	**70.7**	**30.5**
PT	77.7	71.0	78.5	72.0	78.3	71.9	**78.6**	**72.5**
ES	76.8	65.9	77.2	67.7	77.6	68.0	**78.0**	**68.3**
SV	**75.9**	**64.5**	74.5	62.2	74.8	62.9	75.0	62.5
AVG	67.8	53.7	71.5	53.6	72.0	53.8	**72.6**	**55.1**

Table 2.7: **Unsupervised on UD v2.0:** Unlabeled attachment scores (UAS) and Labeled attachment scores (LAS) of different variants of our model with partial lexicalization in unsupervised scenario. "Direct" and "Multiway" indicate the direct transfer and the multiway variants of our model. "NT-Select" indicates our model without tensor component, corresponding to a re-implementation of previous transfer model [97] with extensions to partial lexicalization and labeled parsing. The last column shows the results by our hierarchical tensor-based model. Boldface numbers indicate the best UAS or LAS.

Feature	Weight
87A$\wedge hp$=NOUN$\wedge mp$=ADJ	2.24×10^{-3}
87A$\wedge hp$=VERB$\wedge mp$=NOUN	8.88×10^{-4}
87A$\wedge hp$=VERB$\wedge mp$=PRON	1.21×10^{-4}
87A$\wedge hp$=NOUN$\wedge mp$=NOUN	9.48×10^{-4}
87A$\wedge hp$=ADP$\wedge mp$=NOUN	3.87×10^{-4}

Table 2.8: Examples of weights for feature combinations between the typological feature 87A=Adj-Noun and different types of arcs. The first row shows the weight for the valid feature (conjoined with noun→adjective arcs) and the rest show weights for the invalid features (conjoined with other types of arcs).

Language	Direct		NT-Select		Multiway		Ours	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
EN	76.8	70.3	81.0	75.0	81.5	75.9	**82.5**	**77.2**
FR	78.8	70.2	79.4	71.0	79.0	71.1	**79.6**	**71.8**
DE	68.4	59.8	71.3	62.1	72.1	63.2	**74.2**	**65.6**
ID	63.7	56.1	76.9	68.2	77.8	69.3	**79.1**	**70.4**
IT	78.9	70.3	80.2	72.2	80.8	72.6	**80.9**	**72.6**
JA	68.2	42.1	73.0	58.8	75.6	60.9	**76.4**	**61.3**
KO	65.3	45.2	66.5	50.2	67.8	52.8	**70.2**	**54.2**
PT	78.6	72.9	78.7	73.1	**79.3**	**73.9**	**79.3**	73.5
ES	77.0	68.5	77.0	69.0	77.6	69.5	**78.4**	**70.5**
SV	77.7	67.2	77.6	66.8	77.8	67.5	**78.3**	**67.9**
AVG	73.4	62.3	76.2	66.6	76.9	67.7	**77.9**	**68.5**

Table 2.9: **Semi-supervised on UD v2.0:** UAS and LAS of different variants of our model when 50 annotated sentences in the target language are available. Columns have the same meaning as in Table 2.7. Boldface indicate the best UAS or LAS.

While both our model and NT-Select outperform Direct baseline by a large margin on UAS, we observe that NT-Select achieves a slightly worse LAS than Direct. By adding a tensor component, our model outperforms both baselines on LAS, demonstrating that tensor scoring function is able to capture better labeled features for transfer comparing to Direct and NT-Select baselines.

Transfer Performance in the Context of Supervised Results To assess the contribution of multilingual transfer, we compare against the Sup50 results in which we train our model only on 50 target language sentences. As Table 2.10 shows, our model improves UAS by 2.3% and LAS by 2.7%. We also provide a performance upper bound by training RBGParser on the full training set.[8] When trained with partial lexical information as in our model, RBGParser gives 82.9% on UAS and 74.5% on LAS with partial lexical information. By utilizing source language annotations, our model closes the performance gap between training on the 50 sentences and on the full training set by about 30% on both UAS and LAS. We further compare to the

[8]On average, each language has more than 10,000 training sentences.

Language	Semi-supervised with 50 Sentences				Supervised (RBGParser)			
	Sup50		Ours		Partial Lex.		Full Lex.	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
EN	79.6	74.2	**82.5**	**77.2**	88.7	84.5	92.3	90.3
FR	76.9	66.8	**79.6**	**71.8**	83.3	76.5	83.3	76.5
DE	71.0	62.4	**74.2**	**65.6**	82.0	72.8	84.5	78.2
ID	78.2	68.9	**79.1**	**70.4**	85.0	77.1	85.8	79.8
IT	77.1	69.3	**80.9**	**72.6**	85.5	79.8	87.9	84.7
JA	**76.6**	61.0	76.4	**61.3**	79.0	64.0	82.1	70.3
KO	70.1	**54.7**	**70.2**	54.2	74.0	59.1	90.9	86.1
PT	76.0	70.0	**79.3**	**73.5**	85.2	80.8	88.5	86.5
ES	75.2	66.5	**78.4**	**70.5**	82.0	75.0	85.8	81.6
SV	74.9	64.7	**78.3**	**67.9**	84.4	75.4	87.3	82.3
AVG	75.6	65.8	**77.9**	**68.5**	82.9	74.5	87.3	83.5

Table 2.10: **Semi-supervised and Supervised on UD v2.0:** The same semi-supervised setting as in Table 2.9. "Sup50" column shows the results of our model when only supervised data in the target language is available. We also include in the last four columns the supervised training results with partial or full lexicalization as the performance upper bound.

performance upper bound with full lexical information (87.3% UAS and 83.5% LAS). In this case, our model still closes the performance gap by 21% on UAS and 15% on LAS.

2.7.2 Universal Dependency Treebank v1.0

In this subsection, we present the experimental results on the UD treebank v1.0. We focus on the comparison against previous neural network-based multilingual transfer parser [28]. This neural network approach achieves language transfer via parameter sharing and it uses no language typological information for selective transfer. Our model and this baseline are trained on the same amount of annotated data: 3,000 tokens in the target language and all trees in the source languages. Table 2.11 summarizes the labeled attachment scores (LAS) for all the methods. Our model achieves

Language	Semi-supervised with 3,000 Tokens				Supervised (RBGParser)
	Direct	Duong et al.	RBGParser	Ours	
CS	56.2	55.7	59.3	**60.9**	69.9
DE	59.9	61.8	62.4	**65.5**	75.0
ES	68.1	70.5	70.4	**70.7**	79.5
FI	43.3	**51.5**	48.6	47.8	61.6
FR	65.5	67.2	**69.4**	68.1	77.8
GA	58.6	61.1	64.5	**65.5**	71.0
HU	57.9	51.0	59.0	**61.0**	65.7
IT	70.6	71.3	**73.8**	73.7	84.8
SV	56.2	62.5	60.0	**62.1**	73.1
AVG	59.6	61.4	63.0	**63.9**	73.2

Table 2.11: **Semi-supervised and Supervised on UD v1.0:** LAS of different approaches when trained on 3,000 annotated tokens in the target language and all annotations in other source languages. "Direct" is the direct transfer variant of our model and "Duong et al." is the neural network-based transfer model described in [28]. We also include the results of our RBGParser training on the same 3,000 tokens (column 4) or the full training set (column 6). Boldface numbers indicate the best LAS.

the best LAS on six out of nine languages. On average, our model outperforms the neural network baseline [28] by 2.5% (63.9% vs. 61.4). We also train our supervised parser RBGParser on the same 3,000 tokens in the target language. This parser outperforms the neural network baseline by 1.6% even without language transfer. Our model utilizes annotations in source languages in a better way and and further improves the LAS by 0.9% (63.9% vs. 63.0%). We also train the RBGParser on the full training set as a performance upper bound. The last column in Table 2.11 shows the result. Our model closes the performance gap between training on the 3,000 tokens and on the full training set by 9% on LAS.

2.7.3 Model Analysis

Impact of Lexicalization We first demonstrate that adding the lexicalization component consistently improve parsing performance. Table 2-6 shows the averaged LAS

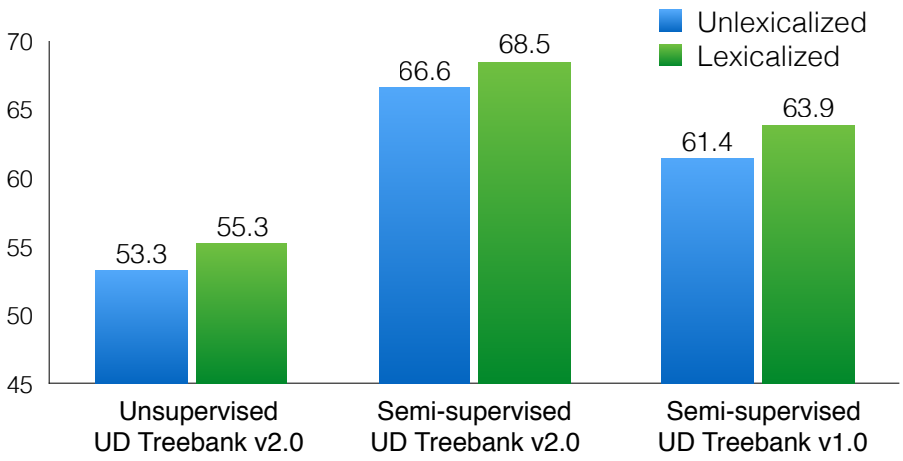

Figure 2-6: Comparisons of averaged LAS between unlexicalized and lexicalized variants of our model.

in the unsupervised and the semi-supervised settings on the universal dependency treebank v2.0 and v1.0. Across all three scenarios, lexicalized models improve over the unlexicalized counterparts by about 2% on LAS.

Time Efficiency of Hierarchical Tensors We observe that our hierarchical structure retains the time efficiency of tensor models. On the English test set, the decoding speed of our hierarchical tensor is close to the multiway counterpart (58.6 vs. 61.2 sentences per second), and is lower than the three-way tensor by a factor of 3.1 (184.4 sentences per second). The time complexity of tensors is linear to the number of low-rank components, and is independent of the factorization structure.

2.8 Conclusions

In this chapter, we introduce a hierarchical tensor based-model which enables us to constrain learned representation based on desired feature interactions. We demonstrate that our model outperforms state-of-the-art multilingual transfer parsers and traditional tensors. These observations, taken together with the fact that hierarchical tensors are efficiently learnable, suggest that the approach can be useful in a broader range of parsing applications; exploring the options is an appealing line of future research.

One limitation of this work is that the model heavily relies on non-lexical features and exclude most lexical information. While we demonstrate that adding partial lexicalization components improves over the unlexicalized counterpart, the practical gains remain limited. In supervised parsing, however, incorporating rich lexical features commonly improve parsing performance by a large margin. Nevertheless, achieving accurate lexical-level transfer is non-trivial and typically requires large amount of parallel resources. In the following chapter of this thesis, we explore algorithms that can effectively learn lexical-level transfer with only a few translation pairs.

Chapter 3

Multilingual POS Tagging via Coarse Mapping between Embeddings

In this chapter, we explore the application of multilingual learning to part-of-speech (POS) tagging for low-resource languages. We demonstrate that effective POS transfer is possible with just ten word translation pairs. Experimental results show that our approach yields a significant improvement over the monolingual prototype-driven method [42] when using a comparable amount of supervision.

3.1 Introduction

After two decades of study, the best performing multilingual methods can in some cases approach their supervised monolingual analogues. To reach this level of performance, however, existing methods crucially depends on the availability of large-scale parallel resources such as parallel translations or bilingual dictionaries. There is not much work on exploring low-resource scenarios where such extensive parallel resources are not available. Indeed, one focus of this chapter is trying to understand how little parallel data is necessary for effective multilingual transfer.

In this chapter, we show that ten translation word pairs are sufficient for effective transfer of multilingual part-of-speech (POS) tagging. To achieve this we make use of and integrate two sources of statistical signal. First, we enable transfer of

information from the source to target languages by establishing a coarse mapping between word embeddings in two languages on the basis of the few available translation pairs. The mapping is useful because of significant structural similarity of embedding spaces across languages. Second, we leverage the potential of unsupervised monolingual models to capture language-specific syntactic properties. The two sources of signals are largely complementary. Embeddings provide a coarse alignment between languages while unsupervised methods fine tune the correspondences in service of the task at hand. While unsupervised methods are fragile and challenging to estimate in general, they can be helpful if initialized and regularized properly, which is our focus.

In order to transfer annotations, we align monolingual embeddings between languages. Clearly, ten translation pairs do not provide sufficient supervision for a full fine-grained alignment because of varied differences across languages. Therefore, we constrain the alignment to be linear and isometric (orthonormal) so as to preserve lengths and angles between word vectors in the embedding space. The main advantage is that this preserves the cosine similarity between vectors, which represents syntactic/semantic similarity between words [70, 80, 44]. The resulting mapping is coarse in the sense that we focus on roughly aligning clusters of words between languages, rather than finding a fine-grained alignment between individual word translations. However, this coarse mapping is still useful for POS tagging because of significant structural similarity of embedding spaces on the POS level. Specifically, we use this coarse alignment to initialize and guide an unsupervised model over the target language.

Our unsupervised model is a feature-based hidden Markov model (HMM) expressed in terms of word embeddings. By establishing a common multilingual embedding space, we can map the source HMM estimated from supervised annotations directly to the target. The resulting "direct transfer" model should be further adjusted as languages differ, and the initial alignment obtained based on embeddings is imperfect. For this reason we cast the direct transfer model as a regularizer for the target HMM, and permit the HMM to further adjust the embedding transformations and relations of embeddings to the tags both globally (overall rotation and scaling)

and locally (introducing small corrections).

Our two phase approach is simple to implement, performs well, and can be adapted to other NLP tasks. We evaluate our approach on POS tagging using the multilingual universal dependency treebanks [77]. We use English as the source language and test on three Indo-European languages (Danish, German and Spanish) and three non-Indo-European-languages (Finnish, Hungarian and Indonesian). Experimental results show that our method consistently outperforms various baselines across languages. On average, our full model achieves 8% absolute improvement over the direct transfer counterpart. We also compare against a prototype-driven tagger [42] using 14 prototypes as supervision. Our model significantly outperforms the model of Haghighi et al. [42] by 37.5% (67.5% vs 30%).

We also introduce a novel task-based evaluation of automatic POS taggers, where tagger predictions are used to determine linguistic typological properties of the target language. In particular, we consider typological features that relates to word ordering preferences as specified in World Atlas of Languages [25]. For example, one task is to predict whether an adjective comes before a noun (as in English) or after a noun (as in Spanish) in the target language. This evaluation highlights key linguistic features of the generated tags. On this task, our model achieves 80% accuracy, yielding 50% error reduction relative to the prototype model.

3.2 Related Work

3.2.1 Multilingual POS Tagging

Our work fits into a broad class of methods for multilingual POS tagging. We first contrast our work to two major categories of approaches for this task, namely tag projection and multilingual word embeddings, followed by discussions on other existing methods.

Tag Projection Much prior work on multilingual POS tagging has focused on the *tag projection* method [113, 106, 26, 27, 96, 21, 92, 73, 13]. The core of this method is to project POS tags from one sentence to its translation in the target language based on word alignment. The projected tag annotations are then used as noisy labeled data to train a tagger for the target language. To automatically induce word alignments, all these approaches have to assume access to a large amount of parallel sentences or bilingual dictionaries. In our work, we focus on a more challenging scenario, in which we do not assume access to parallel sentences. Instead of projecting tag information via word alignment, the transfer in our model is driven by mapping multilingual embedding spaces.

Word Embeddings Our work closely relates to the idea of using multilingual word embeddings for transfer POS tagging. Given this joint word representation, the tagger trained on the source language can be directly applied to the target language. Kim et al. [49] has shown the use of this latent word representation to facilitate multilingual transfer. However, similarly to prior tag projection methods, this representation is learned using parallel data.

Other Methods The feasibility of POS tagging transfer without parallel data has also been shown by Hana et al. [43]. The transfer is performed between languages with similar linguistic typological properties, which enables the model to directly transfer the transition probabilities from source to the target. Moreover, emission

probabilities are hand-engineered to capture language-specific morphological properties. In contrast, our method does not require any language-specific knowledge on the target side.

3.2.2 Multilingual Word Embeddings

There is an expansive body of research on learning multilingual word embeddings [39, 33, 61, 52, 62, 101]. Previous work has shown its effectiveness across a wide range of multilingual transfer tasks including tagging [49], syntactic parsing [108, 41, 30], and machine translation [122, 71]. However, these approaches commonly require parallel sentences or bilingual lexicon to learn multilingual embeddings. Vulic et al. [102] have alleviated the requirements by inducing multilingual word embeddings directly from a document-aligned corpus such as a set of Wikipedia pages on the same theme but in different languages. However, they still used about ten thousands aligned documents as parallel supervision. Our work demonstrates that useful multilingual embeddings can be learned with a minimal amount of parallel supervision.

3.3 Multilingual POS Tagger

Our method is designed to operate in the regime where there are no parallel sentences or target annotations. We assume only a few, in our case ten, word translation pairs. This small number of translation pairs together with the tags that they carry from the source to the target do not provide sufficient information to train a reasonable supervised tagger, even for very close languages where word translations would be mostly one-to-one and tags fully preserved in translation. Other cues are necessary.

The few translation pairs provide just enough information to obtain a coarse global alignment between the source and target language embeddings. We limit the initial linear transformation between embeddings to isometric (orthonormal) mappings so as to preserve norms and angles (e.g., cosine similarities) between words. Once the embeddings are aligned, any source language model expressed in terms of embeddings can be mapped to a target language model. The approach is akin to direct transfer commonly applied in parsing [68, 118] though often with more information. We use the term "direct transfer" to mean the process where no further adjustment is performed beyond the immediate mapping via (coarsely) aligned embeddings.

Direct transfer is insufficient between languages that are syntactically (even moderately) divergent. Instead, we use the directly transferred model to initialize and regularize an unsupervised tagger. Specifically, we employ a feature-based HMM [6] tagger for both the source and target languages with two important modifications. The emission probabilities in the source language HMM are expressed solely in terms of word embeddings (cf. skip-gram models). Such distributions can be directly transferred to the target domain. Our target language HMM is, however, equipped with additional adjustable parameters that can be learned in an unsupervised manner. These include parameters for modifying the initial global linear transformation between embeddings. Beyond this linear transformation, we also add "correction terms" to each tag-word pair that are in principle sufficient to specify any HMM. Both of these additional sets of parameters are regularized towards keeping the initial direct transfer model. As a result, our strongly governed unsupervised tagger can succeed

where an unguided unsupervised tagger would typically fail.

In the remainder of this section, we describe the approach more formally, starting with the coarse alignment between embeddings, followed by the supervised feature-based HMM, and the unsupervised target language HMM.

3.3.1 Isometric Alignment of Word Embeddings

Here we find a linear transformation from the target language embeddings to the source language embeddings using the translation pairs. The resulting transformation permits us to directly apply any source language model on the target language, i.e., it enables direct transfer. To this end, let $\boldsymbol{V}_s \in \mathbb{R}^{n_s \times d}$ and $\boldsymbol{V}_t \in \mathbb{R}^{n_t \times d}$ be the word embeddings estimated for the source and target languages, respectively, with vocabulary sizes n_s and n_t. All the embeddings are of dimension d. The submatrices of embeddings pertaining to k anchor words (from translation pairs) are denoted as $\boldsymbol{\Sigma}_s$ and $\boldsymbol{\Sigma}_t$, where $\boldsymbol{\Sigma}_s, \boldsymbol{\Sigma}_t \in \mathbb{R}^{k \times d}$.

We find a linear transformation $\boldsymbol{P} \in \mathbb{R}^{d \times d}$ that best aligns the embeddings of the translation pairs in the sense of minimizing

$$||\boldsymbol{\Sigma}_t \boldsymbol{P} - \boldsymbol{\Sigma}_s||^2 \tag{3.1}$$

subject to the **isometric** (orthonormal) constraint $\boldsymbol{P}^T \boldsymbol{P} = \boldsymbol{I}$. We use the steepest descent algorithm [1] to solve this optimization problem.[1] Once $\boldsymbol{P}$ is available, we can map all the target language embeddings $\boldsymbol{V}_t$ to the source language space with $\boldsymbol{V}_t \boldsymbol{P}$. Note that since typically in our setting $k < d$ (e.g. $k = 10$) additional constraints such as isometry are required.

Motivation behind the Isometric Constraint We impose isometry on the linear transformation so as to preserve angles and lengths of the word vectors after the transformation. A number of recent studies have explored the use of cosine similarity of word vectors as a measure of semantic relations between words. Thus, for example,

[1] Our implementation is based on the toolkit available at `http://legacy.spa.aalto.fi/sig-legacy/unitary_optimization/`.

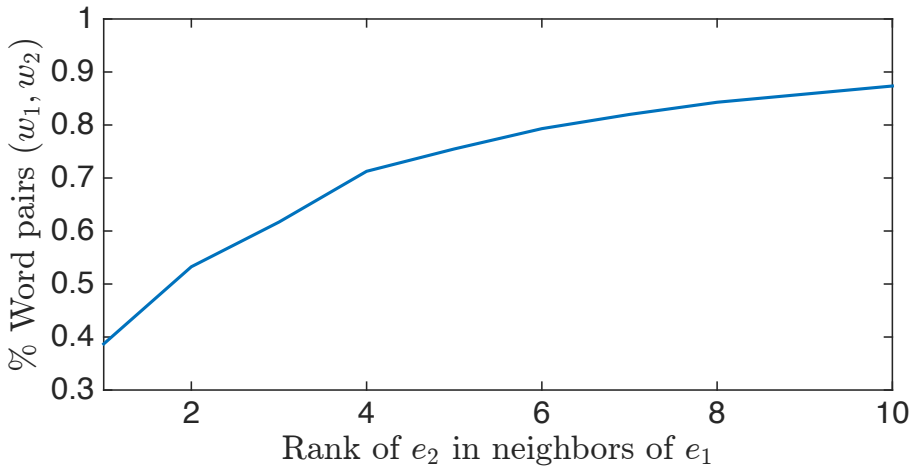

Figure 3-1: Cumulative fraction of word translation pairs among top 1,000 most frequent words where the nearest neighbor of a German word (vector) appears as the r^{th} nearest neighbor after translation, measured in terms of their monolingual word embeddings.

if two words have high cosine similarity in German (target), the corresponding words in English (source) should also be similar. To validate our isometric constraint further, we verify whether nearest neighbors are preserved in monolingual embeddings after translation. To this end, we take the top 1,000 most frequent words in German and their translations into English and ask whether nearest neighbors are preserved if measured in terms of their monolingual embeddings. For each word vector w_1 and its nearest neighbor w_2 in German, let e_1 and e_2 be the corresponding English vectors. We compute the rank of e_2 in the ordered list of nearest neighbors of e_1. As Figure 3-1 shows, in more than 50% of word pairs, e_2 is among the top-2 neighbors of e_1. In over 90% of the word pairs e_2 is among e_1's top-10 closest neighbors.

For the purposes of comparison (see Section 3.5), we introduce also a linear transformation without isometry. In other words, we find $\boldsymbol{P}$ that minimizes $||\boldsymbol{\Sigma}_t\boldsymbol{P} - \boldsymbol{\Sigma}_s||^2$ via the **Moore–Penrose pseudoinverse** [72, 81]. Specifically, let $\boldsymbol{\Sigma}_t^+$ be the pseudoinverse of $\boldsymbol{\Sigma}_t$. Then the solution takes the form $\boldsymbol{P} = \boldsymbol{\Sigma}_t^+\boldsymbol{\Sigma}_s$, and has the minimum Frobenius norm among all possible solutions.

3.3.2 Supervised Source Language HMM

Here we briefly describe how we train a supervised tagger on the source language. The resulting model, together with aligned embeddings, specifies the direct transfer model. It will also be used to initialize and guide the unsupervised tagger on the target language.

Our model has the same structure as the standard HMM but we replace the transition and emission probabilities with log-linear models (cf. feature-based HMM by Berg-Kirkpatrick et al.[6]). The transition probabilities include all indicator features and therefore impose no additional constraints. The emission probabilities, in contrast, are expressed entirely in terms of word embeddings $\boldsymbol{v}_x$ as features. More formally, the emission probability of word x given tag y is given by

$$p_\theta(x|y) \propto \exp\{\, \boldsymbol{v}_x^T \boldsymbol{\mu}_y \,\} \tag{3.2}$$

Note that the parameters $\boldsymbol{\mu}_y$ (one vector per tag) can be viewed as tag embeddings. This supervised tagging model is trained to maximize the joint log-likelihood with l_2-regularization over parameters. We use the L-BFGS [59] algorithm to optimize the parameters.

Once the HMM has been trained, we can specify the direct transfer model. It has the same transition probabilities but the emission probabilities are modified according to $p_\theta^{dt}(x|y) \propto \exp\{\, \boldsymbol{v}_x^T \boldsymbol{P} \boldsymbol{\mu}_y \,\}$ where $\boldsymbol{v}_x$ is now the monolingual target embedding, transformed into the source space via $\boldsymbol{v}_x^T \boldsymbol{P}$. We apply the Viterbi algorithm to predict the most likely POS tag sequence.

3.3.3 Unsupervised Target Language HMM

Our unsupervised HMM for the target language is strictly more expressive than the direct transfer model so as to better tailor it to the target language. Let $\boldsymbol{v}_x$ again be the monolingual target embeddings estimated separately, prior to the HMMs. We map these vectors to the source language embedding space via $\boldsymbol{v}_x^T \boldsymbol{P}$ as discussed earlier, where $\boldsymbol{P}$ is already set and no longer considered a parameter. The form of

the emission probabilities

$$p_\theta^t(x|y) \propto \exp\{\boldsymbol{v}_x^T \boldsymbol{P}\boldsymbol{M}\boldsymbol{\mu}_y + \theta_{x,y}\} \tag{3.3}$$

includes two modifications to the direct transfer model. First, we have introduced an additional global linear transformation $\boldsymbol{M}$ to correct the initial alignment represented by $\boldsymbol{P}$. Second, we include per-symbol parameters $\theta_{x,y}$ which, in principle, are capable of specifying any emission distribution on their own. The adjustable parameters in this model (denoted collectively θ) are $\boldsymbol{M}$, $\{\boldsymbol{\mu}_y\}$, $\{\theta_{x,y}\}$, and the parameters pertaining to the transition probabilities. If we set $\boldsymbol{M} = \boldsymbol{I}$, $\theta_{x,y} = 0$ for all x and y, and borrow $\boldsymbol{\mu}_y$ and the transition parameters from the supervised HMM, then we recover the direct transfer model. Let θ_0 denote this setting of the parameters. In other words, the unsupervised HMM with initial parameters θ_0 is the direct transfer model.

Our approach include initializing $\theta = \theta_0$ and later regularizing θ to remain close to θ_0. The motivation behind this approach is two-fold. First, the initial alignment between embeddings was obtained only on the basis of the few available anchor words and may therefore need to be adjusted. Note that the linear transformation of embeddings now involves scaling and is no longer necessarily isometric. Second, the source and target languages differ and the embeddings are not strictly related to each other via any global linear transformation. We can interpret parameters $\theta_{x,y}$ as local (per word) non-linear deformations of the embedding vectors that specify the emission probabilities. We allow only small non-linear corrections by regularizing $\theta_{x,y}$ to remain close to zero, i.e., the values they have in θ_0.

Our unsupervised HMM is estimated by maximizing the regularized log-likelihood

$$L(\theta) = \sum_{i=1}^{n} \log P_\theta(\mathbf{x}_i) - \frac{\beta}{2}||\theta - \theta_0||_2^2 \tag{3.4}$$

where $\mathbf{x}_i$ is the i^{th} target language sentence, $P_\theta(\mathbf{x}_i)$ is the HMM with parameters θ, and n is the number of sentences in the target text to be annotated. Since all the

parameters in the model are in a log-linear form, we simply use the regularization parameter β. Once estimated, we use the Viterbi algorithm to predict the most likely POS tag sequence.

Estimation Details We maximize $L(\theta)$ using the Expectation-maximization (EM) algorithm. In the E-step, we evaluate expected counts $e_{y',y}$ for tag-tag and $e_{x,y}$ for word-tag pairs, using the forward-backward algorithm. The M-step searches for θ that maximizes

$$l(\theta) = \sum_{y',y} e_{y',y} \log p_\theta^t(y'|y) + \sum_{x,y} e_{x,y} \log p_\theta^t(x|y) - \frac{\beta}{2}||\theta - \theta_0||_2^2 \tag{3.5}$$

The maximization can be be done via L-BFGS which involves computing the gradients of $\log p_\theta^t(y'|y)$ and $\log p_\theta^t(x|y)$ with respect to θ at every iteration. Because the conditional probabilities are expressed in a log-linear form, the gradients take on typical forms such as

$$\frac{dl(\theta)}{d\boldsymbol{\mu}_y} = \sum_x e_{x,y}(\boldsymbol{v}_x^T \boldsymbol{P}\boldsymbol{M} - \sum_{x'} p_\theta^t(x'|y)\boldsymbol{v}_{x'}^T \boldsymbol{P}\boldsymbol{M}) - \beta(\boldsymbol{\mu}_y - \boldsymbol{\mu}_{0y})$$

$$\frac{dl(\theta)}{d\boldsymbol{M}} = \sum_{x,y} e_{x,y}(\boldsymbol{P}^T \boldsymbol{v}_x \boldsymbol{\mu}_y^T - \sum_{x'} p_\theta^t(x'|y)\boldsymbol{P}^T \boldsymbol{v}_{x'} \boldsymbol{\mu}_y^T) - \beta(\boldsymbol{M} - \boldsymbol{I}) \tag{3.6}$$

where $\boldsymbol{\mu}_{0y}$ are initial values for $\boldsymbol{\mu}_y$. See Appendix B.1 for details of parameter updates.

3.4 Experimental Setup

Dataset We evaluate our method on the latest Version 1.2 of the Universal Dependencies Treebanks [77, 69]. We use English as the source language and six other languages as targets. Specifically, we choose three Indo-European languages: Danish (da), German (de), Spanish (es), and three non-Indo-European languages: Finnish (fi), Hungarian (hu), Indonesian (id). All treebanks are annotated with the same universal POS tagset. In our work, we map proper nouns to nouns and map symbol marks[2] and interjections to a catch-all tag X because it is hard and unnecessary to disambiguate them in a low-resource learning scenario. After mapping, our tagset includes the following 14 tags: noun, verb, auxiliary verb, adjective, adverb, pronoun, determiner, adposition, numeral, conjunction, sentence conjunction, particle, punctuation mark, and a catch-all tag X. Note that this universal tagset contains two more tags than the traditional universal tagset proposed by Petrov et al. [82]: auxiliary verb and sentence conjunction. We follow the standard split of the treebanks for every language. For each target language, we use the sentences in the training set as unlabeled data, and evaluate on the testing set.

Word Embeddings To induce monolingual word embeddings, we use the processed Wikipedia text dumps [2] for each language. While Wikipedia texts may contain parallel articles, we show in Table 3.1 that the amount of text varies significantly across languages. Prior work [91] also demonstrated that parallel information in Wikipedia is very noisy. Therefore, direct translations are difficult to get from these texts. We use the `word2vec` tool with the skip-gram learning scheme [70]. In our experiments we use $d = 20$ for the dimension of word embeddings and $w = 1$ for the context window size of the skip-gram, which yields the best overall performance for our model. In our analysis, we also explore the impact of embedding dimension and window size.

[2]Examples of symbol mark include "-", "/" etc.

Language	Tokens (10^6)
English	1,888
Danish	44
German	687
Spanish	399
Finnish	66
Hungarian	89
Indonesian	41

Table 3.1: Number of tokens of the Wikipedia dumps used for inducing word embeddings.

Word Translation Pairs For each target language, we collect English translations for the top ten most frequent words in the training corpus. Our preliminary experiments show that this selection method performs the best. The selected words are typically from closed classes, such as punctuation marks, determiners and prepositions. We find translations using Wiktionary.[3]

Model Variants Our model varies along two dimensions. On one dimension, we use two different methods for inducing multilingual word embeddings: **Pseudoinverse** and **Isometric** alignment as described in Section 3.3.1. On the other dimension, we experiment with two different multilingual transfer models. We use **Direct Transfer** to denote our direct transfer model, and **Transfer+EM** for our unsupervised model trained in the target language.

Baselines We also compare against the prototype-driven method of [42]. Specifically, we use the publicly available implementation provided by the authors.[4] Note that their model requires at least one prototype for each POS category. Therefore, we select 14 prototypes (the most frequent word from each category) for the baseline, while our method only uses ten translation pairs.

[3] `https://www.wiktionary.org/`

[4] `http://code.google.com/p/prototype-sequence-toolkit/`

Evaluation Unlike other unsupervised methods, all models in our experiments can identify the label for each POS tag because of knowledge from either the source languages or prototypes. Therefore, we directly report the token-level POS accuracy for all experiments.

Other Details For all experiments, we use the following regularization weights: $\gamma = 0.001$ for supervised models learned on the source language and $\beta = 0.01$ for unsupervised models learned on the target language. During training, we also normalize the log-likelihood of labeled or unlabeled data by the total number of tokens. As a result, the magnitude of the objective value is independent of the corpus size, hence we do not need to tune the regularization weight for each target language. We run ten iterations of the EM algorithm.

3.5 Results

In this section, we first show the main comparison between the tagging performance of our model and the baselines. In addition, we include an experiment on typology prediction. In Section 3.5.2, we provide a more detailed analysis of model properties.

3.5.1 Main Results

Table 3.2 summarizes the results of the prototype baseline and different variations of our transfer model. Averaged across languages, our model significantly outperforms the prototype baseline by about 37.5% (67.5% vs 30%), demonstrating the effectiveness of multilingual transfer. Moreover, Table 3.2 shows that our full model (Transfer+EM with the isometric alignment mapping) consistently achieves the best performance compared to other model variations. Our model performs better on Indo-European languages than on other languages (72.9% vs. 62.1% on average), because Indo-European languages are linguistically more similar to the source language (English).

Impact of Training in the Target Language We observe that training on unlabeled data in the target language (Transfer+EM model) consistently improves over the direct transfer counterpart. As the bottom part of Table 3.2 shows, running EM on unlabeled data yields an average of 12% absolute gain on Indo-European languages, while on non-Indo-European languages the gain is only 4.4%.

Impact of the Isometric Alignment Constraint As Table 3.2 shows, when we use Transfer+EM models, the isometric alignment method yields a 4.5% improvement over the pseudoinverse method (72.9% vs. 68.4%) on Indo-European languages. However, the improvement margin drops to 0.3% on non-Indo-European languages (62.1% vs. 61.8%). We hypothesis that this discrepancy is due to the difference in the degree of ambiguities of the anchor words across languages. For example, the anchor words of Spanish have an average of 1.5 possible translations to English, while for Indonesian the average ambiguity is 2.7. Therefore, the isometric assumption holds

Method	Indo-European			
	da	de	es	Avg.
Prototype Model	41.3	25.5	28.7	31.8
Pseudoinverse				
Direct Transfer	56.7	49.4	68.4	58.2
Transfer+EM	64.4	65.8	74.9	68.4
Isometric Alignment				
Direct Transfer	59.8	55.4	67.4	60.9
Transfer+EM	**72.5**	**68.7**	**77.5**	**72.9**

Method	Non-Indo-European			
	fi	hu	id	Avg.
Prototype Model	8.2	44.5	30.1	27.6
Pseudoinverse				
Direct Transfer	54.3	60.1	57.7	57.4
Transfer+EM	57.5	**65.3**	62.7	61.8
Isometric Alignment				
Direct Transfer	54.4	61.4	57.2	57.7
Transfer+EM	**58.2**	63.4	**64.8**	**62.1**

Table 3.2: Token-level POS tagging accuracy (%) for different variants of our transfer model. We always use English as the source language. Target languages include Danish (da), German (de), Spanish (es), Finnish (fi), Hungarian (hu) and Indonesian (id). We average the results separately for Indo-European and non-Indo-European languages. The first row shows performance of the prototype-driven baseline [42]. The rest shows results of our model when multilingual embeddings are induced with the pseudoinverse or isometric alignment method. "Direct Transfer" and "Transfer+EM" indicates our direct transfer model and our transfer model trained in the target language respectively.

Feature Description	Possible Values
Order of Subject and Verb	SV, VS, No dominant order
Order of Object and Verb	VO, OV, No dominant order
Order of Adjective and Noun	Adjective-Noun, Noun-adjective
Order of Adposition and Noun	Prepositions, Postpositions
Order of Demonstrative and Noun	Demonstrative-Noun, Noun-Demonstrative

Table 3.3: Linguistic typological features used to evaluate the syntactic quality of automatically generated tags. The goal is to predict word ordering preferences of each language based on POS tag sequences generated by different models.

Tagging Method	**Typology Accuracy**
Prototype	60.0
Direct Transfer	66.7
Transfer + EM	80.0
Gold	93.3

Table 3.4: The accuracy (%) of typological properties prediction using the outputs from different taggers. "Gold" indicates the result using gold POS annotations.

better and the EM algorithm finds a better local optimum for Indo-European languages than for non-Indo-European languages. We also observe a similar pattern in the direct transfer scenario.

Prediction of Linguistic Typology To assess the quality of automatically generated tags, we use them to determine linguistic typological properties of the target language. As shown in Table 3.3, we predict values of the following five linguistic typological properties for each language: subject-verb, verb-object, adjective-noun, adposition-noun and demonstrative-noun. More specifically, the goal is to predict word ordering preferences such as whether an adjective comes before a noun (as in English) or after a noun (as in Spanish). We collect the true ordering preferences

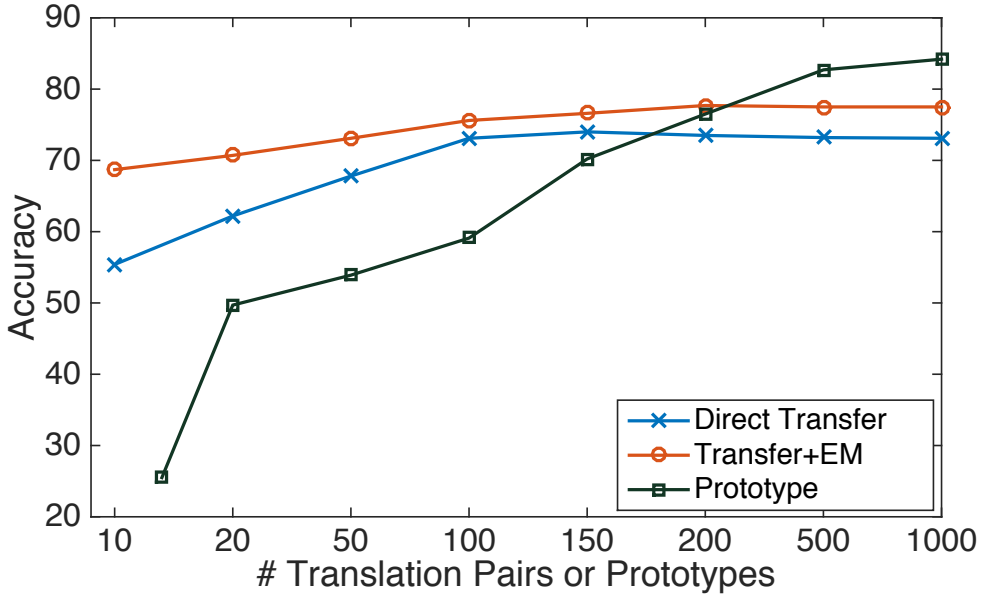

Figure 3-2: Accuracy of our models and the prototype baseline as a function of the amount of supervision, in German. x-axis is the number of translation pairs or prototypes used as supervision. Our models use multilingual embeddings induced with the isometric alignment method. The minimum number of prototypes used by the prototype baseline is 14.

from "The World Atlas of Language Structure (WALS)" [25]. To make predictions, we train a multiclass support vector machine (SVM) classifier [99] on a multilingual corpus using bigrams and trigrams of POS tags as features. The training data for SVM comes from a combination of the Universal Dependencies Treebanks, CoNLL-X, and CoNLL-07 datasets [10, 76], excluding all sentences in the target language. We train one classifier for each typological property, and make predictions for each of the six target languages. For evaluation, we directly report the overall accuracy on all 30 test cases (six languages combined with five typological properties).

Table 3.4 shows the accuracy of predicting typological properties with different tagging models. See Appendix B.2 for detail prediction results for each typological feature and each language. "Gold" corresponds to the result with gold POS annotations and is an upper bound of the prediction accuracy. We observe that the typology

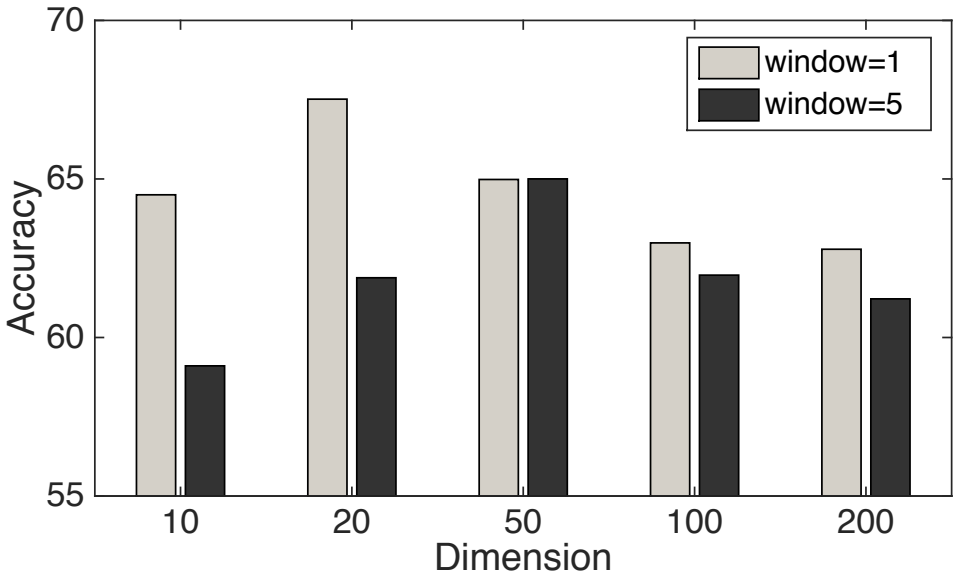

Figure 3-3: The average tagging accuracy (%) with different embedding dimensions and context window sizes. The model is Transfer+EM with the isometric alignment projection method.

prediction accuracy correlates with the tagging quality. With the output of our best model, we predict the correct values for 80% of the typological properties. This corresponds to a 50% error reduction relative to the prototype model.

3.5.2 Analyses

Impact of the Amount of Supervision Figure 3-2 shows the accuracy of the Direct Transfer, Transfer+EM models, and prototype baseline with different amounts of supervision in German. Specifically, the x-axis is the number of translation pairs or prototypes used as supervision. The numbers with ten pairs or prototypes are the same as that in Table 3.2. We automatically extract more translation pairs using the Europarl parallel corpus [51] and select pairs based on the word frequency in the target language. For the prototype model, we select the most frequent words as prototypes based on annotations in the training data, and guarantee that each POS

Model	da	de	es	fi	hu	id	Average
All features	72.5	68.7	77.5	58.2	63.4	64.8	67.5
- Indicator features	70.8	64.8	73.9	53.7	62.9	56.8	63.8
- Transformation matrix M	60.2	65.6	73.2	58.6	59.6	70.8	64.7

Table 3.5: The accuracy (%) of our best Transfer+EM model with different feature sets, removing either indicator features or transformation matrix M at a time.

category has at least one prototype. Note that the minimum number of prototypes used by the prototype model is 14.

One particularly interesting observation is that our model with ten pairs achieves an equivalent performance as that of the prototype-driven method with 150 prototypes. Multilingual transfer compensates for 15 times the amount of supervision. We also observe that the prototype-driven model outperforms our model when large amount of annotations are available. This can be explained by noise in the translation and the limitation from the linear embedding mapping process, which makes POS tags not preserve well across languages.

When comparing between our models, Figure 3-2 shows that Transfer+EM consistently improves over the Direct Transfer, while the gains are more profound in the low-supervision scenario. This is not surprising because with more translation pairs, we are able to induce higher quality multilingual embeddings, which is more beneficial to the direct transfer model.

Impact of Embedding Dimensions and Window Size Figure 3-3 shows the average accuracy across six target languages with different embedding dimensions and context window sizes. First, we observe that a small window size $w = 1$ consistently outperforms window size $w = 5$, demonstrating that smaller window sizes appear to produce word embeddings better for POS tagging. This observation is in line with the finding by Lin et al. [58]. Moreover, we obtain the best performance with dimension $d = 20$ when $w = 1$. On one hand, embeddings with smaller dimension (e.g. $d = 10$)

have too little syntactic information for good POS tagging. On the other hand, if the embedding space has larger dimension, the space will be more complex and mapping embedding spaces will be more difficult given only ten translation pairs. Therefore, we observe a performance drop with either smaller or larger dimensions.

Ablation Analysis on Features In our Transfer+EM model, we add indicator features and transformation matrix M to enhance the emission distribution (see Section 3.3.3). To analyze their contribution, we remove these features in turn and report the results in Table 3.5. Averaged over all languages, adding indicator features improves the accuracy by 3.7%, and adding a transformation matrix increases the accuracy by 2.8%.

3.6 Conclusions

In this chapter, we demonstrate that ten translation pairs suffice for an effective multilingual transfer of POS tagging. Experimental results show that our model significantly outperforms the direct transfer method and the prototype baseline. The effectiveness of our approach suggests its potential application to a broader range of NLP tasks that require word-level multilingual transfer, such as multilingual parsing and machine translation.

In this work, the resulting mapping between monolingual embeddings is coarse in the sense that we only focus on roughly aligning word clusters. While this coarse mapping suffice for multilingual transfer on the POS level, it is still far away from a fine-grained alignment between individual word translations. Some natural directions of future research include (1) studying how little parallel resources are necessary for learning a fine-grained multilingual word embeddings (2) exploring new algorithms that reduce the amount of required parallel resources. We believe in the future our work will have contributions on learning high-quality multilingual word embeddings with low parallel resources.

Chapter 4

Aspect-augmented Adversarial Networks for Domain Adaptation

In this chapter, we introduce a neural method for transfer learning between two (source and target) classification tasks or aspects over the same domain. Instead of target labels, we assume a few keywords pertaining to source and target aspects indicating sentence relevance rather than document class labels. Documents are encoded by learning to embed and softly select relevant sentences in an aspect-dependent manner. A shared classifier is trained on the source encoded documents and labels, and applied to target encoded documents. We ensure transfer through aspect-adversarial training so that encoded documents are, as sets, aspect-invariant. Experimental results demonstrate that our approach outperforms different baselines and model variants on two datasets, yielding an improvement of 24% on a pathology dataset and 5% on a review dataset.

4.1 Introduction

Deep learning methods are highly effective when they can be trained with large amounts of labeled training data in the domain of interest. While such data are not always available in real applications, it is nevertheless often possible to find labeled data in another related domain or for another related task. Considerable ef-

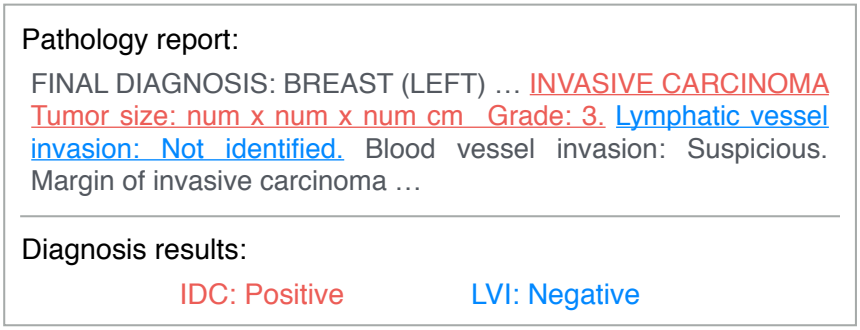

Figure 4-1: A snippet of a breast pathology report with diagnosis results for two types of disease. Evidence for both results is in red and blue, respectively.

fort has gone into designing domain transfer algorithms that leverage such related data [37, 15, 121]. In a typical case, the related domain involves the same classification task (e.g., sentiment analysis) but over different types of examples (e.g., hotel vs restaurant reviews). Labeled training data are available only in the source domain (e.g., hotel reviews) while the task is to provide an effective method for the target domain (e.g., restaurant reviews) without any additional labeled examples.

In this chapter we are primarily interested in transfer between two classification tasks over the same domain, i.e., over the same set of examples. We call this "aspect transfer" as the two classification tasks can be thought to pertain to different aspects of the same examples. For example, the target goal may be to classify pathology reports (shown in Figure 4-1) for the presence of lymph invasion but the available training data involve only annotations for carcinoma in the same reports. Existing domain adaptation methods do not directly solve this aspect transfer problem because input examples are the same across the two tasks. Since there are no labels available for the target aspect, we must learn to properly relate the two tasks. In particular, we bring in auxiliary data to help connect the tasks.

Our approach builds on relevance annotations of sentences which are considerably easier to obtain than actual class labels. Relevance merely indicates a possibility that the answer could be found in a sentence, not what the answer is. One can often write simple keyword rules that identify sentence relevance to a particular aspect

(task) through representative terms, e.g., specific hormonal markers in the context of pathology reports. We can also use keywords of other irrelevant aspects to indicate absence of relevance. Annotations of this kind can be readily provided by domain experts, or extracted from medical literature such as codex rules in pathology [79]. We therefore assume a small number of relevance annotations pertaining to both source and target aspects as a form of weak supervision. These annotations permit us to learn how to encode the examples (e.g., pathology reports) from the point of view of the desired task. Specifically, differential encodings of the same report in our approach arise from softly selecting aspect-relevant sentences from the report.

Our relevance driven encoding returns the aspect-transfer problem closer to the realm of standard domain adaption. We employ a shared end classifier between the tasks but it is exercised differently due to aspect-driven encoding of examples. The two domains as in standard domain adaption are therefore induced by different ways of interpreting the same example in our case. These interpretations are themselves learned based on relevance feedback, thus naturally pulled apart. To ensure that the classifier can be adjusted only based on the source class labels and still reasonably applied to the target encodings, we must align the two sets of encoded examples. As in prior domain adaptation work [22, 7], we assume that the two sets are already partially aligned via common features. For example, the word "presence" commonly exists and indicates positive labels in both cases. The primary goal of transfer is to align the rest features that appear in only one domain or aspect. Also, note that this alignment or invariance is enforced on the level of sets, not individual examples or reports; encoding of any specific report should remain substantially different for label prediction. To learn the invariance, we introduce an adversarial domain classifier analogously to recent successful use of adversarial training in computer vision [35]. The role of the adversarial domain classifier is to learn to distinguish between the two types of encodings, establishing invariance (as sets) when it fails. All the three components in our approach, 1) aspect-driven encoding, 2) classification of source labels, and 3) domain adversary, are trained jointly (concurrently) to complement and balance each other.

Adversarial training of domain and end classifiers can be challenging to stabilize. In our setting, sentences are encoded with a shared convolutional model, weighted by predicted aspect relevance, and then combined into aspect-driven document representations. Feedback from adversarial training can be an unstable guide for how the sentences should be encoded in the first place. To this end, we incorporate an additional word-level autoencoder reconstruction loss to ground the convolutional processing of sentences. We empirically demonstrate that this additional objective yields richer and more diversified feature representations, improving transfer.

We evaluate our approach on pathology reports (aspect transfer) as well as on a more standard review dataset (domain adaptation). On the pathology dataset, we explore cross-aspect transfer across different types of breast disease. Specifically, we test on six adaptation tasks, consistently outperforming all other baselines. Overall, our full model achieves 24% and 12.8% absolute improvement arising from aspect-driven encoding and adversarial training, respectively. Moreover, our unsupervised adaptation method is only 2.8% behind the accuracy of a supervised target model. On the review dataset, we test adaptation from hotel to restaurant reviews. Our model outperforms the marginalized denoising autoencoder [15] by 5%. Finally, we examine and illustrate the impact of individual components on the resulting performance.

4.2 Related Work

Domain Adaptation for Deep Learning Existing approaches commonly induce abstract representations without pulling apart different aspects in the same example, and therefore are likely to fail on the aspect transfer problem. The majority of these prior methods propose to first learn a task-independent representation, and then train a label predictor (e.g. SVM) on this representation in a separate step. For example, earlier researches employ a shared autoencoder [37, 17] or a deep convolutional neural network [24] to learn cross-domain representation. Chen et al. [15] further improve and stabilize the representation learning by utilizing marginalized denoising autoencoders. Later, Zhou et al. [121] propose to minimize domain-shift of the autoencoder in a linear data combination manner. Some other work has focused on learning transferable representations in an end-to-end fashion. Examples include using transduction learning for object recognition [88] and using residual transfer networks for image classification [60]. In contrast, we use adversarial training to encourage learning domain-invariant features in a more explicit way. Our approach offers another two advantages over prior work. First, we jointly optimize features with the final classification task while much previous work only learns task-independent features using autoencoders. Second, our model can handle traditional domain transfer as well as aspect transfer, while previous methods can only handle the former scenario.

Adversarial Learning in Vision and NLP Our approach closely relates to the idea of domain-adversarial training. Adversarial networks and similar approaches have originally been developed for image generation [38, 63, 94, 85, 98, 57], and later applied to domain adaption in computer vision [35, 36, 8, 100] and speech recognition [89]. The core idea of these approaches is to promote the emergence of invariant image features by optimizing the feature extractor as an adversary against the domain classifier. While Ganin et al. [36] also apply this idea to sentiment analysis, their practical gains have remained limited.

Our approach presents two main departures. In computer vision, adversarial learning has been used for transferring across domains, while our method can also handle

aspect transfer. In addition, we introduce reconstruction loss which results in more robust adversarial training. We believe that this formulation will benefit other applications of adversarial training, beyond the ones described in this chapter.

Semi-supervised Learning with Keywords In our work, we use a small set of keywords as a source of weak supervision for aspect-relevance scoring. This relates to prior work on utilizing prototypes and seed words in semi-supervised learning [42, 40, 12, 64, 48, 56, 31]. All these prior approaches utilize prototype annotations primarily targeting for model bootstrapping but not for learning representations. In contrast, our model uses provided keywords to learn aspect-driven encoding of input examples.

Attention Mechanism in NLP One may view our aspect-relevance scorer as a sentence-level "semi-supervised attention", where relevant sentences receive more attention during feature extraction. While traditional attention-based models typically induce attention in an unsupervised manner, they have to rely on a large amount of labeled data for the target task [4, 87, 14, 16, 110, 109, 112, 66, 55]. Unlike them, we assume no label annotations in the target domain. Some other researches have focused on utilizing human-provided rationales as "supervised attention" to improve prediction [117, 65, 119, 9]. In contrast, our model only assumes access to a small set of keywords as a source of weak supervision. Moreover, all these prior approaches focus on in-domain classification. In this chapter, however, we study the task in the context of domain adaptation.

4.3 Methods

We formalize here the aspect transfer problem between the source and target classification tasks over the same set of examples (here documents, e.g., pathology reports). Class labels are available only for the source task, and the goal is to solve the target classification task. While we develop our method under the assumption that the examples in the two tasks are the same (as an extreme case), this is not a requirement for our method and it will work fine in more traditional domain adaptation settings as well, which we demonstrate.

Let $\mathbf{d} = \{\mathbf{s}_i\}_{i=1}^{|\mathbf{d}|}$ be a document that consists of a sequence of $|\mathbf{d}|$ sentences. Each sentence is a sequence of words, namely $\mathbf{s}_i = \{\mathbf{x}_{i,j}\}_{j=1}^{|\mathbf{s}_i|}$, where $\mathbf{x}_{i,j} \in \mathbb{R}^d$ denotes the vector representation of the j-th word in the i-th sentence. Given a document $\mathbf{d}$ we wish to predict the corresponding class label y (e.g., $y \in \{-1, 1\}$) which varies for the same document depending on which aspect (source, target) we are interested in. We assume that the set of possible labels are the same across tasks. Moreover, as in standard domain adaptation, we assume that for particular keywords $\mathbf{x}$ that exist in both tasks (e.g. the word "presence"), their underlying label distributions $p(y|\mathbf{x})$ are the same (e.g. "presence" always indicates positive labels). We use $y^s_{l;k}$ to denote the k-th coordinate of a one-hot vector indicating the correct source label for document $\mathbf{d}_l$.

Beyond labeled documents for the source task $\{\mathbf{d}_l, y^s_l\}_{l\in L}$, and shared unlabeled documents for source and target tasks $\{\mathbf{d}_l\}_{l\in U}$, we assume further that we have relevance scores pertaining to each aspect. The relevance is given per sentence, for some subset of sentences across the documents, and indicates the possibility that the answer for that document would be found in the sentence but without indicating which way the answer goes. Relevance is always task (aspect) dependent yet often easy to provide with simple keyword rules. We use $r^a_i \in \{0, 1\}$ to denote the given relevance label pertaining to aspect a for sentence $\mathbf{s}_i$. Specifically, if sentence $\mathbf{s}_i$ has a relevance label, then $r^a_i = 1$ when the sentence contains any keywords pertaining to aspect a and $r^a_i = 0$ if it has any keywords of other aspects. Separate subsets of relevance

labels are available for each task as the keywords differ. Let $R = \{(a, l, i)\}$ denote the index set of relevance labels such that if $(a, l, i) \in R$ then relevance label $r^a_{l,i}$ is available for aspect a and the i^{th} sentence in document $\mathbf{d}_l$.

4.3.1 Our Approach

We commence with a summary of our approach and describe each part in more technical detail in following subsections. Figure 4-2 outlines the overall model. Figure 4-3 depicts details of the aspect-driven document encoding process. Each sentence is first encoded into a vector using a shared convolutional model. We ground this convolutional model by including a reconstruction step for each word based on the internal state centered at the same position. The sentence vectors are then passed on to a single hidden layer network, a separate network for each aspect with a shared hidden layer, to determine whether the sentences are relevant for the chosen aspect. Our relevance predictors are non-negative regression methods as relevance varies more on a linear rather than binary scale. The predicted relevance scores are used to construct document vectors by taking relevance-weighted combinations of the associated sentence vectors. Thus the document vector is always aspect-dependent due to the chosen relevance weights. The constrained manner in which these document vectors arise from sentence vectors means that they will retain explicit information about the aspect they were based on. Such explicit cues are not helpful in our setting: the end classifier, trained only on source labels, would unnecessarily rely on cues present only in source-aspect encodings. To remove those cues, we introduce an additional linear transformation layer after the initial document encoding.

During training, the resulting adjusted document vectors are used by two classifiers, each involving one hidden layer. The primary end classifier aims to predict the source labels (when available), while the domain classifier determines whether the document vector pertains to the source or target aspect (i.e., label that we know by construction). The two classifiers involve separate training losses that interact only in terms of the document representation. Specifically, the training signal from the primary classifier is used to co-operatively adjust the document representation

whereas the gradient from the domain classifier (the adversary) is reversed therefore encouraging representations that make it fail.

The four training losses pertaining to word reconstruction, relevance labels, source class labels, and domain labels are used concurrently in our adversarial training scheme to adjust the model parameters. At the conclusion of training, we expect that the primary classifier is able to predict the source labels while appearing to be exercised in a domain invariant manner, enabling transfer to the target task.

4.3.2 Components in detail

We provide here additional details of each of the components in the model, including how they are trained as part of the overall approach.

Sentence embedding We apply a convolutional model illustrated in Figure 4-4 to each sentence $\mathbf{s}_i$ to obtain sentence-level vector embeddings $\mathbf{x}_i^{sen}$. The use of RNNs or bi-LSTMs would result in more flexible sentence embeddings but based on our initial experiments, we did not observe any significant gains over the simpler CNNs.

With adversarial training, we observe that the document representation $\mathbf{x}^{doc}$ always has non-zero values only on a small *fixed* set of dimensions, while all other dimensions have zero values. This feature distribution is trivially domain-invariant, but it eliminates too much information for label predictions. To address this issue, we introduce an additional word-level reconstruction step in the convolutional model to further ground the resulting sentence embeddings. The purpose of this reconstruction step is to balance adversarial training signals propagating back from the domain classifier. Specifically, it forces the sentence encoder to keep rich word-level information in contrast to adversarial training that seeks to eliminate task/aspect specific features. We provide an empirical analysis of the impact of this reconstruction in the experiment section (Section 4.6).

More concretely, we reconstruct word embedding from the corresponding convolutional layer, as shown in Figure 4-4. Let $\mathbf{h}_{i,j}$ be the convolutional output when $\mathbf{x}_{i,j}$

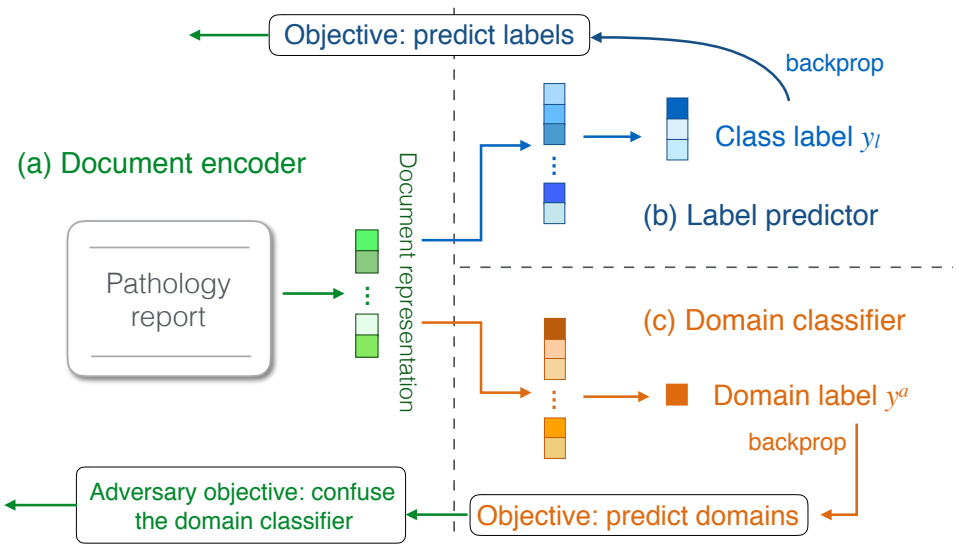

Figure 4-2: Aspect-augmented adversarial network for domain adaptation. The model is composed of (a) an aspect-driven document encoder, (b) a label predictor and (c) a domain classifier. Parameters of all the components are learned jointly during training.

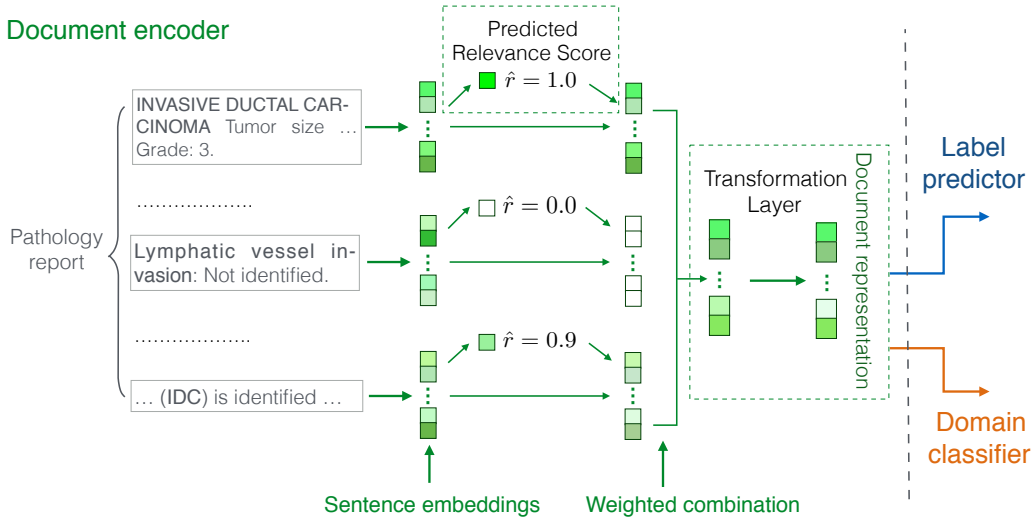

Figure 4-3: Document encoder of our aspect-augmented adversarial network. Each document is encoded in a relevance weighted, aspect-dependent manner and passed on to both the primary label classifier and the domain classifier as the adversary to ensure invariance.

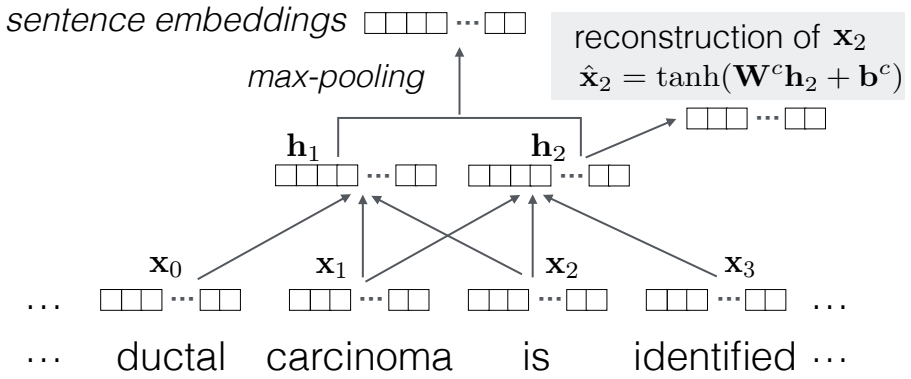

Figure 4-4: Illustration of the convolutional model and the reconstruction of word embeddings from the associated convolutional layer.

is at the center of the window. We reconstruct $\mathbf{x}_{i,j}$ by

$$\hat{\mathbf{x}}_{i,j} = \tanh(\mathbf{W}^c\mathbf{h}_{i,j} + \mathbf{b}^c) \tag{4.1}$$

where $\mathbf{W}^c$ and $\mathbf{b}^c$ are parameters of the reconstruction layer. The loss associated with the reconstruction for document $\mathbf{d}$ is

$$\mathcal{L}^{rec}(\mathbf{d}) = \frac{1}{n}\sum_{i,j} ||\hat{\mathbf{x}}_{i,j} - \tanh(\mathbf{x}_{i,j})||_2^2 \tag{4.2}$$

where n is the number of tokens in the document and indexes i, j identify the sentence and word, respectively. The overall loss $\mathcal{L}^{rec}$ is obtained by summing over all labeled/unlabeled documents.

Relevance prediction We use a small set of keyword rules to generate binary relevance labels, both positive ($r = 1$) and negative relevance ($r = 0$). These labels represent the only supervision available to predict relevance. The prediction is made on the basis of the sentence vector $\mathbf{x}_i^{sen}$ passed through a feed-forward network with a ReLU output unit. The network has a single shared hidden layer and a separate output layer for each aspect. Note that our relevance prediction network is trained as a regression model even though the available labels are binary.

Given relevance labels indexed by $R = \{(a, l, i)\}$, we minimize

$$\mathcal{L}^{rel} = \sum_{(a,l,i)\in R} \left(r^a_{l,i} - \hat{r}^a_{l,i}\right)^2 \tag{4.3}$$

where $\hat{r}^a_{l,i}$ is the predicted (non-negative) relevance score pertaining to aspect a for the i^{th} sentence in document $\mathbf{d}_l$, as shown in Figure 4-3. $r^a_{l,i}$, defined earlier, is the given binary (0/1) relevance label.

Document encoding The initial vector representation for each document such as $\mathbf{d}_l$ is obtained as a relevance weighted combination of the associated sentence vectors, i.e.,

$$\mathbf{x}^{doc,a}_l = \frac{\sum_i \hat{r}^a_{l,i} \cdot \mathbf{x}^{sen}_{l,i}}{\sum_i \hat{r}^a_{l,i}} \tag{4.4}$$

The resulting vector selectively encodes information from the sentences based on relevance to the focal aspect.

Transformation layer We add a transformation layer to help map the initial document vectors $\mathbf{x}^{doc,a}_l$ to their domain invariant (as a set) versions. Specifically, the transformed representation is given by $\mathbf{x}^{tr,a}_l = \mathbf{W}^{tr}\mathbf{x}^{doc,a}_l$. The transformation has to be strongly regularized lest the gradient from the adversary would wipe out all the document signal. We add the following regularization term

$$\Omega^{tr} = \lambda^{tr}||\mathbf{W}^{tr} - \mathbf{I}||^2_F \tag{4.5}$$

to discourage significant deviation away from identity $\mathbf{I}$. λ^{tr} is a regularization parameter that has to be set separately based on validation performance. We show an empirical analysis of the impact of this transformation layer in Section 4.6.

Primary label classifier As shown in the top-right part of Figure 4-2, the classifier takes in the adjusted document representation as an input and predicts a probability distribution over the possible class labels. The classifier is a feed-forward network with a single hidden layer using ReLU activations and a softmax output layer over

the possible class labels. Note that the classifier operates the same regardless of the aspect relative to which the document was encoded. It must therefore be co-operatively learned together with the encodings.

Let $\hat{p}_{l;k}$ denote the predicted probability of class k for document $\mathbf{d}_l$ when the document is encoded from the point of view of the source aspect. Recall that $[y^s_{l;1}, \ldots, y^s_{l;m}]$ is a one-hot vector for the correct (given) source class label for document $\mathbf{d}_l$, hence also a distribution. We use the cross-entropy loss for the label classifier

$$\mathcal{L}^{lab} = \sum_{l \in L} \left[-\sum_{k=1}^{m} y^s_{l;k} \log \hat{p}_{l;k} \right] \tag{4.6}$$

Domain classifier As shown in the bottom-right part of Figure 4-2, the domain classifier functions as an adversary to ensure that the documents encoded with respect to the source and target aspects look the same as sets of examples. The invariance is achieved when the domain classifier (as the adversary) fails to distinguish between the two. Structurally, the domain classifier is a feed-forward network with a single ReLU hidden layer and a softmax output layer over the two aspect labels.

Let $y^a = [y^a_1, y^a_2]$ denote the one-hot domain label vector for aspect $a \in \{s, t\}$. In other words, $y^s = [1, 0]$ and $y^t = [0, 1]$. We use $\hat{q}_k(\mathbf{x}^{tr,a}_l)$ as the predicted probability that the domain label is k when the domain classifier receives $\mathbf{x}^{tr,a}_l$ as the input. The domain classifier is trained to minimize

$$\mathcal{L}^{dom} = \sum_{l \in L \cup U} \sum_{a \in \{s,t\}} \left[-\sum_{k=1}^{2} y^a_k \log \hat{q}_k(\mathbf{x}^{tr,a}_l) \right] \tag{4.7}$$

4.3.3 Joint learning

We combine the individual component losses into an overall objective function

$$\mathcal{L}^{all} = \mathcal{L}^{rec} + \mathcal{L}^{rel} + \Omega^{tr} + \mathcal{L}^{lab} - \rho \mathcal{L}^{dom} \tag{4.8}$$

which is minimized with respect to the model parameters except for the adversary (domain classifier). The adversary is maximizing the same objective with respect to

its own parameters. The last term $-\rho\mathcal{L}^{dom}$ corresponds to the objective of failing the domain classifier. The proportionality constant ρ controls the impact of gradients from the adversary on the document representation; the adversary itself is always directly minimizing $\mathcal{L}^{dom}$.

All the parameters are optimized jointly using standard backpropagation (concurrent for the adversary). Each mini-batch is balanced by aspect, half coming from the source, the other half from the target. All the loss functions except $\mathcal{L}^{lab}$ make use of both labeled and unlabeled documents. It would be straightforward to add a loss term also for target labels if they are available.

4.4 Experimental Setup

Pathology dataset This dataset contains 96.6k breast pathology reports collected from three hospitals [111]. A portion of this dataset is manually annotated with 20 categorical values, representing various aspects of breast disease. In our experiments, we focus on four aspects related to carcinomas and atypias: Ductal Carcinoma In-Situ (DCIS), Lobular Carcinoma In-Situ (LCIS), Invasive Ductal Carcinoma (IDC) and Atypical Lobular Hyperplasia (ALH). Each aspect is annotated using binary labels. We use 500 held out reports as our test set and use the rest labeled data as our training set: 23.8k reports for DCIS, 10.7k for LCIS, 22.9k for IDC, and 9.2k for ALH. Table 4.1 summarizes statistics of the dataset.

We explore the adaptation problem from one aspect to the other. For example, we want to train a model on annotations of DCIS and apply it on LCIS. For each aspect, we use up to three common names as a source of supervision for learning the relevance scorer, as illustrated in Table 4.2. Note that the provided list is by no means exhaustive. In fact Buckley et al. [11] provide example of 60 different verbalizations of LCIS, not counting negations.

Review dataset Our second experiment is based on a domain transfer of sentiment classification. As the source domain, we use the hotel review dataset introduced in previous work [103, 104]. For the target domain, we use the restaurant review dataset from Yelp.[1] Both datasets have ratings on a scale of 1 to 5 stars. Following previous work [7], we label reviews with ratings > 3 as positive and those with ratings < 3 as negative, and we discard the rest. The hotel dataset includes a total of around 200k reviews collected from TripAdvisor,[2] so we split 100k as labeled and the other 100k as unlabeled data. We randomly select 200k restaurant reviews as the unlabeled data in the target domain. Our testing set consists of 2k reviews. Table 4.1 summarizes the statistics of the review dataset.

The hotel reviews naturally have ratings for six aspects, including *value*, *room*

[1] `https://www.yelp.com/dataset_challenge`
[2] `https://www.tripadvisor.com/`

Dataset		#Labeled	#Unlabeled
Pathology	DCIS	23.8k	96.6k
	LCIS	10.7k	
	IDC	22.9k	
	ALH	9.2k	
Review	Hotel	100k	100k
	Restaurant	-	200k

Table 4.1: Statistics of the pathology reports dataset and the reviews dataset that we use for training. Our model utilizes both labeled and unlabeled data. The same set of unlabeled reports is used for all different aspects in the pathology reports dataset.

Aspect	Keywords
DCIS	DCIS, Ductal Carcinoma In-Situ, Ductal Carcinoma In Situ
LCIS	LCIS, Lobular Carcinoma In-Situ, Lobular Carcinoma In Situ
IDC	IDC, Invasive Ductal Carcinoma
ALH	ALH, Atypical Lobular Hyperplasia

Table 4.2: Aspects and their corresponding keywords (case insensitive) in the pathology dataset.

quality, *checkin* service, room *service*, *cleanliness* and *location*. We use the first five aspects because the sixth aspect *location* has positive labels for over 95% of the reviews and thus the trained model will suffer from the lack of negative examples. The restaurant reviews, however, only have single ratings for an *overall* impression. Therefore, we explore the task of adaptation from each of the five hotel aspects to the restaurant domain. The hotel reviews dataset also provides a total of 290 keywords for different aspects that are generated by the bootstrapping method used in [103]. We use those keywords as supervision for learning the relevance scorer.

Baselines We compare against different baselines and variants of our model.

- **SVM**: a linear SVM trained on the raw bag-of-words representation of labeled data on source and test it on target.

Method	Source		Target	
	Label	Unlabel	Label	Unlabel
SVM	✓	×	×	×
SourceOnly	✓	✓	×	×
mSDA	✓	✓	×	✓
Ours-NA	✓	✓	×	✓
Ours-NR	✓	✓	×	✓
In-Domain	×	×	✓	×
Ours-Full	✓	✓	×	✓

Table 4.3: Usage of labeled and unlabeled data in each domain by our model and other baseline methods.

- **SourceOnly**: our model trained with only labeled and unlabeled data in the source domain. No target domain data is used. It therefore has no adversarial training or target aspect-relevance scoring.

- **mSDA**: marginalized Stacked Denoising Autoencoders [15], a domain adaptation algorithm that outperforms both prior deep learning and shallow learning approaches.[3]

- **Ours-NA**: our model without the adversarial component. To implement this model we simply set the strength of adversarial training to zero.

- **Ours-NR**: our model without the aspect-relevance scorer. We set the relevance score $\tilde{r}$ to a constant 1.0 for every sentence in this model.

- **In-Domain**: supervised model trained on the full set of in-domain annotations as the performance upper bound.

Table 4.3 summarizes the usage of labeled and unlabeled data in each domain by our model and different baselines. Note that our model assumes the same set of data as Ours-NA, Ours-NR and mSDA methods.

[3]We use the publicly available implementation provided by the authors at `http://www.cse.wustl.edu/~mchen/code/mSDA.tar`. We use the hyper-parameters from the authors and their models have more parameters than ours.

Implementation details Following prior work [35], we gradually increase the adversarial strength ρ and decay the learning rate during training. We use Adam [50] as the optimization method with the default setting suggested by the authors. We also apply batch normalization [47] on the sentence encoder and apply dropout with ratio 0.2 on word embeddings and each hidden layer activation. We set the hidden layer size to 150 and pick the transformation regularization weight $\lambda^t = 0.1$ for the pathology dataset and $\lambda^t = 10.0$ for the review dataset.

4.5 Main Results

Table 4.4 summarizes the classification accuracy of different methods on the pathology dataset, including the results of six adaptation tasks. Our full model (Ours-Full) consistently achieves the best performance on each task compared with other baselines and model variants. It is not surprising that SVM and mSDA perform poorly on this dataset because they only predict labels based on an overall feature representation of the input, and do not utilize weak supervision provided by aspect-specific keywords. As a reference, we also provide a performance upper bound by training our model on the full labeled set in the target domain, denoted as In-Domain in the last column of Table 4.4. On average, the accuracy of our model is only 2.8% behind this upper bound.

Table 4.5 shows the adaptation results from each aspect in the hotel reviews to the overall ratings of restaurant reviews. Ours-Full and Ours-NR are the two best performing systems on this review dataset, attaining around 5% improvement over the mSDA baseline. Below, we summarize our findings when comparing the full model with the two model variants Ours-NA and Ours-NR.

Impact of adversarial training We first focus on comparisons between Ours-Full and Ours-NA. The only difference between the two models is that Ours-NA has no adversarial training. On the pathology dataset, our model significantly outperforms Ours-NA, yielding a 12.8% absolute average gain (see Table 4.4). On the review dataset, our model obtains 2.5% average improvement over Our-NA. As shown in Table 4.5, the gains are more significant when training on room quality and check-in service aspects, reaching 6.9% and 4.5%, respectively.

Impact of relevance scoring As shown in Table 4.4, the relevance scoring component plays a crucial role in classification on the pathology dataset. Our model achieves more than 24% improvement over Ours-NR. This is because in general aspects have zero correlations to each other in pathology reports. Therefore, it is essential for the model to have the capacity of distinguishing across different aspects in order to

Domain Source	Domain Target	SVM	Source Only	mSDA	Ours_{NA}	Ours_{NR}	$\text{Ours}_{\text{Full}}$	In-domain
LCIS	DCIS	45.8	25.2	45.0	81.2	50.0	**93.0**	96.2
DCIS	LCIS	73.8	75.4	76.2	89.0	81.2	**95.2**	97.8
DCIS	IDC	94.0	77.4	94.0	92.4	93.8	**95.4**	96.8
IDC	DCIS	71.8	62.4	73.0	87.6	81.4	**94.8**	96.2
ALH	LCIS	54.4	46.4	54.2	84.8	52.4	**93.2**	97.8
LCIS	ALH	59.0	51.6	60.4	52.6	60.0	**92.8**	96.8
Average		66.5	56.4	67.1	81.3	69.8	**94.1**	96.9

Table 4.4: **Pathology:** Classification accuracy (%) of different approaches on the pathology reports dataset, including the results of six adaptation scenarios from four different aspects (IDC, ALH, DCIS and LCIS) in breast cancer pathology reports. "mSDA" indicates the marginalized denoising autoencoder in [15]. "Ours-NA" and "Ours-NR" corresponds to our model without the adversarial training and the aspect-relevance scoring component, respectively. We also include in the last column the in-domain supervised training results of our model as the performance upper bound. Boldface numbers indicate the best accuracy for each testing scenario.

Domain Source	Domain Target	SVM	Source Only	mSDA	Ours_{NA}	Ours_{NR}	$\text{Ours}_{\text{Full}}$	In-domain
Value	Restaurant Overall	82.2	87.4	84.7	87.1	**91.1**	89.6	93.4
Room		75.6	79.3	80.3	79.7	86.1	**86.6**	
Checkin		77.8	83.0	81.0	80.9	**87.2**	85.4	
Service		82.2	88.0	83.8	88.8	87.9	**89.1**	
Cleanliness		77.9	83.2	78.4	83.1	**84.5**	81.4	
Average		79.1	84.2	81.6	83.9	**87.3**	86.4	93.4

Table 4.5: **Review:** Classification accuracy (%) of different approaches on the reviews dataset. The hotel reviews from TripAdvisor (source domain) consist of five different aspects while the restaurant reviews from Yelp (target domain) has labels only for a single *overall* aspect. Columns have the same meaning as in Table 4.4. Boldface numbers indicate the best accuracy for each testing scenario.

succeed in this task.

On the review dataset, however, we observe that relevance scoring has no significant impact on performance. On average, Ours-NR actually outperforms Ours-Full by 0.9%. This observation can be explained by the fact that different aspects in hotel reviews are highly correlated to each other. For example, the correlation between room quality and cleanliness is 0.81, much higher than aspect correlations in the pathology dataset. In other words, the sentiment is typically consistent across all sentences in a review, so that selecting aspect-specific sentences becomes unnecessary. Moreover, our supervision for the relevance scorer is weak and noisy because the aspect keywords are obtained in a semi-automatic way. Therefore, it is not surprising that Ours-NR sometimes delivers a better classification accuracy than Ours-Full.

4.6 Analysis

When is adversarial training useful? As shown in Table 4.4 and 4.5, the gains from using adversarial training vary significantly over different adaptation scenarios. To better understand when adversarial training is useful, we further test on four synthetic datasets that represent different challenges in domain adaptation. We generate the synthetic datasets as follows. Each synthetic document consists of around ten randomly generated sentences. Each sentence is always associated with a random aspect and contains a special token as the aspect name (e.g. `ASP0_NAME0`). Except for the first dataset, each sentence also contains another special token as the aspect polarity (e.g. `ASP0_POS_NAME0` or `POS_NAME0`). Aspect names and polarity tokens each have about ten different options (i.e. `NAME0` to `NAME9`). Document labels are either positive or negative, indicated by the polarity tokens of the focal aspect, except for the first dataset (see below). The adaptation task is to transfer the model from one aspect to another. The characteristics of each dataset are as follows.

- SYN1: Sentences do not contain polarity tokens. Instead, class labels are indicated by the occurrence of aspect names. The label is positive if a name of the particular aspect (e.g. `ASP0_NAME0`) occurs, otherwise negative.

- SYN2: Class labels are indicated by polarity tokens. To make the transfer a possible task, positive polarity tokens have 20% overlap across aspects. In other words, for 20% of the sentences with positive polarity tokens, the tokens have the format `POS_NAME0` while the rest have the format `ASP0_POS_NAME0`. In contrast, negative polarity tokens have no overlap.

- SYN3: Both positive and negative polarity tokens have 20% overlap across aspects.

- SYN4: The last dataset is similar to the third one. However, both positive and negative polarity tokens have only 5% overlap across aspects.

To distinguish aspect names and polarity words from others, we surround each of them with a different distribution of context words. We fill in the rest place of sentences

Method	Syn1	Syn2	Syn3	Syn4
Ours-NA	75.4	52.8	100.0	49.2
Ours-Full	99.8	100.0	100.0	100.0

Table 4.6: Classification accuracy on four synthetic datasets that represent different challenges in domain adaptation.

with other random words. Here are two examples synthetic sentence, both associated with aspect 0 (`ASP0`). The label of the first sentence is negative while the second one is positive.

> WORD1146 ASP0_CTXT1 ASP0_CTXT3 ASP0_CTXT8 ASP0_NAME4 ASP0_CTXT9 ASP0_CTXT7 ASP0_CTXT1 ASP0_NEU_CTXT7 ASP0_NEU_CTXT7 ASP0_NEU_CTXT4 ASP0_NEG_NAME3 ASP0_NEU_CTXT5 ASP0_NEG_CTXT1 ASP0_NEU_CTXT4 WORD9
>
> WORD402 WORD48 WORD1242 WORD94 ASP0_CTXT0 ASP0_CTXT2 ASP0_NAME9 ASP0_CTXT7 ASP0_CTXT5 NEU_CTXT2 NEU_CTXT2 POS_NAME0 POS_CTXT6 POS_CTXT5 NEU_CTXT6 WORD3815 WORD3595 WORD3326 WORD4942 WORD4909

Table 4.6 summarizes the prediction accuracy of Ours-Full and Ours-NA. We can see that our full model with adversarial training successfully solve all transfer tasks. The model without adversarial training (Ours-NA) only solve the third task. This task has the largest amount of common polarity words across aspects, and therefore is the easiest task.

Impact of the reconstruction loss Table 4.7 summarizes the impact of the reconstruction loss on the model performance. For our full model (Ours-Full), adding the reconstruction loss yields an average of 4.6% gain on the pathology dataset and 5.2% on the review dataset.

To analyze the reasons behind this difference, consider Figure 4-5 that shows the heat maps of the learned document representations on the review dataset. The top half of the matrices corresponds to input documents from the source domain and the

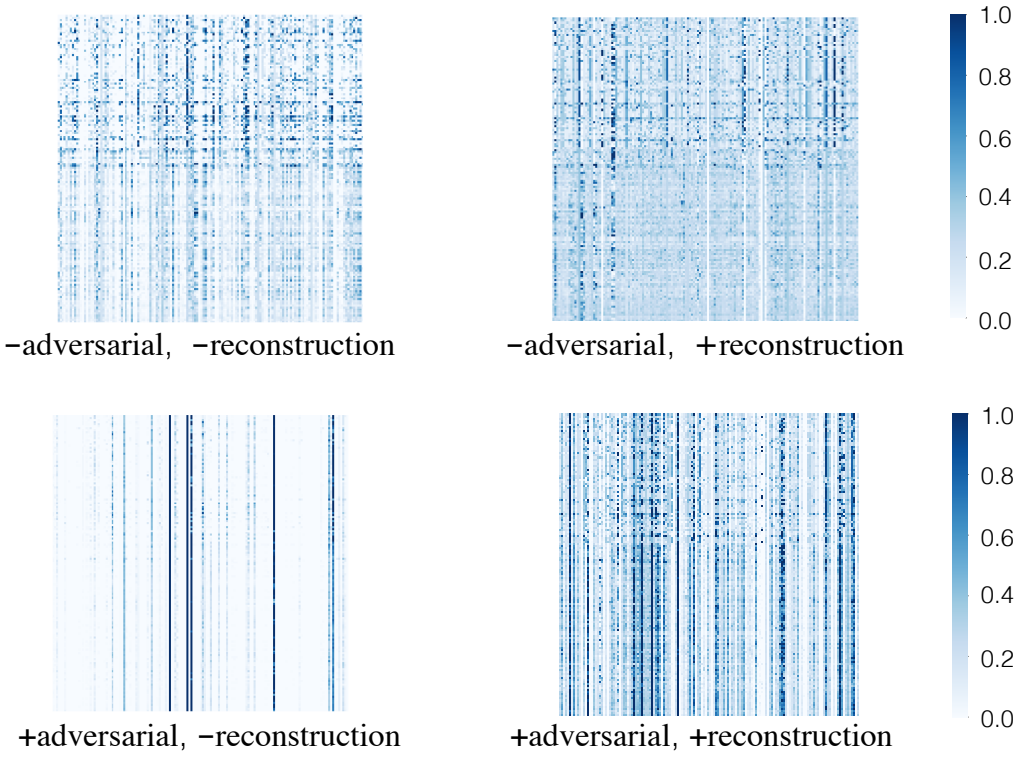

Figure 4-5: Heat map of 150×150 matrices. Each row of the matrix corresponds to the vector representation of a document that comes from either the source domain (top half of each matrix) or the target domain (bottom half of each matrix). Models are trained on the review dataset when room quality is the source aspect.

bottom half corresponds to the target domain. Unlike the top two matrices, the two matrices in the bottom part have no significant difference between the two halves, indicating that adversarial training helps learning of domain-invariant representations. However, adversarial training also removes a lot of information from representations, as the bottom-left matrix is much more sparse than the top-left one. The bottom-right matrix shows that adding reconstruction loss effectively addresses this sparsity issue. Almost 85% entries of the bottom-left matrix have small values ($< 10^{-6}$) while the sparsity is only about 30% for bottom-right one. Moreover, the standard deviation of the bottom-right matrix is also ten times higher than the bottom-left one. These comparisons demonstrate that the reconstruction loss function improves both the richness and diversity of the learned representations. Note that in the case of no adversarial training (Ours-NA), adding the reconstruction component has no clear

Dataset	Ours-Full		Ours-NA	
	-REC.	+REC.	-REC.	+REC.
Pathology	89.5	94.1	78.6	81.3
Review	80.8	86.4	85.0	83.9

Table 4.7: Impact of adding the reconstruction component in the model, measured by the average accuracy on each dataset. +REC. and -REC. denote the presence and absence of the reconstruction loss, respectively.

Dataset	$\lambda^t = 0$	$0 < \lambda^t < \infty$	$\lambda^t = \infty$
Pathology	84.1	94.1	77.0
Review	80.9	86.4	84.3

Table 4.8: The effect of regularization of the transformation layer λ^t on the performance.

effect. This is expected because the main motivation of adding this component is to achieve a more robust adversarial training.

Regularization on the transformation layer Table 4.8 shows the averaged accuracy with different regularization weights λ^t in Equation 4.5. We change λ^t to reflect different model variants. First, $\lambda^t = \infty$ corresponds to the removal of the transformation layer because the transformation is always identity in this case. Our model performs better than this variant on both datasets, yielding an average improvement of 17.1% on the pathology dataset and 2.1% on the review dataset. This result indicates the importance of adding the transformation layer. Second, using zero regularization ($\lambda^t = 0$) also consistently results in inferior performance, such as 10% loss on the pathology dataset. We hypothesize that zero regularization will dilute the effect from reconstruction because of too much flexibility in transformation. As a result, the transformed representation will become sparse due to the adversarial training, leading to the performance loss.

Examples of neighboring reviews Finally, we illustrate in Figure 4-6 a case study on the characteristics of learned abstract representations by different models.

(a) Restaurant Reviews:

- the fries were **undercooked** and **thrown haphazardly** into the sauce holder . the shrimp was **over cooked** and just **deep fried** . … even the water **tasted weird** .

(b) Nearest Hotel Reviews by **Ours-Full:**

- the room was **old** . … we did n't like the night shows at all . …
- however , the decor **was just fair** . … in the second bedroom it literally **rained water from above** .
- the room decor **was not entirely modern** . … we just had the run of the mill hotel room **without a view** .

(c) Nearest Hotel Reviews by **Ours-NA:**

- rest room in this restaurant is **very dirty** . …
- the only **problem** i had was that … i was very ill with what was suspected to be **food poison**
- probably **the noisiest room** he could have given us in the whole hotel .

Figure 4-6: Examples of restaurant reviews and their nearest neighboring hotel reviews induced by different models (part (b) and (c)). The distance between reviews is measure by the cosine similarity between their vector representations induced by the model. We use room quality as the source aspect and we show the sentences that have high relevance score. The sentiment phrases of each review are in blue, and some reviews are also shortened for space.

Part (a) of Table 4-6 shows an example restaurant review. Sentiment phrases in this example are mostly food-specific, such as "undercooked" and "tasted weird". In the other two parts, we show example hotel reviews that are nearest neighbors to the restaurant reviews, measured by cosine similarity between their representations. In part (b), many sentiment phrases are specific for room quality, such as "old" and "rained water from above". In part (c), however, most sentiment phrases are either common sentiment expressions (e.g. dirty) or food-related (e.g. food poison), even though the focus of the reviews is room quality of hotels. This observation indicates that adversarial training (Ours-Full) successfully learns to eliminate domain-specific information and to map those domain-specific words into similar domain-invariant representations. In contrast, Ours-NA only captures domain-invariant features from phrases that commonly present in both domains.

4.7 Conclusions

In this chapter, we propose a novel aspect-augmented adversarial network for cross-aspect and cross-domain adaptation tasks. Experimental results demonstrate that our approach successfully learns invariant representation from aspect-relevant fragments, yielding significant improvement over the mSDA baseline and our model variants. The effectiveness of our approach suggests the potential application of adversarial networks to a broader range of NLP tasks for improved representation learning, such as machine translation and language generation.

Chapter 5

Conclusions and Future Work

In this thesis, we have shown how to leverage annotations in other tasks to boost unsupervised learning performance of our target task. In particular, we demonstrate the effectiveness of our methods in two scenarios: transfer across languages and transfer across domains.

In the multilingual scenario, we study the transfer learning problem in the context of dependency parsing and POS tagging tasks. For dependency parsing, we present a hierarchical tensor-based approach that allows the model to incorporate linguistic typology knowledge in the form of capturing desired feature combination. As shown in our results, our model outperforms traditional tensor methods and the prior state-of-the-art multilingual parser on a wide range of datasets. In the case of POS tagging, our focus is to understand how little parallel data is necessary to enable effective multilingual transfer on the word level. We demonstrate that only ten translation pairs suffice for this task, indicating a promising direction on using fewer parallel resources for better multilingual transfer.

In the domain transfer scenario, we design an aspect-augmented adversarial network that handles both traditional domain transfer as well as aspect transfer. We present an adversarial training framework combined with aspect-driven encoding to solve this aspect transfer problem. Experimental results show that both aspect-driven encoding and adversarial training play a crucial role in the overall performance, yielding significant improvements on a pathology report dataset and a review dataset.

5.1 Future Work

The work presented in this thesis is not an end. It is only a beginning. Our work can be extended in a number of ways, as discussed below:

Multi-source Transfer The models we have described in Chapter 3 and Chapter 4 transfer knowledge only from one source (e.g. language, domain or aspect). However, as discussed in Chapter 2 and in previous work [68, 75, 97], it is usually beneficial to transfer from more than one source language at the same time. Intuitively, transferring from multiple sources allows the model to selectively learn from a closer language or from a more related domain, resulting a better transfer performance. Thus, one essential research direction is to develop methods that are able to utilize multiple training sources and automatically discover a better transfer regime, or use some external knowledge as a guidance to this selective transfer process.

Model-agnostic Multilingual Transfer Both our multilingual parser and the state-of-the-art parsers [97, 75] utilize linguistic typological features by encoding them in the model structure or in the feature construction process. As a result, all these methods can only be applied on the basis of simple parsing models such as a generative parser or a first-order parser. This is clearly not ideal because in supervised settings these simple models have much lower performance than current state-of-the-art parsers that use neural networks [3] or high-order features [120]. The transfer performance is therefore capped by the supervised training upper bound. While annotation projection [86, 96] is a common approach for model-agnostic transfer, it typically requires large amount of parallel data to establish the projection. One interesting question is whether we can achieve model-agnostic transfer without any parallel resources. Such transfer methods will be attractive because they can capitalize on the use of most advanced supervised models (e.g. deep neural networks) to boost performance.

Model-level Transfer In Chapter 4, we demonstrate how to achieve transfer using a shared classifier for both source and target tasks. The key assumption is that the model is able to learn invariant feature representations of inputs, so the source classifier can be directly transferred to the target domain upon these representations. However, this assumption may not hold well in every practical scenario. For example, we show in Chapter 3 that fine-tuning the model results in a better performance over direct transfer models. The model shifts from the one trained on the source language and better fits the target language. Previous work also demonstrates that adding a small perturbation function between source and target classifiers yields a performance gain [60]. These observations indicate that learning a model-level transformation from source to target, even in an unsupervised manner, may be helpful for the overall transfer performance.

Appendix A

Learning Hierarchical Tensors

A.1 Derivations of Parameter Updates

This section presents the derivation of parameter updates of hierarchical tensors. We start by formalizing our learning problem. Let $D = \{(\hat{\mathbf{x}}, \hat{\mathbf{y}})\}$ be a collection of training example pairs, where each pair consists of a sentence $\hat{\mathbf{x}}$ (with universal POS annotations) and the corresponding gold dependency tree $\hat{\mathbf{y}}$. The goal of learning is to search values for the parameters $\theta = (\mathbf{w}, H, M, D, L, T_u, T_l, H_c, M_c)$ that optimize the combined scoring function below for parsing performance.

$$S_\theta(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = \gamma \sum_{h \xrightarrow{l} m \in \hat{\mathbf{y}}} \mathbf{w} \cdot \phi(h \xrightarrow{l} m) + (1-\gamma) \sum_{h \xrightarrow{l} m \in \hat{\mathbf{y}}} S_{tensor}(h \xrightarrow{l} m) \tag{A.1}$$

See Table 2.3 for notation of parameters. γ is the hyper-parameter that balances the two scores in our model. We suppress the dependence of the scoring function $S_{tensor}(\cdot)$ on θ whenever it is clear from context. We optimize the parameter values in a maximum soft-margin framework. Specifically, for each training pair $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$, we adjust parameter values to separate the gold tree and other incorrect alternatives.

$$S_\theta(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \geq S_\theta(\hat{\mathbf{x}}, \mathbf{y}) + ||\hat{\mathbf{y}} - \mathbf{y}||_1, \quad \forall \mathbf{y} \in \mathcal{Y}(\hat{\mathbf{x}}) \tag{A.2}$$

where $||\hat{\mathbf{y}} - \mathbf{y}||_1$ is the number of mismatched arcs between the two trees.

To solve this learning objective, we adopt the passive-aggressive (PA) online learning algorithm [20]. In this algorithm, the parameters are updated successively after each training sentence. Consider the t-th update where parameters have value $\theta^{(t)}$ and are updated with the training pair $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$. The algorithm first checks whether the constraint in Equation A.2 is violated under $\theta^{(t)}$. This requires "cost-augmented decoding" to find the maximum violation with respect to the gold tree:

$$\tilde{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}(\hat{\mathbf{x}})} S_{\theta^{(t)}}(\hat{\mathbf{x}}, \mathbf{y}) + ||\mathbf{y} - \hat{\mathbf{y}}||_1$$

When the constraint A.2 is violated (i.e. $\tilde{\mathbf{y}} \neq \hat{\mathbf{y}}$), we seek new parameters $\theta^{(t+1)}$ to fix this violation by solving

$$\begin{aligned} &\min_{\theta, \xi \geq 0} \frac{1}{2}||\theta - \theta^{(t)}||^2 + C\xi \\ &\text{s.t.} \quad S_\theta(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \geq S_\theta(\hat{\mathbf{x}}, \tilde{\mathbf{y}}) + ||\hat{\mathbf{y}} - \tilde{\mathbf{y}}||_1 - \xi \end{aligned}$$

where $\xi \geq 0$ is a slack variable and and C is a regularization hyper-parameter. This problem has a closed form solution.

$$\theta^{(t+1)} \leftarrow \theta^{(t)} + \min\left\{C, \frac{loss(\theta)}{||\nabla\theta||^2}\right\} \nabla\theta$$

where

$$loss(\theta) = \max\{0, S_{\theta^{(t)}}(\hat{\mathbf{x}}, \tilde{\mathbf{y}}) + ||\hat{\mathbf{y}} - \tilde{\mathbf{y}}||_1 - S_{\theta^{(t)}}(\hat{\mathbf{x}}, \hat{\mathbf{y}})\}$$

$$\nabla\theta = \frac{\partial S_\theta(\hat{\mathbf{x}}, \hat{\mathbf{y}})}{\partial\theta} - \frac{\partial S_\theta(\hat{\mathbf{x}}, \tilde{\mathbf{y}})}{\partial\theta} \tag{A.3}$$

Note that θ is the set of all parameters, and the update jointly adjusts all low-rank matrices and the traditional weight vector $\mathbf{w}$. We skip the detail proof of the solution because it has been shown in prior work [20] and is beyond this thesis. By plugging the combined scoring (Equation A.1) into $\nabla\theta$ (Equation A.3), we have the gradient

for the tradition weight vector

$$\nabla \mathbf{w} = \gamma \left[\sum_{h \xrightarrow{l} m \in \hat{\mathbf{y}}} \phi(h \xrightarrow{l} m) - \sum_{h \xrightarrow{l} m \in \tilde{\mathbf{y}}} \phi(h \xrightarrow{l} m) \right]$$

The gradient w.r.t to parameter matrices of the tensor takes a similar form. For example, the gradient of H is

$$\nabla H = (1-\gamma) \left[\sum_{h \xrightarrow{l} m \in \hat{\mathbf{y}}} \frac{S_{tensor}(h \xrightarrow{l} m)}{\partial H} - \sum_{h \xrightarrow{l} m \in \tilde{\mathbf{y}}} \frac{S_{tensor}(h \xrightarrow{l} m)}{\partial H} \right]$$

Next, we focus on the derivation of $\partial S_{tensor}(h \xrightarrow{l} m)/\partial H$. First, recall that we can view our hierarchical tensor as the combination of three multiway tensors with parameter sharing.

$$\begin{aligned} S_{tensor}(h \xrightarrow{l} m) = & \sum_{i=1}^{r} [H_c \phi_{h_c}]_i [M_c \phi_{m_c}]_i [T_l \phi_{t_l}]_i \\ & + \sum_{i=1}^{r} [H_c \phi_{h_c}]_i [M_c \phi_{m_c}]_i [L \phi_l]_i [T_u \phi_{t_u}]_i \\ & + \sum_{i=1}^{r} [H_c \phi_{h_c}]_i [M_c \phi_{m_c}]_i [L \phi_l]_i [H \phi_h]_i [M \phi_m]_i [D \phi_d]_i \end{aligned} \tag{A.4}$$

The definition of features vectors $\phi_\cdot$ are summarized in Table 2.3, and we suppress their dependence on $h \xrightarrow{l} m$. We denote the score of each multiway tensor as

$$\begin{aligned} S_{t1}(h \xrightarrow{l} m) &\equiv \sum_{i=1}^{r} [H_c \phi_{h_c}]_i [M_c \phi_{m_c}]_i [T_l \phi_{t_l}]_i \\ S_{t2}(h \xrightarrow{l} m) &\equiv \sum_{i=1}^{r} [H_c \phi_{h_c}]_i [M_c \phi_{m_c}]_i [L \phi_l]_i [T_u \phi_{t_u}]_i \\ S_{t3}(h \xrightarrow{l} m) &\equiv \sum_{i=1}^{r} [H_c \phi_{h_c}]_i [M_c \phi_{m_c}]_i [L \phi_l]_i [H \phi_h]_i [M \phi_m]_i [D \phi_d]_i \end{aligned}$$

The gradient w.r.t H can be rewritten as

$$\frac{S_{tensor}(h \xrightarrow{l} m)}{\partial H} = \frac{S_{t1}(h \xrightarrow{l} m)}{\partial H} + \frac{S_{t2}(h \xrightarrow{l} m)}{\partial H} + \frac{S_{t3}(h \xrightarrow{l} m)}{\partial H}$$

Because only S_{t3} relates to H

$$\begin{aligned}\frac{S_{tensor}(h \xrightarrow{l} m)}{\partial H} &= \frac{S_{t3}(h \xrightarrow{l} m)}{\partial H} \\ &= [(H_c\phi_{h_c}) \odot (M_c\phi_{m_c}) \odot (L\phi_l) \odot (M\phi_m) \odot (D\phi_d)] \otimes \phi_h\end{aligned}$$

where $(u \odot v)_i = u_i v_i$ is the element-wise product and $\otimes$ is the tensor product. Other parameter matrices can be computed similarly.

$$\frac{S_{tensor}(h \xrightarrow{l} m)}{\partial M} = [(H_c\phi_{h_c}) \odot (M_c\phi_{m_c}) \odot (L\phi_l) \odot (H\phi_H) \odot (D\phi_d)] \otimes \phi_h$$

$$\frac{S_{tensor}(h \xrightarrow{l} m)}{\partial D} = [(H_c\phi_{h_c}) \odot (M_c\phi_{m_c}) \odot (L\phi_l) \odot (H\phi_H) \odot (M\phi_m)] \otimes \phi_h$$

$$\frac{S_{tensor}(h \xrightarrow{l} m)}{\partial T_u} = [(H_c\phi_{h_c}) \odot (M_c\phi_{m_c}) \odot (L\phi_l)] \otimes \phi_{t_u}$$

$$\begin{aligned}\frac{S_{tensor}(h \xrightarrow{l} m)}{\partial L} &= [(H_c\phi_{h_c}) \odot (M_c\phi_{m_c}) \odot (H\phi_H) \odot (M\phi_m) \odot (D\phi_d)] \otimes \phi_l \\ &\quad + [(H_c\phi_{h_c}) \odot (M_c\phi_{m_c}) \odot (T_u\phi_{t_u})] \otimes \phi_l\end{aligned}$$

$$\frac{S_{tensor}(h \xrightarrow{l} m)}{\partial T_l} = [(H_c\phi_{h_c}) \odot (M_c\phi_{m_c})] \otimes \phi_{t_l}$$

$$\frac{S_{tensor}(h \xrightarrow{l} m)}{\partial H_c} = [(M_c\phi_{m_c}) \odot (L\phi_l) \odot (H\phi_H) \odot (M\phi_m) \odot (D\phi_d)] \otimes \phi_{h_c}$$
$$+ [(M_c\phi_{m_c}) \odot (L_c\phi_{l_c}) \odot (T_u\phi_{t_u})] \otimes \phi_{h_c}$$
$$+ [(M_c\phi_{m_c}) \odot (T_l\phi_{t_l})] \otimes \phi_{h_c}$$

$$\frac{S_{tensor}(h \xrightarrow{l} m)}{\partial M_c} = [(H_c\phi_{h_c}) \odot (L\phi_l) \odot (H\phi_H) \odot (M\phi_m) \odot (D\phi_d)] \otimes \phi_{m_c}$$
$$+ [(H_c\phi_{h_c}) \odot (L_c\phi_{l_c}) \odot (T_u\phi_{t_u})] \otimes \phi_{m_c}$$
$$+ [(H_c\phi_{h_c}) \odot (T_l\phi_{t_l})] \otimes \phi_{m_c}$$

Appendix B

Multilingual Transfer for POS Tagging

B.1 Parameter Updates of Unsupervised HMM

This section presents the derivation of parameter updates of our unsupervised HMM. We start by showing the log-likelihood objective that we want to maximize during the M-step.

$$l(\theta) = \sum_{y',y \in \mathcal{Y} \times \mathcal{Y}} e_{y',y} \log p_\theta(y'|y) + \sum_{x,y \in \mathcal{X} \times \mathcal{Y}} e_{x,y} \log p_\theta(x|y) - \frac{\beta}{2}||\theta - \theta_0||_2^2 \tag{B.1}$$

where

$$p_\theta(y'|y) \propto \exp\{\theta_{y',y}\}$$

$$p_\theta(x|y) \propto \exp\{\boldsymbol{v}_x^T \boldsymbol{P} \boldsymbol{M} \boldsymbol{\mu}_y + \theta_{x,y}\}$$

Note that we use p_θ instead of p_θ^t (shown in Equation 3.5) for simplicity when in the context of no ambiguity. $\mathcal{Y}$ denotes all possible tags and $\mathcal{X}$ denotes all possible words. Next, we show the derivation of gradients w.r.t $\theta_{y',y}, \boldsymbol{M}, \boldsymbol{\mu}_y$ and $\theta_{x,y}$. The gradients of the regularization term in Equation B.1 are straightforward, so we will first skip

them. We start by showing that the gradient of $\log p_\theta(y'|y)$ takes the form

$$\frac{\partial \log p_\theta(y'|y)}{\partial \theta_{y',y}} = \frac{1}{p_\theta(y'|y)} \frac{\partial}{\partial \theta_{y',y}} \frac{\exp\{\theta_{y',y}\}}{\sum_{u \in \mathcal{Y}} \exp\{\theta_{u,y}\}}$$

$$= 1 - p_\theta(y'|y)$$

and for other $y'' \neq y'$

$$\frac{\partial \log p_\theta(y''|y)}{\partial \theta_{y',y}} = \frac{1}{p_\theta(y''|y)} \frac{\partial}{\partial \theta_{y'',y}} \frac{\exp\{\theta_{y',y}\}}{\sum_{u \in \mathcal{Y}} \exp\{\theta_{u,y}\}}$$

$$= -p_\theta(y'|y)$$

By plugging them into Equation B.1

$$\frac{\partial l(\theta)}{\partial \theta_{y',y}} = e_{y',y} - p_\theta(y'|y) \sum_{u \in \mathcal{Y}} e_{u,y} \tag{B.2}$$

Similarly, the gradient with regard to $\theta_{x,y}$ is

$$\frac{\partial l(\theta)}{\partial \theta_{x,y}} = e_{x,y} - p_\theta(x|y) \sum_{w \in \mathcal{X}} e_{u,y} \tag{B.3}$$

For parameters $\mu_y \in \mathbb{R}^d$ (d is the dimension of word embeddings)

$$\frac{\partial \log p_\theta(x|y)}{\partial \boldsymbol{\mu_y}} = \frac{1}{p_\theta(x|y)} \frac{\partial}{\partial \boldsymbol{\mu_y}} \frac{\exp\{\boldsymbol{v}_x^T \boldsymbol{P} \boldsymbol{M} \boldsymbol{\mu}_y + \theta_{x,y}\}}{\sum_{w \in \mathcal{X}} \exp\{\boldsymbol{v}_x^T \boldsymbol{P} \boldsymbol{M} \boldsymbol{\mu}_y + \theta_{w,y}\}}$$

$$= \boldsymbol{v}_x^T \boldsymbol{P} \boldsymbol{M} - \sum_{w \in \mathcal{X}} p_\theta(w|y) \boldsymbol{v}_w^T \boldsymbol{P} \boldsymbol{M}$$

Therefore

$$\frac{\partial l(\theta)}{\partial \boldsymbol{\mu_y}} = \sum_{x \in \mathcal{X}} e_{x,y} \boldsymbol{v}_x^T \boldsymbol{P} \boldsymbol{M} - \sum_{x \in \mathcal{X}} e_{x,y} \sum_{x \in \mathcal{X}} p_\theta(x|y) \boldsymbol{v}_x^T \boldsymbol{P} \boldsymbol{M} \tag{B.4}$$

Finally, the gradient with regard to $\boldsymbol{M}$ is

$$\begin{aligned}\frac{\partial \log p_\theta(x|y)}{\partial \boldsymbol{M}} &= \frac{1}{p_\theta(x|y)} \frac{\partial}{\partial \boldsymbol{M}} \frac{\exp\{\boldsymbol{v}_x^T \boldsymbol{P} \boldsymbol{M} \boldsymbol{\mu}_y + \theta_{x,y}\}}{\sum_{w \in \mathcal{X}} \exp\{\boldsymbol{v}_x^T \boldsymbol{P} \boldsymbol{M} \boldsymbol{\mu}_y + \theta_{w,y}\}} \\ &= \boldsymbol{P}^T \boldsymbol{v}_x \boldsymbol{\mu}_y^T - \sum_{w \in \mathcal{X}} p_\theta(w|y) \boldsymbol{P}^T \boldsymbol{v}_w \boldsymbol{\mu}_y^T\end{aligned}$$

and

$$\frac{\partial l(\theta)}{\partial \boldsymbol{M}} = \sum_{y \in \mathcal{Y}} \left\{ \sum_{x \in \mathcal{X}} e_{x,y} \boldsymbol{P}^T \boldsymbol{v}_x \boldsymbol{\mu}_y^T - \sum_{x \in \mathcal{X}} e_{x,y} \sum_{x \in \mathcal{X}} p_\theta(x|y) \boldsymbol{P}^T \boldsymbol{v}_x \boldsymbol{\mu}_y^T \right\} \tag{B.5}$$

B.2 Detail Results of typological Prediction

In this section, we show the detail results of five linguistic typological properties for each language: subject-verb, verb-object, adjective-noun, adposition-noun and demonstrative noun. We predict values for six languages: Danish, German, Spanish, Finnish, Hungarian and Indonesian. The evaluation therefore consists of 30 test cases. We demonstrate the performance of using POS tag sequences from four systems: prototype-driven method [42] (Prototype), direct transfer model (Direct Transfer), our full model (Transfer+EM) and gold POS annotations (Gold).

Subject-Verb Table B.1 shows the prediction results of the subject-verb typological feature. ✓ indicates correct predictions and × indicates wrong predictions. The feature has three possible values: Subject-Verb (SV), Verb-Subject (VS) and No-dominant-order (NDO).

	Prototype	Direct Transfer	Transfer+EM	Gold	Ground Truth
Danish	✓	✓	✓	✓	SV
German	×	✓	✓	✓	SV
Spanish	×	✓	✓	✓	NDO
Finnish	✓	✓	✓	✓	SV
Hungarian	✓	✓	✓	✓	SV
Indonesian	✓	✓	✓	✓	SV
Score	4	6	6	6	-

Table B.1: Predictions of the subject-verb typological feature using POS tags from different methods.

Verb-Object Table B.2 shows the prediction results of the verb-object typological feature. The feature has three possible values: Verb-Object (VO), Object-Verb (OV) and No-dominant-order (NDO).

	Prototype	Direct Transfer	Transfer+EM	Gold	Ground Truth
Danish	✓	✓	✓	✓	VO
German	✓	×	×	✓	NDO
Spanish	×	✓	✓	✓	VO
Finnish	×	✓	✓	✓	VO
Hungarian	×	×	×	✓	VO
Indonesian	×	✓	✓	✓	VO
Score	2	4	4	6	-

Table B.2: Predictions of the verb-object typological feature using POS tags from different methods.

Adjective-Noun Table B.3 shows the prediction results of the adjective-noun typological feature. The feature has two possible values: Adjective-Noun (AN), Noun-Adjective (NA).

	Prototype	Direct Transfer	Transfer+EM	Gold	Ground Truth
Danish	✓	✓	✓	✓	AN
German	×	×	✓	✓	AN
Spanish	×	✓	✓	✓	NA
Finnish	✓	✓	✓	✓	AN
Hungarian	✓	✓	✓	✓	AN
Indonesian	×	×	×	×	NA
Score	3	4	5	5	-

Table B.3: Predictions of the adjective-noun typological feature using POS tags from different methods.

Adposition-Noun Table B.4 shows the prediction results of the adposition-noun typological feature. The feature has two possible values: Preposition (Prep), Postposition (Post).

	Prototype	Direct Transfer	Transfer+EM	Gold	Ground Truth
Danish	✓	✓	✓	✓	Prep
German	✓	✓	✓	✓	Prep
Spanish	✓	✓	✓	✓	Prep
Finnish	✓	×	✓	✓	Post
Hungarian	×	×	×	×	Post
Indonesian	✓	✓	✓	✓	Prep
Score	5	4	5	5	-

Table B.4: Predictions of the adposition-noun typological feature using POS tags from different methods.

Demonstrative-Noun Table B.5 shows the prediction results of the demonstrative-noun typological feature. The feature has two possible values: Demonstrative-Noun (DN), Noun-Demonstrative (ND).

	Prototype	Direct Transfer	Transfer+EM	Gold	Ground Truth
Danish	×	×	✓	✓	DN
German	✓	✓	✓	✓	DN
Spanish	✓	✓	✓	✓	DN
Finnish	×	×	×	✓	DN
Hungarian	✓	×	×	✓	DN
Indonesian	✓	×	✓	✓	ND
Score	4	2	4	6	-

Table B.5: Predictions of the demonstrative-noun typological feature using POS tags from different methods.

Bibliography

[1] Traian E. Abrudan, Jan Eriksson, and Visa Koivunen. Steepest descent algorithms for optimization under unitary matrix constraint. *IEEE Transaction on Signal Processing*, 56(3):1134–1147, 2008.

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

[3] Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. Globally normalized transition-based neural networks. *arXiv preprint arXiv:1603.06042*, 2016.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[5] Taylor Berg-Kirkpatrick and Dan Klein. Phylogenetic grammar induction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1288–1297. Association for Computational Linguistics, 2010.

[6] Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. Painless unsupervised learning with features. In *Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 582–590. Association for Computational Linguistics, 2010.

[7] John Blitzer, Mark Dredze, Fernando Pereira, et al. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, volume 7, pages 440–447, 2007.

[8] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Neural Information Processing Systems (NIPS)*, 2016.

[9] Caroline Brun, Julien Perez, and Claude Roux. Xrce at semeval-2016 task 5: Feedbacked ensemble modeling on syntactico-semantic knowledge for aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 277–281, 2016.

[10] Sabine Buchholz and Erwin Marsi. Conll-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 149–164. Association for Computational Linguistics, 2006.

[11] Julliette M Buckley, Suzanne B Coopey, John Sharko, Fernanda Polubriaginof, Brian Drohan, Ahmet K Belli, Elizabeth MH Kim, Judy E Garber, Barbara L Smith, Michele A Gadd, et al. The feasibility of using natural language processing to extract clinical information from breast pathology reports. *Journal of pathology informatics*, 3(1):23, 2012.

[12] Ming-Wei Chang, Lev Ratinov, and Dan Roth. Guiding semi-supervision with constraint-driven learning. In *Annual Meeting-Association for Computational Linguistics*, volume 45, page 280, 2007.

[13] Desai Chen, Chris Dyer, Shay B Cohen, and Noah A Smith. Unsupervised bilingual POS tagging with markov random fields. In *Proceedings of the First Workshop on Unsupervised Learning in NLP*, pages 64–71. Association for Computational Linguistics, 2011.

[14] Kan Chen, Jiang Wang, Liang-Chieh Chen, Haoyuan Gao, Wei Xu, and Ram Nevatia. Abc-cnn: An attention based convolutional neural network for visual question answering. *arXiv preprint arXiv:1511.05960*, 2015.

[15] Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. Marginalized denoising autoencoders for domain adaptation. *arXiv preprint arXiv:1206.4683*, 2012.

[16] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*, 2016.

[17] Sumit Chopra, Suhrid Balakrishnan, and Raghuraman Gopalan. Dlid: Deep learning for domain adaptation by interpolating between domains. In *ICML Workshop on Challenges in Representation Learning*, 2013.

[18] Shay B Cohen and Noah A Smith. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 74–82. Association for Computational Linguistics, 2009.

[19] Shay B Cohen, Dipanjan Das, and Noah A Smith. Unsupervised structure prediction with non-parallel multilingual guidance. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 50–61. Association for Computational Linguistics, 2011.

[20] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *The Journal of Machine Learning Research*, 7:551–585, 2006.

[21] Dipanjan Das and Slav Petrov. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 600–609. Association for Computational Linguistics, 2011.

[22] Hal Daumé III. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*, 2009.

[23] Marie-Catherine De Marneffe and Christopher D Manning. The stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8. Association for Computational Linguistics, 2008.

[24] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, pages 647–655, 2014.

[25] Matthew S Dryer, David Gil, Bernard Comrie, Hagen Jung, Claudia Schmidt, et al. The world atlas of language structures. 2005.

[26] Long Duong, Paul Cook, Steven Bird, and Pavel Pecina. Increasing the quality and quantity of source language data for unsupervised cross-lingual POS tagging. In *Sixth International Joint Conference on Natural Language Processing, IJCNLP 2013, Nagoya, Japan, October 14-18, 2013*, pages 1243–1249, 2013.

[27] Long Duong, Trevor Cohn, Karin Verspoor, Steven Bird, and Paul Cook. 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*, pages 886–897, 2014.

[28] Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. Cross-lingual transfer for unsupervised dependency parsing without parallel data. *Proceedings of the SIGNLL Conference on Computational Natural Language Learning*, page 113, 2015.

[29] Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. A neural network model for low-resource universal dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 339–348. Citeseer, 2015.

[30] Greg Durrett, Adam Pauls, and Dan Klein. 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*, pages 1–11. Association for Computational Linguistics, 2012.

[31] Jacob Eisenstein. Unsupervised learning for lexicon-based classification. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2017.

[32] Jason M Eisner. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pages 340–345. Association for Computational Linguistics, 1996.

[33] Manaal Faruqui and Chris Dyer. Improving vector space word representations using multilingual correlation. In *Proceedings of the Annual Conference of the European Chapter of the Association for Computational Linguistics.* Association for Computational Linguistics, 2014.

[34] Daniel Fried, Tamara Polajnar, and Stephen Clark. Low-rank tensors for verbs in compositional distributional semantics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2015.

[35] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

[36] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *arXiv preprint arXiv:1505.07818*, 2015.

[37] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 513–520, 2011.

[38] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[39] Stephan Gouws, Yoshua Bengio, and Greg Corrado. Bilbowa: Fast bilingual distributed representations without word alignments. *arXiv preprint arXiv:1410.2455*, 2014.

[40] Trond Grenager, Dan Klein, and Christopher D Manning. Unsupervised learning of field segmentation models for information extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 371–378. Association for Computational Linguistics, 2005.

[41] Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. Cross-lingual dependency parsing based on distributed representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 1234–1244, 2015.

[42] Aria Haghighi and Dan Klein. Prototype-driven learning for sequence models. In *Proceedings of the 2006 Annual Conference of the North American Chapter*

of the Association for Computational Linguistics, pages 320–327. Association for Computational Linguistics, 2006.

[43] Jiri Hana, Anna Feldman, and Chris Brew. A resource-light approach to Russian morphology: Tagging Russian using Czech resources. In *EMNLP*, pages 222–229, 2004.

[44] Aurélie Herbelot and Eva Maria Vecchi. Building a shared world: mapping distributional to model-theoretic semantic spaces. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 22–32. Association for Computational Linguistics, 2015.

[45] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701, 2015.

[46] Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. Bootstrapping parsers via syntactic projection across parallel texts. *Natural language engineering*, 11(03):311–325, 2005.

[47] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[48] Jagadeesh Jagarlamudi, Hal Daumé III, and Raghavendra Udupa. Incorporating lexical priors into topic models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 204–213. Association for Computational Linguistics, 2012.

[49] Young-Bum Kim, Benjamin Snyder, and Ruhi Sarikaya. Part-of-speech taggers for low-resource languages using CCA features. In *Proceedings of the Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2015.

[50] Diederik P Kingma and Jimmy Ba Adam. A method for stochastic optimization. In *International Conference on Learning Representation*, 2015.

[51] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86, 2005.

[52] Stanislas Lauly, Hugo Larochelle, Mitesh Khapra, Balaraman Ravindran, Vikas C Raykar, and Amrita Saha. An autoencoder approach to learning bilingual word representations. In *Advances in Neural Information Processing Systems*, pages 1853–1861, 2014.

[53] Tao Lei, Yu Xin, Yuan Zhang, Regina Barzilay, and Tommi Jaakkola. Low-rank tensors for scoring dependency structures. In *Proceedings of the 52nd*

Annual Meeting of the Association for Computational Linguistics, volume 1, pages 1381–1391. Association for Computational Linguistics, 2014.

[54] Tao Lei, Yuan Zhang, Regina Barzilay, Lluís Màrquez, and Alessandro Moschitti. High-order low-rank tensors for semantic role labeling. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2015.

[55] Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*, 2016.

[56] Shen Li, Joao V Graça, and Ben Taskar. 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*, pages 1389–1398. Association for Computational Linguistics, 2012.

[57] Yujia Li, Kevin Swersky, and Richard Zemel. Generative moment matching networks. In *International Conference on Machine Learning*, pages 1718–1727, 2015.

[58] Chu-Cheng Lin, Waleed Ammar, Chris Dyer, and Lori Levin. Unsupervised POS induction with word embeddings. *arXiv preprint arXiv:1503.06760*, 2015.

[59] Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[60] Mingsheng Long, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. *arXiv preprint arXiv:1602.04433*, 2016.

[61] Ang Lu, Weiran Wang, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Deep multilingual correlation for improved word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics*, 2015.

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

[63] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[64] Gideon S Mann and Andrew McCallum. Generalized expectation criteria for semi-supervised learning of conditional random fields. 2008.

[65] Iain J Marshall, Joël Kuiper, and Byron C Wallace. Robotreviewer: evaluation of a system for automatically assessing bias in clinical trials. *Journal of the American Medical Informatics Association*, page ocv044, 2015.

[66] André FT Martins and Ramón Fernandez Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. *arXiv preprint arXiv:1602.02068*, 2016.

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

[68] Ryan McDonald, Slav Petrov, and Keith Hall. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 62–72. Association for Computational Linguistics, 2011.

[69] Ryan T McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith B Hall, Slav Petrov, Hao Zhang, Oscar Täckström, et al. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 92–97. Association for Computational Linguistics, 2013.

[70] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*, 2013.

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

[72] Eliakim H. Moore. On the reciprocal of the general algebraic matrix. *Bulletin of the American Mathematical Society*, 26(9):394–395, 1920.

[73] Tahira Naseem, Benjamin Snyder, Jacob Eisenstein, and Regina Barzilay. Multilingual part-of-speech tagging: Two unsupervised approaches. *Journal of Artificial Intelligence Research*, 36:341–385, 2009.

[74] Tahira Naseem, Harr Chen, Regina Barzilay, and Mark Johnson. Using universal linguistic knowledge to guide grammar induction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1234–1244. Association for Computational Linguistics, 2010.

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

[76] Jens Nilsson, Sebastian Riedel, and Deniz Yuret. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL shared task session of EMNLP-CoNLL*, pages 915–932, 2007.

[77] Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of LREC*, 2016.

[78] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[79] Liron Pantanowitz, Maryanne Hornish, Robert A Goulart, et al. Informatics applied to cytology. *Cytojournal*, 5(1):16, 2008.

[80] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the Empiricial Methods in Natural Language Processing*, volume 12, pages 1532–1543, 2014.

[81] Roger Penrose. A generalized inverse for matrices. In *Proceedings of the Cambridge Philosophical Society*, pages 406–413, 1955.

[82] Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086*, 2011.

[83] Audi Primadhanty, Xavier Carreras, and Ariadna Quattoni. Low-rank regularization for sparse conjunctive feature spaces: An application to named entity classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2015.

[84] Ariadna Quattoni, Borja Balle, Xavier Carreras, and Amir Globerson. Spectral regularization for max-margin sequence tagging. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1710–1718, 2014.

[85] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[86] Mohammad Sadegh Rasooli and Michael Collins. Density-driven cross-lingual transfer of dependency parsers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2015.

[87] Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.

[88] Ozan Sener, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese. Learning transferrable representations for unsupervised domain adaptation. In *Advances In Neural Information Processing Systems*, pages 2110–2118, 2016.

[89] Yusuke Shinohara. Adversarial multi-task learning of deep neural networks for robust speech recognition. *Interspeech 2016*, pages 2369–2372, 2016.

[90] Sameer Singh, Tim Rocktaschel, and Sebastian Riedel. Towards combined matrix and tensor factorization for universal schema relation extraction. In *NAACL Workshop on Vector Space Modeling for NLP*. Association for Computational Linguistics, 2015.

[91] Jason R Smith, Chris Quirk, and Kristina Toutanova. Extracting parallel sentences from comparable corpora using document level alignment. In *The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 403–411. Association for Computational Linguistics, 2010.

[92] Benjamin Snyder, Tahira Naseem, Jacob Eisenstein, and Regina Barzilay. Unsupervised multilingual learning for POS tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1041–1050. Association for Computational Linguistics, 2008.

[93] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 926–934, 2013.

[94] Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390*, 2015.

[95] Vivek Srikumar and Christopher D Manning. Learning distributed representations for structured output prediction. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 3266–3274, 2014.

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

[97] Oscar Täckström, Ryan McDonald, and Joakim Nivre. Target language adaptation of discriminative transfer parsers. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2013.

[98] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016.

[99] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the twenty-first international conference on Machine learning*, page 104. ACM, 2004.

[100] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[101] Ivan Vulic and Anna Korhonen. On the role of seed lexicons in learning bilingual word embeddings. ACL, 2016.

[102] Ivan Vulic and Marie-Francine Moens. 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*. Association for Computational Linguistics, 2015.

[103] Hongning Wang, Yue Lu, and Chengxiang Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 783–792. ACM, 2010.

[104] Hongning Wang, Yue Lu, and ChengXiang Zhai. Latent aspect rating analysis without aspect keyword supervision. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 618–626. ACM, 2011.

[105] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.

[106] Guillaume Wisniewski, Nicolas Pécheux, Souhir Gahbiche-Braham, and François Yvon. Cross-lingual part-of-speech tagging through ambiguous learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1779–1785, 2014.

[107] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ÅĄukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016. URL `http://arxiv.org/abs/1609.08144`.

[108] Min Xiao and Yuhong Guo. Distributed word representation learning for cross-lingual dependency parsing. *CoNLL-2014*, page 119, 2014.

[109] Huijuan Xu and Kate Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. *arXiv preprint arXiv:1511.05234*, 2015.

[110] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. Show, attend and

tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, page 5, 2015.

[111] Adam Yala, Regina Barzilay, Laura Salama, Molly Griffin, Grace Sollender, Aditya Bardia, Constance Lehman, Julliette M Buckley, Suzanne B Coopey, Fernanda Polubriaginof, J Garber, BL Smith, MA Gadd, MC Specht, and TM Gudewicz. Using machine learning to parse breast pathology reports. *Breast Cancer Research and Treatment*, 2016.

[112] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. *arXiv preprint arXiv:1511.02274*, 2015.

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

[114] Dong Yu, Li Deng, and Frank Seide. The deep tensor neural network with applications to large vocabulary speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 21(2):388–396, 2013.

[115] Mo Yu and Mark Dredze. Learning composition models for phrase embeddings. *Transactions of the Association for Computational Linguistics*, 3:227–242, 2015.

[116] Mo Yu, Tiejun Zhao, Yalong Bai, Hao Tian, and Dianhai Yu. Cross-lingual projections between languages from different families. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 312–317. Association for Computational Linguistics, 2013.

[117] Omar Zaidan, Jason Eisner, and Christine D Piatko. Using" annotator rationales" to improve machine learning for text categorization. In *HLT-NAACL*, pages 260–267. Citeseer, 2007.

[118] Daniel Zeman and Philip Resnik. Cross-language parser adaptation between related languages. In *Proceedings of the International Joint Conference on Natural Language Processing*, pages 35–42, 2008.

[119] Ye Zhang, Iain Marshall, and Byron C Wallace. Rationale-augmented convolutional neural networks for text classification. *arXiv preprint arXiv:1605.04469*, 2016.

[120] Yuan Zhang, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Greed is good if randomized: New inference for dependency parsing. In *EMNLP*, 2014.

[121] Guangyou Zhou, Zhiwen Xie, Jimmy Xiangji Huang, and Tingting He. Bi-transferring deep neural networks for domain adaptation. ACL, 2016.

[122] Will Y Zou, Richard Socher, Daniel M Cer, and Christopher D Manning. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the Empiricial Methods in Natural Language Processing*, pages 1393–1398, 2013.