Cross-lingual Transfer of Semantics in Low-resource Settings

by Maryam Aminian

B.Sc. in Software Engineering, May 2010, Iran University of Science and Technology
M.Phil. in Artificial Intelligence and Robotics, May 2012, Sharif University of Technology

A Dissertation submitted to

The Faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial satisfaction of the requirements
for the degree of Doctor of Philosophy

January 10, 2020

Dissertation directed by

Mona T. Diab
Professor of Computer Science
The School of Engineering and Applied Science of The George Washington University certifies that Maryam Aminian has passed the Final Examination for the degree of Doctor of Philosophy as of September 23, 2019. This is the final and approved form of the dissertation.

Cross-lingual Transfer of Semantics in Low-resource Settings

Maryam Aminian

Dissertation Research Committee:

Mona T. Diab, Professor of Computer Science, Dissertation Director

Abdou Youssef, Professor of Computer Science, Committee Member

Robert Pless, Professor of Computer Science, Committee Member

Ivan Titov, Associate Professor of School of Informatics, University of Edinburgh, Committee Member

Mirella Lapata, Professor of School of Informatics, University of Edinburgh, Committee Member
Acknowledgments

I would like to express my deepest gratitude to my advisor Professor Mona Diab for her invaluable guidance and continuing support during my PhD years. I would like to thank my proposal and dissertation committee members Professors Robert Pless, Abdou Youssef, Claire Monteleoni, Ivan Titov and Mirella Lapata for their insightful feedback and suggestions in this research. A special thanks to all my friends and lab mates at Care4Lang for their help and encouraging support during these years: Hanan Aldarmaki, Shabnam Tafreshi, Mahmoud Ghoneim, Mohamed ElBadrashiny, Sardar Hamidian, Ali Seyfi, Pedram Hosseini, Sawsan Alqahtani, Nada Almarwani, Fahad Ghamdi, Ayah Zirikly, Abdelati Hawwari, Heba Elfardy and all other friends who lifted my spirits one way or another.

The most special heartfelt thanks to my family, especially my beloved mother, Soheila, and my dearest brothers, Masoud and Hamid, for their endless love and tolerance during all these years. Although we were miles apart, they never let me feel alone and always stood by me. They are the most amazing teachers I have ever had in my life and I feel really grateful for having them. May Allah bless them all with His mercy.

And with deepest possible love, I would like to thank my best friend, collaborator and life companion Mohammad Sadegh for his unflagging love, unwavering support and encouragement during these years. This journey would have not been possible without him. And the most sincere thank you to the joy of my life, Ali, who taught me what real love means and how to fight for that.
Abstract

Cross-lingual Transfer of Semantics in Low-resource Settings

Despite the significant improvements yielded by aggregating supervised semantic analysis in various natural language processing applications, annotated data are available for only a few languages, mainly due to the significant costs of producing semantically annotated resources. Stark deficiency in semantically annotated resources for the majority of languages worldwide has led to a growing interest in transfer methods as a cost-effective complement or even alternative to for data annotation. The ultimate goal of transfer methods is to benefit from the semantic knowledge present for one or more language(s) with semantically annotated resources (high-resource) to model semantics in languages suffering from data deficiency. Annotation projection is one of the widely used approaches for transferring semantics from a high-resource language to a low-resource target language using the alignment links acquired from sentence-aligned corpora or bilingual dictionaries.

In this dissertation, we demonstrate the effectiveness of annotation projection for producing semantic analysis in a spectrum of low-resource scenarios ranging from the case that the only missing part of the puzzle is semantic annotations to the extreme low-resource scenario where no adequate explicit lexico-syntactic features are available. In this study, we target cross-lingual semantic transfer on both the lexical and sentential levels. On the lexical level, we propose an unsupervised system based on annotation projection to address the phenomenon of word sense divergence, mainly observed when the underlying semantic distribution of the test set is different from that of the train data. We extrinsically evaluate our lexical transfer model in an SMT framework, as one of the NLP applications heavily impacted by the words with sense divergence. We demonstrate that our proposed model for identifying and disambiguating words with sense divergence improves SMT lexical choice. Our method solely relies on the transferred word sense information and does not utilize any labeled or in-domain training data.
On the sentential level, we begin with a cross-lingual semantic role labeling (SRL) model that mainly focuses on improving the quality of projection instances used to train the model by taking advantage of cues automatically acquired from word alignments as well as syntactic analysis of the sentence. We devise a customized cost function to effectively weight some projections over other instances. We show that utilizing this simple cost function yields significant improvements over a standard annotation projection method on English to German. We then move to a more realistic low-resource scenario where accurate linguistic features or sizable parallel corpora might not be readily available. We first demonstrate the power of character representations to confront the need for other morpho-syntactic features. We additionally look into a cross-lingual SRL model that uses the Bible, a smaller but widely available parallel corpus and analyze the effectiveness of conventional transfer techniques when applied on smaller projection corpora.

Finally, we explore the role of supervised syntactic information on the performance of a cross-lingual semantic dependency parsing (SDP) model built over projections in a multitask framework. We report the performance of various multitasking models on a subset of projections with different densities to find the optimal level of supervision required by each framework. We show empirically that multitask learning yields significantly better performance in annotation projection models compared to supervised baselines.
# Table of Contents

Acknowledgments ........................................ iv

Abstract of Dissertation .................................. v

List of Figures ........................................... xi

List of Tables ........................................... xiii

1. Introduction ........................................ 1

2. Background ......................................... 7
   1 Lexical Semantics .................................. 7
   2 Semantic Role Labeling ............................ 8
      2.1 Available SRL Resources ...................... 9
      2.2 Dependency-based SRL ....................... 11
   3 Broad-coverage Semantic Dependency Parsing .... 13
   4 Low-resource Natural Language Processing ...... 14
   5 Parallel Data Notations ............................ 17
   6 Annotation Projection .............................. 19

3. Word Sense Divergence ................................ 20
   1 Introduction ...................................... 20
   2 Transfer-based Approach .......................... 22
      2.1 Sense OOD Identifier ......................... 22
      2.1.1 Generating Synthetic Train Data for Sense OOD Identifier .... 23
      2.2 Sense OOD Disambiguation ................... 24
   3 Knowledge-oriented Approach ..................... 25
   4 SMT Experiments ................................... 27
   5 Evaluation Results ................................ 29
5.1 Transfer-based Approach ............................................. 30
5.2 Knowledge-oriented Approach ................................. 32
6 Analysis and Discussion ........................................... 33
7 Related Work .......................................................... 35
8 Conclusion ............................................................... 39
4. Multilingual Correspondence Learning ......................... 41
1 Introduction ............................................................ 41
2 Our Model ............................................................... 42
3 Experiments ............................................................. 44
4 Results .................................................................... 47
5 Discussion ............................................................... 49
6 Related Work .......................................................... 54
7 Conclusion ............................................................... 55
5. Cross-Lingual SRL with Limited Resources .................... 56
1 Introduction ............................................................ 56
2 Baseline Model .......................................................... 57
3 Model Improvements .................................................. 58
3.1 Bootstrapping ......................................................... 59
3.2 Data-dependent Cost Function ................................. 59
4 Experiments ............................................................. 62
5 Results .................................................................... 64
6 Analysis ................................................................. 65
6.1 Effect of POS Filter ................................................. 65
6.2 Effect of Relabeling ................................................. 66
7 Related Work .......................................................... 68
<table>
<thead>
<tr>
<th></th>
<th>Analysis</th>
<th>103</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Related Work</td>
<td>104</td>
</tr>
<tr>
<td>8</td>
<td>Conclusion</td>
<td>107</td>
</tr>
<tr>
<td>8</td>
<td><strong>Conclusion</strong></td>
<td>108</td>
</tr>
<tr>
<td></td>
<td><strong>Bibliography</strong></td>
<td>113</td>
</tr>
</tbody>
</table>
# List of Figures

| Figure 2.1 | Frame semantics annotation example from FrameNet 1.5 | 10 |
| Figure 2.2 | FrameNet annotation compared to the PropBank annotation | 11 |
| Figure 2.3 | Abstract Meaning Representation example | 11 |
| Figure 2.4 | Different SDP target representations | 13 |
| Figure 2.5 | Bi-lexical semantic dependencies in PropBank compared to DM representation | 15 |
| Figure 2.6 | Example of projecting bi-lexical semantic dependencies in SRL and SDP | 18 |
| Figure 3.1 | Block diagram of the knowledge-oriented sense OOD identification and disambiguation model | 26 |
| Figure 5.1 | The iterative bootstrapping algorithm used in our cross-lingual SRL model | 60 |
| Figure 5.2 | Example of parallel sentences with their syntactic and semantic dependencies | 63 |
| Figure 5.3 | Example of syntactic dependency match in SRL annotation projection | 63 |
| Figure 5.4 | Precision, recall and F1–score during relabeling iterations | 67 |
| Figure 5.5 | Relabeling result for sentence shown in Figure 5.3 | 68 |
| Figure 6.1 | Graphical depiction of our joint argument identification and classification model | 76 |
| Figure 6.2 | The effect of projection data size on the performance of the CModel for different languages | 83 |
| Figure 7.1 | Graphical depiction of our supervised SDP model | 94 |
| Figure 7.2 | MTL model with shared FNN layer | 95 |
| Figure 7.3 | MTL model with task-specific FNN layer | 96 |
| Figure 7.4 | Labeled and unlabeled F1–scores of different SDP models on the in-domain and out-of-domain Czech test sets | 99 |
Figure 7.5  Performance of different MTL models after adding BERT features on the in-domain and out-of-domain test sets .......................... 101

Figure 7.7  Percentage of improved semantic dependencies obtained from the best performing MTL setup for different syntactic relations .................. 104

Figure 7.8  Percentage of degraded semantic dependencies resulted from the best performing MTL setup for different syntactic relations .................. 105
List of Tables

Table 2.1  Set of widely recognized semantic roles ........................................ 9
Table 2.2  Two senses of the verb *decline* and their frame set from PropBank. .... 12
Table 3.1  Empirical results for our transfer-based SOOD identification and disambiguation method ................................................................. 29
Table 3.2  Example of correct sense OOD identification and replacement with non-improving BLEU score ......................................................... 32
Table 3.3  Empirical results for our knowledge-oriented sense OOD identification and disambiguation method ......................................................... 33
Table 3.4  Percentage of BLEU enhanced/degraded sentences ......................... 33
Table 3.5  Translation examples produced by our transfer-based approach .......... 34
Table 4.1  Evaluation of our multilingual correspondence learning method against BabelNet and Tharwa ................................................................. 48
Table 4.2  Evaluation of the dialectal component of our correspondence learning method ................................................................. 49
Table 4.3  Mapping between MSA POS tags and EGY POS tags ..................... 50
Table 4.4  Evaluation of our correspondence learning method with different POS constraints ................................................................. 51
Table 4.5  Examples of EGY candidates generated by our multilingual correspondence learning model ................................................................. 53
Table 5.1  Empirical results of our cost-sensitive training method for transferring SRL from English to German ......................................................... 65
Table 5.2  POS equivalent classes ................................................................. 66
Table 5.3  Empirical results of our cost-sensitive training method for transferring SRL from English to German after applying POS filter ................. 66
Table 6.1  Size of the Europarl projected data for different languages ............... 79
Table 6.2  Empirical results of the CModel for different Europarl languages .......... 80
Table 6.3 Results of the CModel and SLem models on the CoNLL2009 in-domain test set compared to other supervised systems .......................................................... 82

Table 6.4 Size of the Bible projected data for different languages ........................................ 88

Table 6.5 List of CModel parameters ................................................................................. 88

Table 7.1 Our SDP model parameters ................................................................................. 97
Chapter 1: Introduction

Semantics is the study of meaning communicated through language, namely, investigating how meaning is created and transmitted in a language. Notwithstanding the significant improvements yielded by aggregating supervised semantic analysis systems in a wide range of downstream Natural Language Processing (NLP) applications (e.g. statistical machine translation (SMT) (Klein et al., 2017; Johnson et al., 2017; Lee et al., 2017), information extraction (Khot et al., 2017; Meng et al., 2017) and Question Answering (Shen and Lapata, 2007a)), many languages still suffer from limited availability of semantically annotated data with adequate size and quality that can be used to train supervised systems. One conventional approach is to build semantically annotated resources however this typically requires significant amount of time and money. A cost-effective alternative to data annotation is transfer approaches that ultimately aim to use the semantic knowledge already annotated for a (many) high-resource language(s) to model semantics in low-resource languages.

Semantic knowledge is transferred through different means; one way is to directly adapt the model trained on the source language to make it applicable to the target language. The basic module of direct model transfer that facilitates learning is the shared feature space provided through cross-lingual features such as cross-lingual embeddings or word clusters. Another approach is to project semantic annotations from a high-resource language to a low-resource one through alignments between the words in the source (high-resource) and target (low-resource) languages, acquired via sentence-aligned corpora or bilingual dictionaries. In this dissertation, we aim to build enhanced semantic analysis models for languages that suffer from a lack in annotated data through annotation projection. We demonstrate the effectiveness of annotation projection for producing semantic analysis in a range of low-resource scenarios. We cover the spectrum from the case where the only missing part of the puzzle is semantic annotations to the extreme low-resource scenario where no adequate
explicit lexico-syntactic features are available. We propose different cross-lingual semantic transfer models on both the word and the sentence levels. At the word-level, we specifically target the problem of word sense divergence in an SMT framework. At the sentence-level, we explore different aspects of building and enhancing a cross-lingual semantic role labeling model as well as a semantic dependency parsing model using annotation projection under different low-resource conditions.

At the word-level, we target the phenomenon of word sense divergence which is one of the main sources of performance degradation in many supervised systems. Word sense divergence happens when the sense of a particular word in the input test sentence at inference varies from all observed senses of that word in the training data. Words with sense divergence are substantially observed when the underlying semantic distribution of the test set is different from that of the training data. The presence of words with sense divergence is prominently highlighted in informal spoken varieties of a language, or dialects. Dialects typically share a set of cognates that could bear the same meaning in both varieties or only be shared homographs/homophones with sense divergences. This problem is exacerbated when the dialectal usage of a language coexists with the standard variety in the form of code-switching. The ubiquitous presence of dialectal language in informal media on one hand, and stark deficiency of linguistic resources for dialects on the other hand, further highlights the need for transfer models to address the dialectal sense divergence phenomenon. As part of our efforts in this vein, we propose a cross-lingual model for identifying and disambiguating words with sense divergence using lexical semantic knowledge transferred from another language through parallel corpora. We extrinsically evaluate our lexical transfer model in an SMT framework as one of the NLP applications where its performance is heavily impacted by word sense divergence. We further compare our transfer approach with a knowledge-oriented method that utilizes a range of linguistic tools for sense divergence identification and disambiguation. It is worth noting that disambiguation in the mentioned approaches is limited to word senses observed in the training data. To expand the set of correspondents,
we describe a multilingual correspondence learning framework that uses monolingual and cross-lingual information transferred through a bridge language to the target side, thereby building augmented lexical resources expanding multilingual dictionaries. We particularly focus on extracting dialectal Arabic and English correspondents motivated by the ubiquitous presence of dialectal language in informal Arabic social media.

Performance of semantic models built using annotation projection is often dominated by the considerable amount of noise present in projection. The noise is usually caused by different factors including but not limited to erroneous alignments, noise in the source annotations and divergences in word order, as well as semantic shifts and drifts between the source and target languages. As part of our efforts to tackle this aspect of cross-lingual semantic transfer at the sentence-level, we propose a Semantic Role Labeling (SRL) model that improves the quality of projection instances used to train the model. We use translation cues that are automatically acquired from word alignments. We initially demonstrate that exclusively relying on partially projected data does not yield good performance. Accordingly, we propose a weighting algorithm to improve annotation projection based on cues obtained from the syntactic analysis of the sentence together with translation information. Our approach is an alternative to the traditional manually-defined rules used to filter noisy projections in previous studies. We propose a customized objective function especially designed to train the model over projected instances.

Besides all challenges posed by the lack of data, many supervised semantic analysis systems are based on the assumption that accurate linguistic features such as lemma, POS tags, and syntactic analyses are readily available for the target language. Thus, we face a substantial performance degradation when it comes to a truly low-resource language where these resources and tools are hardly available. To remedy this, we move one step forward in the spectrum of low-resource assumptions and assume that the only available resource to build an SRL system for the target language is parallel data that is used to provide links from a high-resource language to a target language. In the process, we demonstrate the
power of character representations to compensate for other morpho-syntactic features. Since projected annotations often comprise noisy and partial labels, performance of the target SRL model built using these projections heavily depends on the size of the parallel data used for annotation projection. Lack of access to a sizable parallel corpus (on the scale of Europarl) for majority of languages motivated us to look into a cross-lingual SRL model that uses a smaller but widely available parallel corpus (such as the Bible). We initially analyze the effectiveness of conventional transfer techniques when applied on the smaller projection corpus. We then mitigate the need for large parallel corpora by leveraging two direct transfer techniques in our annotation projection model namely polyglot training (Mulcaire et al., 2018) and employing cross-lingual embeddings. We empirically demonstrate the efficacy of our enhanced SRL model when applied on smaller projection corpora.

The close connection between semantic analysis of a sentence and its underlying syntactic structure has led previous studies to consider syntax as an effective auxiliary task in multitask learning frameworks, where relevant syntactic features help the semantic model focus its attention on the features that matter the most. We argue that multitasking with syntax would be even more effective on models trained on noisy and partial projected annotations. In other words, we expect that the inductive bias provided by syntactic analyses leads to a more powerful SDP model in filtering noisy instances and focusing on the relevant features. As one of our efforts in this vein, we propose a cross-lingual Semantic Dependency Parsing (SDP) model based on annotation projection that aggregates syntax in a multitask learning framework. We empirically demonstrate that multitask learning yields significantly better performance in annotation projection models compared to supervised baselines which further highlights its effectiveness when applied in low-resource settings.

The main contributions of this dissertation can be summarized as follows:

- We developed an unsupervised model for identifying and disambiguating words with sense divergence using lexical semantic evidence provided by parallel corpus and demonstrated the efficacy of transfer approaches to improve the SMT lexical choice
for words with sense divergence;

- We devised a pivoting algorithm that can be utilized to expand the multilingual correspondents used for sense divergence disambiguation. Additionally, our multilingual correspondence learning model can be used for automatic augmentation of multilingual lexica and is capable of detecting inconsistencies in the lexicon entries and possibly providing or suggesting candidates to replace them.

- We developed a cross-lingual SRL model based on annotation projection that particularly focuses on identifying noisy projections during the training phase by utilizing a customized objective function sensitive to the noise. We demonstrated that our model yields in particular improvements in filtering noisy training instances and overall performance of the target SRL model.

- We demonstrated the power of character representations to compensate for other morpho-syntactic features by building a cross-lingual SRL model based on annotation projection that is agnostic to linguistic features. We further explored the effectiveness of conventional transfer techniques when applied on the smaller projection corpus by using the Bible for projecting annotations.

- We also explored the efficacy of using multitask learning framework in low-resource settings by aggregating supervised syntactic information in a cross-lingual SDP model and demonstrated the particular power of multitask learning to improve the target SDP model in truly low-resource settings where (enough) in-domain training data might not be available.

This thesis is outlined as follows: In Chapter 2, we provide a brief overview of the background knowledge, common notations, and definitions used throughout this thesis. In Chapter 3, we describe our annotation projection model for addressing sense divergence in a dialectal SMT model. In Chapter 4, we present our multilingual correspondence learning
model that aims to enhance the sense divergence disambiguation module. In Chapter 5, we present our work on improving the quality of annotation projection by using translation cues automatically discovered from syntax and word alignments. In Chapter 6, we explain our cross-lingual model to build an SRL system from raw text. In Chapter 7, we describe our cross-lingual SDP model and demonstrate the particular effectiveness of multitasking in low-resource settings. Finally in Chapter 8, we summarize the main findings of this thesis and list our research contributions and limitations. Each chapter contains a dedicated related work section to survey the related studies, discuss our contributions and how they relate to the existing literature, to the best of our knowledge, at the time of their publication.

It is worth noting that the approach, experiments and analysis presented in Chapter 3 are modified and reproduced from two publications; Sections 2, 4 and 5.1 are almost entirely reproduced from (Aminian et al., 2015)\(^1\). Sections 3 and 5.2 are modified from (Aminian et al., 2014). Chapter 4 is the modified version of the content presented in (Aminian et al., 2016). Sections 1 to 5 of Chapter 5 are modified from (Aminian et al., 2017). All experiments and analysis in Section 6 of Chapter 5 are original contributions published here for the first time. Sections 1 to 3.3 of Chapter 6 are reproduced from (Aminian et al., 2019). Experiments and results presented in Section 4 of Chapter 6 as well as the content described in Chapter 7 are original and published here for the first time.

---

\(^1\)We recognize that state of the art in MT currently is based on neural MT. At the time of conducting the research, SMT was the community state of the art. Moreover, our work (Aminian et al., 2015) received the best paper award for advances in MT using lexical semantics and deep language processing in SSST-09 at the time of its publication.
Chapter 2: Background

In this chapter, we provide some background information on lexical semantics where word sense divergence is one of the most important challenges in lexical semantics. We then give an overview of sentential semantic analysis in the form of (a) semantic role labeling and (b) broad-coverage semantic dependency parsing. Moreover, we briefly review the NLP approaches employed in the low-resource settings and explain some notations and definitions that we use throughout this thesis including parallel data and annotation projection.

1 Lexical Semantics

The ultimate goal of semantics is to investigate how meaning is created and transmitted in a language. Meaning of an utterance can be understood by the meaning of its components and the way in which they are combined. Morphemes are the smallest units of meaning in a language that cannot be divided into smaller meaningful parts. Morphemes that are space delimited are expressed as words. Words and their meaning that are studied under lexical semantics are considered the core components responsible for meaning transmission in different lexical semantic theories (Allan, 1986; Dowty, 1986; Jackendoff, 1992). Meaning of a word, commonly known as word sense, is usually determined by the context in which it appears; for instance, the word foot can have different meanings, or be polysemous, as in the following sentences (among others):

- His left foot was severely injured during the game.
- They ordered a foot-long hot dog at the restaurant.
- She decided to set up the tent at the foot of the mountain that night.

In the context of NLP and statistical learning models, word sense divergence is a linguistic phenomenon that particularly occurs when the sense of a word in the input test sentence at
inference varies from all observed senses of that word in the training data. Words with sense divergence, or sense out-of-domain words (sense OOD), are frequently observed when the underlying distribution of the test set is different from that of the training data. Thus, any type of domain shift leads to an, oftentimes significant, increase in sense out-of-domain words. All polysemous words can potentially lead to sense divergence issue when their sense at inference varies from all their observed senses in training.

2 Semantic Role Labeling

The explicit modeling of syntactic analysis of a sentence has been shown to have a major effect in almost all (up/down-stream) NLP applications. However, modeling syntax alone is not sufficient to represent the full meaning of a sentence. In particular, many semantic components that are involved in a condition or event can not be adequately determined alone by their syntactic roles (subject, object, etc.) in a sentence. For instance, some transitive verbs can have their Theme (the entity that is moving or being located) in both subject and object positions. The following sentences show examples of transitive verb play. In both sentences, Taps acts as a theme for play but occurs in two different syntactic positions (examples from (Palmer et al., 2010)):

- The sergeant played taps.
- Taps played quietly in the background.

Semantic role labeling (SRL) is the task of automatically labeling predicates and arguments of a sentence with shallow semantic labels characterizing “Who did What to Whom, How, When and Where?” (Palmer et al., 2010). Table 2.1 shows some of the widely recognized semantic roles that is common across all different SRL formalisms (Saeed, 1997). The following sentence shows the usage of the listed semantic roles in Table 2.1 that make up the semantic components responsible for meaning representation for the verb sold.
<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent</td>
<td>Initiator of action, capable of volition</td>
<td><em>The batter</em> smashed the pitch into left field. <em>The pilot</em> landed the plane as lightly as a feather.</td>
</tr>
<tr>
<td>Patient</td>
<td>Affected by action, undergoes change of state</td>
<td>David trimmed <em>his beard</em>. John broke <em>the window</em>.</td>
</tr>
<tr>
<td>Theme</td>
<td>Entity moving, or being “locate”</td>
<td>Paola threw <em>the Frisbee</em>. <em>The picture</em> hangs above the fireplace.</td>
</tr>
<tr>
<td>Experiencer</td>
<td>Perceives action but not in control</td>
<td><em>He</em> tasted the delicate flavor of the baby lettuce. <em>Chris</em> noticed the cat slip through the partially open door.</td>
</tr>
<tr>
<td>Beneficiary</td>
<td>For whose benefit action is performed</td>
<td>He sliced <em>me</em> a large chunk of prime rib, and I could hardly wait to sit down to start in on it. <em>The Smiths</em> rented an apartment <em>for their son</em>.</td>
</tr>
<tr>
<td>Instrument</td>
<td>Intermediary/means used to perform an action</td>
<td>He shot the wounded buffalo with <em>a rifle</em>. <em>The surgeon</em> performed the incision with <em>a scalpel</em>.</td>
</tr>
<tr>
<td>Location</td>
<td>Place of object or action</td>
<td>There are some real monsters hiding in <em>the anxiety closet</em>. <em>The band</em> played on <em>the stage</em>.</td>
</tr>
<tr>
<td>Source</td>
<td>Starting point</td>
<td><em>The jet took off from Nairobi</em>. <em>We heard the rumor from a friend</em>.</td>
</tr>
<tr>
<td>Goal</td>
<td>Ending point</td>
<td><em>The ball rolled to the other end of the hall</em>. <em>Laura lectured to the class</em>.</td>
</tr>
</tbody>
</table>

Table 2.1: Set of widely recognized semantic roles (Palmer et al., 2010).

\[ Agent(Source) Esau sold Theme his birthright Goal to Jacob Meaning for nothing \]

2.1 Available SRL Resources

There are two main English lexical resources which are explicitly annotated with semantic roles: FrameNet (Baker et al., 1998) and PropBank (Palmer et al., 2005). Moreover, Abstract Meaning Representation (AMR) (Banarescu et al., 2013a) is another resource with semantic role annotations that has been extensively built upon the PropBank annotation.
FrameNet  Based on Fillmore’s Frame Semantics (Baker et al., 1998), each semantic frame in FrameNet is defined using its frame elements (which are essentially equivalent to traditionally defined semantic roles shown in Table 2.1). For instance the semantic frame Apply-heat in FrameNet corresponds to the following frame elements: a cook (agent), a food (theme) and heating instrument (instrument) and involves some lexical units as its members such as: bake, barbecue, blanch, boil, broil, brown, etc. Figure 2.1 shows a sentence from FrameNet 1.5 that contains three semantic frames: Motion-Directional frame associated with the lexical unit –the predicate – fell, Locative-Relation frame associated with the preposition in and the Quantity frame corresponding to the article a few.

PropBank  In contrast to FrameNet, the primary goal of PropBank is to provide annotated data that can be used for training supervised SRL systems. PropBank semantic roles are purposely chosen to be quite generic and theory-neutral: Arg0, Arg1, etc. Due to the issues encountered by defining a universal set of semantic roles covering all types of predicates, PropBank defines semantic roles on a verb by verb basis. For a particular verb, Arg0 generally denotes the prototypical Agent and Arg1 is the prototypical Patient or Theme. Although lower numbered arguments are generally interpreted in a common way across different predicates and verbs, there is no consistent generalization across higher numbered arguments such as Arg4 and Arg5. Table 2.2 represents the verb-specific definition of PropBank semantic roles for two senses of the verb decline. Also, Figure 2.2 clarifies differences between FrameNet and PropBank annotation for the verb bought in a single sentence.
In the realm of SRL dataset, thereby methods, there are two main paradigms for representing semantic arguments: span-based arguments and dependency-based arguments. In some span-based SRL datasets (Palmer et al., 2005; Hovy et al., 2006), argument spans correspond to the syntactic constituents. So arguments spans can be directly converted to dependencies by applying syntactic head rules heuristics (Collins, 2003). However, such conversion from spans to dependencies and vice versa might not be applicable to other
Table 2.2: Two senses of the verb *decline* and their frame set from PropBank.

<table>
<thead>
<tr>
<th>Frameset</th>
<th>Arguments</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>decline.01 (go down incrementally)</td>
<td><strong>Arg1</strong>: entity going down, <strong>Arg2</strong>: amount gone down by EXT, <strong>Arg3</strong>: start point, <strong>Arg4</strong>: end point</td>
<td><strong>Arg1</strong>: its income declining <strong>Arg2</strong>: EXT 42% <strong>Arg4</strong>: to 121 million <strong>ArgM-TMP</strong>: in two months</td>
</tr>
<tr>
<td>decline.02 (demur, reject)</td>
<td><strong>Arg0</strong>: agent, <strong>Arg1</strong>: rejected thing</td>
<td><strong>Arg0</strong>: The spokesman declined <strong>Arg1</strong>: to elaborate</td>
</tr>
</tbody>
</table>

semantic argument representations (Hajič et al., 2012; Banarescu et al., 2013b; Baker et al., 1998; He et al., 2015). In this thesis, we utilize dependency-based SRL annotation and resources.

**Definitions** In dependency-based SRL, the goal is to find arguments along with their roles for each predicate in a sentence. Formally, in sentence $x = [w_1, \cdots, w_n]$ with $n$ words and $m$ predicates that belong to $\mathbb{P} = [(p_i, \psi_i); 1 \leq p_i \leq n]_{i=1}^m$ where $\psi_i$ is the sense of the predicate with index $p_i$, we should find semantic dependencies between each word in the sentence with respect to each predicate:

$$\mathbb{L}_x = \{(p_i \xrightarrow{r} j | \psi_i); 1 \leq j \leq n, p_i \in \mathbb{P}\}$$

where $r$ is the role of the $j$th word as an argument for the predicate word $w_{p_i}$. In cases where a word is not an argument, $r$ would be equal to NULL. Evaluation of system output is conducted on semantic dependencies $(p_i \xrightarrow{r} j | \psi_i)$; thus the SRL system should find predicate senses as well as the argument roles. During training, these dependencies are used as training instances for a machine learning algorithm. Most classic SRL systems decompose this task into a pipeline of steps including: predicate sense disambiguation, argument identification, and argument classification (Björkelund et al., 2009; Roth and Lapata, 2016; Marcheggiani et al., 2017).
Balcor, which has interests in real estate, said the position is newly created

(a) DM

Balcor, which has interests in real estate, said the position is newly created

(b) PAS

Balcor, which has interests in real estate, said the position is newly created

(c) PSD

Figure 2.4: Different SDP target representations: (a) DM, (b) PAS, and (c) PSD.

3 Broad-coverage Semantic Dependency Parsing

Broad-coverage semantic dependency parsing (SDP) (Oepen et al., 2014, 2015; Che et al., 2016) aims to discover the underlying semantic relation existing between all content words in the sentence. In this semantic representation scheme, each word in the sentence might be the semantic argument of multiple predicates. Therefore, unlike dependency parsing that integrates labeled bi-lexical dependencies as a tree, SDP forms a directed acyclic graph over syntactic dependencies. SDP yields a labeled directed graph whose vertices are tokens in the sentence. So in this sense, it is considered similar to dependency-based SRL. Contrary to SRL, however, where the main focus is on verbal and nominal predicates, SDP produces a semantic analysis for all words in the sentence.
The SDP dataset introduced in the SemEval shared tasks (Oepen et al., 2014, 2015; Che et al., 2016) consists of three main semantic formalisms, also known as target representations, slightly differ in the way they determine and label semantic relations. These representations are usually referred to as DM (DELPH-IN minimal recursion), PSD (Prague semantic dependencies) and PAS (Enju predicate–argument structures). These formalisms consist of labeled directed graphs between content words in the sentence. In general, the node with an outgoing edge corresponds to the predicate, likewise, the node with an incoming edge shows the semantic argument. Analogously, edge labels are equivalent to the semantic roles in semantic role labeling. Each sentence has a unique top node that shows the main predicate of the sentence. Figure 2.4 shows a sentence from SemEval 2015 data along with its bi-lexical semantic dependencies in DM, PAS and PSD representations.

Besides differences in the way each semantic formalism determines and labels semantic relations, the main difference between bi-lexical semantic dependencies in SDP and other semantic representation formalisms such as PropBank/FrameNet SRL and AMR lies in the range of semantic phenomena that are addressed in each model. Some semantic phenomena such as comparatives, possessives and various types of modification that convey the sentence meaning are not addressed/handled/targeted in the PropBank or AMR representations, while being extensively analyzed in the DM representation. Figure 2.5 further demonstrates the difference between PropBank (SRL) and DM (SDP) bi-lexical semantic dependencies.

4 Low-resource Natural Language Processing

In the last decade, various supervised natural language processing systems have shown great performance, thereby gaining popularity among researchers, however, the performance of these systems heavily relies on availability of high quality annotated data. Since creating annotated data costs time and effort, many languages still lack annotated data. Consequently, no supervised NLP systems are developed for these languages. The data scarcity problem is exacerbated for (supervised) systems when facing the challenge of new domains or language
A similar technique is almost impossible to apply to other crops

(a) Semantic dependencies in PropBank.

A similar technique is almost impossible to apply to other crops

(b) DELPH-IN Minimal Recursion Semantics–derived bi-lexical dependencies (DM)

Figure 2.5: Bi-lexical semantic dependencies in (a) PropBank and (b) DM representation.

variants. To mitigate data deficiency in low-resource scenarios, various methods have been used:

- **Unsupervised Methods**: The main goal of unsupervised NLP methods is to learn a model without using any annotated data relevant to the task at hand. There has been considerable amount of work on employing unsupervised techniques for building different NLP applications such as syntactic parsers (Spitkovsky et al., 2013; Le and Zuidema, 2015) and semantic role labelers (Lang and Lapata, 2010; Garg and Henderson, 2012; Luan et al., 2016). Notwithstanding the substantial progress of unsupervised systems in the last decade, their performance still lags considerably behind supervised systems.

- **Semi-supervised Methods**: Semi-supervised methods are a class of machine learning techniques which use a small amount of labeled data together with large amounts of unlabeled data for training an NLP system. Most semi-supervised learning strategies extend the methods used in either unsupervised or supervised learning models. Self-training or bootstrapping methods (McClosky et al., 2006) are one of the most commonly used semi-supervised techniques in low-resource NLP settings. In self-training, the model utilizes its own predictions to teach itself to improve via an iterative
learning process. Semi-supervised data augmentation oftentimes leads to improved performance of the model at each iteration. However, problem formulation and data characteristics play a major role in selecting the semi-supervised learning method suitable for a specific task. In fact, unlabeled data can lead to worse performance with the wrong link assumptions when a semi-supervised method is blindly applied (Zhu and Goldberg, 2009).

• **Transfer Methods:** Transfer approaches are another class of NLP methods that mainly aim to transfer the linguistic knowledge embedded in a rich-resource scenario to a low-resource scenario through different means. Transfer methods typically devise one of the following approaches or a combination of them:

(a) Learning word correspondences through links provided by either parallel or comparable corpora; these links are then used to transfer supervised annotations from the source language/domain to the target language/domain (Padó and Lapata, 2005; Akbik et al., 2015; Kozhevnikov and Titov, 2013a). Word correspondences can be based on the contextual usage of a particular word in the source and target sentences or simply ignore the context.

(b) Direct model transfer which aims to adapt the model trained on the source language/domain to make it directly applicable to the target language/domain (Täckström et al., 2013; Rosa and Žabokrtský, 2015). The backbone of direct model transfer approaches in low resource language scenarios that facilitates learning is cross-lingual features/embeddings that map words from distinct vocabularies, here tokens of different languages, to a shared representation space. This provides the opportunity for a cross-lingual model to learn from shared or common linguistic and statistical patterns in different languages.

• **Representation Learning from Language Models (LMs):** The main goal of representation learning from LMs, recently proposed in different models (e.g. ELMO
(Peters et al., 2018), BERT (Devlin et al., 2019), XML (Lample and Conneau, 2019), etc.), is to facilitate learning from large amounts of unlabeled data to improve existing NLP benchmarks in two forms: (a) pre-trained LM features that can be used to extract contextualized representations of words. These representations can be later utilized in the form of features in the model; (b) Task-specific fine-tuning of the pre-trained models on a specific task to produce state of the art predictions.

In this thesis, we primarily focus on using transfer methods for building semantic analysis models for languages that suffer from a lack in annotated data through annotation projection. We assume that we have access to parallel corpora that can be used for learning word correspondences across the source and target languages, thereby facilitating projections. The usage of representation learning techniques is complementary to our work. We explored its effectiveness in our cross-lingual SDP model in Chapter 7.

5 Parallel Data Notations

We show parallel data as $\mathcal{P} = [(s^{(1)}, t^{(1)}), \ldots, (s^{(n)}, t^{(n)})]$ such that each sentence $s^{(i)}$ is a translation of sentence $t^{(i)}$. Assuming that $s^{(i)} = [s_1^{(i)}, \ldots, s_l^{(i)}]$ and $t^{(i)} = [t_1^{(i)}, \ldots, t_l^{(i)}]$, we use an automatic word alignment system to obtain one-to-one word alignments with the following definitions:

1. $0 \leq a_{k,j}^S \leq l_k$ as the index of the source word that is aligned to the $j$th word in the $k$th target sentence, where $A_{k,j}^S = 0$ indicates a missing alignment.

2. $a_{k,i}^T \in \{t_1^{(k)}, \ldots, t_{l_k}^{(k)}\}$ as the target word that is aligned to the $i$th word in the $k$th source sentence. We use a special NULL notation to indicate missing alignment.

Note that we use $a_{k,j}^S$ to indicate the source word index while $a_{k,i}^T$ shows the target word form.
Figure 2.6: (a) Example of SRL annotation projection for an English-German sentence pair from the Europarl corpus (Koehn, 2005). (b) Example of SDP annotation projection for an English-Czech sentence pair from the SemEval2015 corpus (Oepen et al., 2015). Supervised predicate-argument structure of the English sentences (edges on top) are generated using the supervised SRL/SDP systems. Dashed lines in the middle show intersected word alignments from Giza++ (Och and Ney, 2003). Dashed edges at the bottom show the projected predicate-arguments.
6 Annotation Projection

We assume that the source sentences in \( \mathcal{P} \) belong to a rich-resource language for which accurate annotated resources and tools (required by the task of interest) are available. In contrast, the target sentences belong to a low-resource target language without accurate annotated data and tools such as semantic roles, semantic dependencies, syntactic dependency parses, part-of-speech tags, word senses, or even lemma information.

In this thesis, we explore two sentential semantic parsing tasks: semantic role labeling and broad-coverage semantic dependency parsing. Both of these tasks aim to predict a labeled bi-lexical semantic dependency between a predicate and its arguments. Hence, regardless of differences between these tasks, annotation projection transfers the labeled bi-lexical dependency from the source side to the target side.

For every source sentence \( s^{(i)} \), we run a supervised SRL (SDP) system to obtain the supervised argument structure \( L_{s^{(i)}} \). Additionally, we use an automatic word alignment system such as Giza++ (Och and Ney, 2003) to obtain one-to-one word alignments. We use the following conditions to project a bi-lexical semantic dependency from a source sentence to a target sentence:

\[
(p \rightarrow m|y) \in L_{s^{(i)}} \Rightarrow \text{add } (p \rightarrow m|y) \text{ to } L_{t^{(i)}}
\]

where \( L_{s^{(i)}} \) is the supervised argument structure and \( L_{t^{(i)}} \) is the projected argument structure for the \( i \)th sentence. We assume that there is a universal predicate sense that is common across languages (this is the case in the recently published Universal Proposition Bank (Wang et al., 2017)). Figures 2.6a and 2.6b show SRL and SDP annotation projection for an English-German and English-Czech translation pair, respectively. We use the projected data as training data in a supervised learning framework to train a SRL (SDP) system in the target language. In cases of missing alignments, the projected data is partial.
Chapter 3: Word Sense Divergence

1 Introduction

One of the main sources of performance degradation in many supervised systems is word sense divergence, also known as false friends and faux amis in the literature. Words with sense divergence, which we call them sense out-of-domain words (sense OOD) throughout this chapter, are words in two or more language variants that are orthographically and/or phonetically similar but do not convey the same meaning (Brown and Allan, 2010). These words are particularly found when the underlying distribution of the test set is different from that of the train data. Statistical machine translation (SMT) systems are one of the supervised models whose performance is substantially affected by sense OOD words. Sense of a word with sense divergence in the input sentence varies from all observed senses of that word in the train data. Thus, the SMT model may choose the target language translation which is considered inappropriate based on the context.

Standard form of a language has different informal spoken varieties which are known as dialects. For instance, the standard form of Arabic has different dialects (Habash, 2010). These dialects typically share a set of cognates that could bear the same meaning in both varieties or only be shared homographs but serve as sense OOD words. For instance the word “\(yEny\)^1” in Modern Standard Arabic (MSA) means “to mean”, but in Egyptian (EGY), it is a pragmatic marker meaning “to some extent”. The usage of dialects in textual social media and communication channels is rapidly increasing but there is not usually enough dialectal parallel data to train the translation model and build stand-alone dialectal SMT systems. This is while the standard official forms of language usually have a wealth of resources and tools that can be adapted to dialects of that language.

---

^1We use the Buckwalter Transliteration as represented in www.qamus.org for Romanized Arabic representation throughout this chapter.
The main goal of this chapter is to address the word sense divergence issue using word-level semantic information transferred through parallel data from another language and verify its applicability to enhance SMT performance for low-resource dialects without any in-domain training data. We move towards this goal by performing a pre-processing phase consisting: (a) identifying sense OOD words in the input sentence and, (b) disambiguating the identified sense OOD words to find an appropriate equivalent from standard variant which bears the same meaning. We investigate two types of models for this purpose:

- **Transfer-based approach** This model uses clues provided by the available parallel data to identify sense OOD words and choose the most appropriate target translation for that word. Our transfer-based model aims to identify sense OOD words without any labeled training data. We then try to choose equivalent from the standard language for the identified sense OOD word. We exploit a classifier for identifying these words and designing a word sense disambiguator for finding the best equivalent from the standard language. We employ unsupervised word alignment from parallel text and a taxonomy-based semantic similarity measure (Wu and Palmer, 1994) to automatically acquire training data for the sense OOD identifier. Our word sense disambiguator benefits from unsupervised word clusters to model the context. We obtain word clusters from a large monolingual text in the standard language. Training the model only involves counting the cooccurrences of each word with word clusters for different context definitions. During decoding (disambiguation), for a word in a sentence, we estimate the likelihood of different candidates given word clusters in its surrounding context.

- **Knowledge-oriented Approach** This model performs sense OOD identification and replacement using two available supervised tools: first, a state-of-the-art morphological analyzer and disambiguation that generates full fledged morphological analysis for both MSA and EGY, and second a dialect identification tool which is able to identify and classify dialectal words on the token and sentence levels as well as providing both
the MSA equivalent lemma(s) and corresponding English gloss(es) for the identified dialectal word.

We evaluate our method on Egyptian to English (EN) SMT system using a translation model trained on Modern Standard Arabic. We show that our transfer-based approach improves EGY-to-EN SMT lexical choice and yields 0.6% and 0.1% reduction in word error rate (WER) and position-independent error (PER) (Tillmann et al., 1997) over the baseline respectively. Furthermore, we compare our transfer-based model with the knowledge-oriented approach. We show that the transfer-based model obtains noticeably close results to the knowledge-oriented approach, which demonstrates the capabilities of the transfer-based model for further improvements. In summary, the main contributions of our transfer-based model are: 1) designing a sense OOD identifier with a supervised classifier trained on automatically acquired labeled data, 2) designing a disambiguator for replacing sense OOD words with their equivalent standard form, and 3) improving the SMT lexical choice on dialectal data without using any in-domain parallel data to train SMT model.

2 Transfer-based Approach

Our transfer-based model consists of two main modules: 1) a sense OOD identifier (sometimes referred as identifier for brevity) and, 2) a classifier to disambiguate the sense of identified OOD word (sometimes referred as disambiguator). The identifier is based on a supervised classifier. The training data for the identifier is automatically obtained from parallel data. The disambiguator is based on the likelihood of each standard equivalent given the contextual information. In all of our definitions, we use $DA$ and $ST$ to refer to a dialectal and standard language, respectively.

2.1 Sense OOD Identifier

We first give some basic definitions about the setup. Our parallel data $\mathcal{P}$ is a set of aligned sentences $[(s^{(1)}, t^{(1)}), \cdots, (s^{(n)}, t^{(n)})]$ from the source and target languages. We use
English as the source language throughout this chapter. We denote $S$ to be $[s^{(1)}, \ldots, s^{(n)}]$ and assume that it consists of both ST and DA sentences. Each training instance for our identifier is shown with a tuple $(k, i, y)$ in which $1 \leq k \leq n$, $1 \leq i \leq |s^{(k)}|$ and $y \in \mathcal{Y} = \{\text{SO, SI}\}$ representing sense OOD and sense in-domain respectively. We represent the $i$th word in the $k$th sentence as $s_i^{(k)}$ and its features as $\phi(k, i) \in \mathbb{R}^d$ where $d$ is the size of feature vector. Given the set of training tuples $(k, i, y)$, a classification algorithm is used to train the model. We use Averaged Perceptron (Freund and Schapire, 1999a) algorithm as our classifier with the following features: word form for the current word and part of speech tag for the previous, current and next words.

### 2.1.1 Generating Synthetic Train Data for Sense OOD Identifier

In this section we describe the steps undertaken to generate train data for the sense OOD identification module. Assuming we have access to a dialect identification tool, we define a function $L(k, i)$ that identifies the dialect for $s_i^{(k)}$ out of two possibilities: DA and ST. We perform automatic word alignment on $P$ using Giza++ to obtain $a_{k,i}^T$ (c.f. §5 for details). We define $a_{k,i}^T$ to be the English word aligned to $s_i^{(k)}$. We define $E_{w}^{ST}$ as the set of all English words aligned to the source word $w$ for the cases where our language identification tool has identified $w$ as ST. $E_{w}^{ST}$ can be written as:

$$E_{w}^{ST} = \{\forall e \in En| \exists k, i A_{k,i}^T = e, w_{k,i} = w, L(k, i) = ST\}$$

To confront the noise in the automatically acquired word alignments, we limit training instances to the aligned word pairs with frequency more than 5. For each $s_i^{(k)}$ where $L(k, i)$ is equal to DA, we have to decide whether the word is sense OOD or not. We define a function $F(s_i^{(k)}, a_{k,i}^T)$ that returns true if we decide to predict $s_i^{(k)}$ as sense OOD and false otherwise.
\[ F(w_{ki}, A^T_{k,i}) = \begin{cases} \text{SOD} & \text{Sim}(A^T_{ki}, E^ST_{w_{ki}}) < \delta \\ \text{SID} & \text{otherwise} \end{cases} \]

where \( \delta \) is a manually defined threshold and \( \text{Sim}(\cdot) \) is defined as follows:

\[ \text{Sim}(e, E) = \frac{1}{|C_E|} \sum_{c \in C_E} \frac{\sum_{e' \in c} \text{dist}(e, e')}{|c|} \]

where \( C_E \) partitions \( E \) into non-overlapping clusters. Each \( c \in C_E \) contains a cluster of words in \( E \) with similar meaning. The clusters are obtained using the Wu-Palmer distance measure (Wu and Palmer, 1994) which is calculated as:

\[ \text{dist}(e, e') = \frac{2 \cdot d(s_{e,e'})}{d(e) + d(e')} \]

where \( s_{e,e'} \) is a maximally specific superclass of \( e \) and \( e' \) in WordNet (Miller, 1995) and \( d \) is the depth of the node in the WordNet taxonomy. This way we compute a weighted average similarity between various ST senses of the target word and its DA sense in the sentence \( k \). The intuition behind this setting is as follows: for a particular word that is identified as DA in a sentence, we measure similarity of its aligned English word to the set of all English words aligned to ST occurrences of the same word \( (E^ST_{w_{ki}}) \). If this similarity is less than a threshold \( \delta \), we label that word as sense OOD. We set \( \delta \) to 0.5 in our experiments.

### 2.2 Sense OOD Disambiguation

We now describe our disambiguation model. Here, we aim to replace a word that is identified as sense ood by one of its ST equivalents based on the context. We use a large amount of monolingual data \( \mathcal{D} \) in the ST form and perform unsupervised word clustering on \( \mathcal{D} \) to obtain word cluster assignments for each word. We then use word clusters to build our disambiguation model. Our model comprises five parameters:

- **Word context parameters**: \( P_{\tau}(c|w) \) for \( \tau \in \{-2, -1, +1, +2\} \) and \( c \in \{1, 2, ..., C\} \),
where $C$ refers to the number of clusters. $P_{\tau}(c|w)$ shows the probability that a word from cluster $c$ appears in the window within the offset $\tau$ from word $w$. We estimated this probability using maximum likelihood estimation with additive smoothing. The smoothing parameter is set to 0.1 in our experiments. To avoid sparsity, we assume that all previous contexts are the same and analogously all next contexts are also the same. In other words, we tie $P_{-2}(c|w)$ and $P_{-1}(c|w)$ into one parameter and $P_{2}(c|w)$ and $P_{1}(c|w)$ into another distinct parameter.

- Word probability distribution $P(w)$.

Let $\Omega(w)$ be the list of ST equivalents for the DA word $w$. We choose the most probable candidate $\omega^*$ using the following equation:

$$w^* = \arg\max_{w \in \Omega(w)} \log P(w) + \sum_{\tau \in \{-2,-1,1,2\}} \log P_{\tau}(c|w)$$

The intuition here is: if a particular DA word in a sentence is identified as FF, we want to replace it by one of its ST equivalents. If an alternative word is more likely to appear in that context compared to other possible equivalents, we expect our model to select that as the replacement. Since we train word clusters on ST data, the model tends to assign more weight on words that fit better to ST contexts.

3 Knowledge-oriented Approach

We further compare the performance of our unsupervised approach for sense OOD identification and disambiguation (described in §2.1 and §2.2) to a knowledge-oriented approach that exploits two publicly available Arabic dialect identification tools to identify dialectal words and replace them by the appropriate MSA equivalent. In the context of SMT for DA-to-EN, we encounter a significant sense OOV rate between test and training data since the size of the training data is relatively small. On the other hand, we have significant amounts of MSA-to-EN parallel data to construct rich phrase tables. MSA and DA, though
divergent, they share many phenomena that can be leveraged for the purposes of SMT. Hence, if we combine training data from MSA with that from DA, and then at the decode time normalize sense OOV words into their equivalent MSA counterparts we should be able to overcome the resource challenges in the DA-to-EN SMT context, yielding better overall translation performance. Figure 3.1 shows the block diagram of the proposed system. To achieve this goal, we use the following tools and resources:

- **Madamira**: A system for morphological analysis and disambiguation for both MSA and EGY (Pasha et al., 2014b). Madamira indicates whether a word is EGY or MSA based on its underlying lexicon and generates an equivalent EN gloss for the identified word. If the word is EGY, then EN gloss from Madamira is used to find the most probable equivalent MSA lemma(s) from Tharwa (Diab et al., 2014), a three-way lexicon between EGY, MSA, and EN.

- **AIDA**: A full-fledged dialect identification tool which is able to identify and classify EGY words on the token and sentence level (Elfardy and Diab, 2013). AIDA exploits Madamira internally in addition to more information from context to identify EGY words. AIDA provides both the MSA equivalent lemma and corresponding EN gloss for the identified EGY word.
• **Tharwa:** A three-way lexicon between EGY, MSA and EN Diab et al. (2014).

To evaluate the effectiveness of using each of these resources in identification and disambiguation of sense OOD, we have exploited the following replacement schemes:

• AIDA identifies DA words in the context and replaces them with the most probable equivalent MSA lemma;

• MADAMIRA determines whether a word is DA or not. If the word is DA, then EN gloss(es) from MADAMIRA are used to find the most probable equivalent MSA lemma(s) from Tharwa.

Since the DA identification resources devised here return MSA equivalents in the lemma form, we adopt a factored translation model to introduce the extra information in the form of lemma factors, thereby DA replacement affects only the lemma factor in the factored input. We consider the following setups to properly translate replaced MSA lemma to the corresponding inflected form (lexeme):\(^2\) lexeme-to-lexeme and lemma+POS-to-lexeme translation. The first path translates directly from a source lexeme to the target lexeme. So it provides appropriate lexeme translations for the words (MSA or DA) which have been observed in the trained model. The second path is similar to the lem+POS-to-lex path and is used to translate DA words that do not exist in the trained model.

4 SMT Experiments

In this section, we give details about the dataset and tools used in our experiments.

**Data Set and Tools**  To train the sense OOD identifier, we use parallel data \(P_{ME}\) which is a collection of MSA and EGY texts created from multiple LDC catalogs\(^3\). The data comprises 29 million MSA and 5 million DA tokenized words from multiple genres including newswire,

\(^2\)We use the term lexeme to indicate an inflected tokenized uncliticized form of the lemma. A lemma in principle is a lexeme but it is also a citation form in a dictionary.

\(^3\)41 LDC catalogs including data prepared for GALE and BOLT projects.
broadcast news, broadcast conversations, and weblogs. To train the disambiguator, we use the Arabic Gigaword 4 (Graff and Cieri, 2003) containing 848 million tokenized MSA words. To train the model described in §2.2, we exclude punctuation as well as clitics from the target word local context. These words usually do not provide much information about the target word and will increase model sparsity. All data sets used in our experiments have undergone the following preprocessing steps: all Arabic data is Alef/Ya normalized and tokenized using MADAMIRA v1 using the Arabic Treebank (ATB) tokenization scheme (Maamouri et al., 2004). We used Tree Tagger (Schmid, 1994) to tokenize English data. We evaluate our method on a highly DA test set selected from LDC2012E30 data set (BOLT-arz-test).

We use GIZA++ (Och and Ney, 2003) for word alignment. We obtain word clusters from word2vec (Mikolov et al., 2013a) K-means word clustering tool. We use the continuous bag of word model to build word vectors of size 200 using a word window of size 8 for both left and right. The number of negative samples for logistic regression is set to 25 and threshold used for sub-sampling of frequent words is set to $10^{-5}$ in the model with 15 iterations. We also use full softmax to obtain the probability distribution. We use AIDA as the dialect identification tool. AIDA also provides a list of MSA equivalents for identified DA words in context.

**SMT System Configurations** We use Moses decoder (Koehn et al., 2007) to build a standard phrase-based SMT system. Feature weights are tuned to maximize BLEU score on the tuning set using Minimum Error Rate Training (MERT) (Och, 2003) algorithm. Final results are reported by averaging over three tuning sessions with random initialization. Significance test is also performed to make sure that gains in the results are statistically significant. We use the implementation of (Clark et al., 2011) to compute the p-value via approximate randomization algorithms. Since AIDA generates MSA equivalents in the lemma form, we use a factored translation model with lemma and POS factors. We use GIZA++ (Och and Ney, 2003) to word align the parallel corpus. We use SRILM (Stolcke
<table>
<thead>
<tr>
<th></th>
<th>BLEU</th>
<th>METEOR</th>
<th>TER</th>
<th>WER</th>
<th>PER</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASELINE1</td>
<td>20.6</td>
<td>27.5</td>
<td>65.9</td>
<td>69.2</td>
<td>45.3</td>
</tr>
<tr>
<td>BASELINE2</td>
<td>20.1</td>
<td>27.2</td>
<td>68.3</td>
<td>71.6</td>
<td>46.6</td>
</tr>
<tr>
<td>BASELINE3</td>
<td>21.3</td>
<td>28.0</td>
<td>65.2</td>
<td>68.6</td>
<td>44.6</td>
</tr>
<tr>
<td>PARL</td>
<td>20.7</td>
<td>27.1</td>
<td>67.5</td>
<td>69.6</td>
<td>45.5</td>
</tr>
<tr>
<td>WC&lt;sub&gt;cor&lt;/sub&gt;</td>
<td>20.9</td>
<td>27.7</td>
<td>65.4</td>
<td>68.7</td>
<td>44.8</td>
</tr>
<tr>
<td>PARL+WC</td>
<td>21.0</td>
<td>27.7</td>
<td>66.2</td>
<td>68.5</td>
<td>45.3</td>
</tr>
<tr>
<td>PARL+WC&lt;sub&gt;cor&lt;/sub&gt;</td>
<td>21.3</td>
<td>27.9</td>
<td>65.5</td>
<td>68.0</td>
<td>44.5</td>
</tr>
</tbody>
</table>

Table 3.1: BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), TER (Snover et al., 2006), WER, PER results on the Bolt-arz test set compared to the baselines.

et al., 2002) to build 5-gram language models with modified Kneser-Ney smoothing (Kneser and Ney, 1995). Our language modeling data consists of three data sets: a) The English Gigaword; b) the English side of the BOLT Phase 1 parallel data; and, c) different LDC English corpora collected from discussion forums<sup>4</sup>. The translation model is trained using the MSA part of $\mathcal{P}_{ME}$ with 29M words. Therefore, any improvement in translating DA words on the test set is gained by our sense OOD identification and disambiguation approach. Our test set comprises 16K tokenized EGY words and is acquired by selecting 1065 sentences from BOLT-arz-test<sup>5</sup>. The tuning set contains 1547 sentences obtained from multiple LDC catalogs<sup>6</sup> and comprises 20k tokens.

5 Evaluation Results

The main goal of this work is to improve the translation chosen by SMT for a sense OOD word based on its surrounding context. In this section, we first demonstrate performance of our transfer-based sense OOD identifier and disambiguator (§2.1 and §2.2) and then compare it with the results obtained from knowledge-based approach described in §3.
5.1 Transfer-based Approach

The final SMT performance is affected by two factors: first, the accuracy of sense OOD identifier and disambiguator. Second, the quality of predefined candidates generated by AIDA which are later used in the disambiguation module. In order to accurately evaluate the quality of our identification and disambiguation process, we design three different baselines: as the first baseline, we randomly tag EGY words which have been observed as MSA in the train data as sense OOD. These words are then replaced by a randomly selected sense from the respective candidates list (BASELINE1). As the second baseline, we follow the setup introduced in (Aminian et al., 2014). In this baseline, all EGY words that fulfill the mentioned criteria, are replaced with one randomly selected sense from the list of candidates (BASELINE2). As the third baseline, we use the results of the raw baseline without any replacement (BASELINE3). The first two baselines can be used to evaluate the accuracy of sense OOD identifier and disambiguator modules. The last baseline evaluates the overall effectiveness of our approach to enhance EGY-to-EN SMT which depends on both factors mentioned before.

The first three rows of Table 3.1 show BASELINE1, BASELINE2 and BASELINE3 results on our test set. PARL in the fourth row demonstrates the setup that only parallel data is exploited to identify sense OOD words. The identified DA word is then replaced by a randomly selected MSA sense from the candidate list. Similarly, WC_{cor} shows the setup where WC is directly used to identify and replace sense OOD words. In this setup, the original EGY word is manually added to the list of MSA candidates generated by AIDA. Thus, the disambiguation module has access to the list that contains both MSA equivalents and original EGY word and can choose the most adequate candidate from this list. In other words, the disambiguation module simultaneously performs identification and sense disambiguation. PARL+WC refers to the system that uses PARL to identify sense OOD and
then WC to disambiguate them. It is to be emphasized that in this setup, WC chooses the most appropriate MSA equivalent of each sense OOD only from the list of candidates generated by AIDA. We also define PARL+WC$_{cor}$ in which WC$_{cor}$ is used as an OOD identifier as well as disambiguator (similar to the second setup above). In fact, we prevent mistakes from PARL by using WC as an identifier as well. This setup replaces a word by its MSA equivalent only if both PARL and WC identify it as FF.

As shown in Table 3.1, all replacement experiments outperform BASELINE1 and BASELINE2 in terms of BLEU score. PARL improves BASELINE1 and BASELINE2 BLEU score by 0.1% absolute (0.5% relative) and 0.6% absolute (3% relative) receptively. This implies that our sense OOD identifier achieves more accurate predictions compared to random and blind predictions. Using WC$_{cor}$ for sense OOD identification and disambiguation shows a noticeable improvement over the case that we just use PARL for identification (in terms of BLEU, WER and PER). This shows that contextual similarity plays an important role compared to the information extracted from parallel data to train a sense OOD identification model. PARL shows high sensitivity to errors in the word alignment because the noise existing in the alignment leads to incorrect prediction and thereby, inadequate replacement.

As expected, combining PARL and WC for sense OOD identification and replacement (PARL+WC) outperforms the individual decisions made by each module solely. This setup benefits from the evidences provided by both modules for identification and sense disambiguation. Eventually, the last setup PARL+WC$_{cor}$ yields in 0.3% absolute (1.4% relative) BLEU improvement over PARL+WC. It also outperforms PARL+WC in terms of other SMT evaluation metrics such as METEOR, TER, WER and PER: it achieves 0.7%, 0.5% and 0.8% reduction in TER, WER and PER respectively compared to PARL+WC. In the last setup, we just replace words which both PARL and WC commonly identify them as sense OOD. In other words, WC refines some of the PARL mistakes and avoids it from replacing words which are mistakenly identified as sense OOD by PARL. It is worth noting that our significance tests show all gains in the BLEU, METEOR, and TER over BASELINE2 and
Ref. | i will tell you a story, and you judge whose fault it is.
---|---
Baseline | Tb AnA H+ AHky l+ HDrp +k mwqf w+ tqwly myn Ally glLTAn
Replacement | tmAm AnA H+ AHky l+ HDrp +k mwqf w+ tqwly myn Ally glLTAn
Baseline Trans. | ok, i am going to talk to you and say who was wrong.
Replacement Trans. | i will talk to you stand and say who was wrong.

Table 3.2: Example of correct sense OOD identification and replacement with non-improving BLEU score.

BASELINE3 are statistically significant at the 95% level.

Our best performing setup, PARL+WC<sub>cor</sub>, reduces BASELINE3 WER and PER by the noticeable amount of 0.6% and 0.1% respectively. This demonstrates that our approach has the power to enhance SMT lexical choice and select more accurate target translations for the sense OOD words. However, our method does not outperform BASELINE3 BLEU score. Our analysis shows that the main reason for this phenomenon is that the SMT translation table does not contain adequate bilingual phrase pairs for some of the replaced MSA equivalents (suggested by AIDA). Thus, the decoder can not generate coherent phrases while translating these words. As an example, consider the sentence shown in Table 3.2. Word “Tb” in the baseline sentence means “all right”, “very well” or “ok” in EGY while it means “medicine” when used as MSA. Our sense OOD identifier has correctly identified this word. The disambiguator module also has adequately replaced word “Tb” with the MSA word “tmAm” which means “ok”. However, this replacement did not yield to a better translation for this word. This happens because word “tmAm” has not been observed as an interjection in our SMT phrase table. Thus, SMT decoder is not able to find a good translation for this word.

5.2 Knowledge-oriented Approach

We additionally compare our transfer-based system with the knowledge-based approach described in §3. Experimental results in Table 3.3 shows that transfer-based approach results in reduction of 0.4 and 0.7 in the BLEU score on the same test set compared to the knowledge-based approaches of Aida and Madamira respectively that is considered a modest reduction having the fact that transfer approaches do not have access to any annotated data.

32
and only rely on parallel cues.

### 6 Analysis and Discussion

We conducted another analysis to closely assess the impact of our disambiguator module (wc) in improving target BLEU score. We ran our replacement setups on the proportion of Bolt-arz sentences which contain at least one sense OOD. These words are predicted by the PARL module. We ended up getting a set with 796 sentences. Table 3.4 shows the percentage of BLEU-enhanced and BLEU-degraded sentences in this set for each setup compared to the baselines separately. The setup which exploits \( WC_{cor} \) for sense OOD identification and disambiguation is excluded from this comparison as it does not use PARL for sense OOD identification. As the percentages in Table 3.4 indicate, PARL+WC noticeably increases percentage of BLEU-enhanced sentences compared to the PARL setup with respect to BASELINE1 and BASELINE2. As shown before (Table 3.1), the last setup PARL+WC\(_{cor} \) did not improve BASELINE3 BLEU score. However, results in Table 3.4 show that this setup increases percentage of BLEU-enhanced sentences compared to the PARL+WC and PARL with respect to BASELINE3 significantly. Comparing percentages of BLEU\_degraded sentences for mentioned setups gives the same results.

<table>
<thead>
<tr>
<th>Setup</th>
<th>BASELINE1</th>
<th>BASELINE2</th>
<th>BASELINE3</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARL</td>
<td>37.7/38.5</td>
<td>41.5/40.3</td>
<td>34.7/44.2</td>
</tr>
<tr>
<td>PARL+WC</td>
<td>38.7/32.8</td>
<td>45.5/36.4</td>
<td>34.0/35.2</td>
</tr>
<tr>
<td>PARL+WC(_{cor} )</td>
<td>40.5/32.2</td>
<td>46.4/36.3</td>
<td>35.7/35.1</td>
</tr>
</tbody>
</table>

Table 3.4: Percentage of BLEU enhanced/degraded sentences for different replacement approaches compared to each baseline separately.
<table>
<thead>
<tr>
<th>Ref.</th>
<th>not private , i mean like buses and the metro and trains ... etc .</th>
</tr>
</thead>
</table>
| Baseline | mc mlkyp xASp yEny AqSd  
| Replacement | mc mlkyp xASp yEny AqSd mvl AqTAr . . . Alx |
| Baseline Trans. | privately , i mean , i mean , i do not like the bus and subway train , etc . |
| Replacement Trans. | not privately , i mean , i mean , such as the bus and subway train , etc . |
| Ref. | let us forget about our differences and unite . |
| Baseline | nsyb +nA mn AlAxAf w+ ntwHd |
| Replacement | trk +nA mn AlAxAf w+ ntwHd |
| Baseline Trans. | we disagree and suffering from |
| Replacement Trans. | let us from the difference and unify |
| Ref. | and those who said that the girls ... indeed , i heard very bad words , why ? |
| Baseline | w+ Ally yqwl AlbnAt . . . b+ jd smEt AllfAZ wHcp qwy lyh kdh |
| Replacement | w+ Ally yqwl AlbnAt . . . b+ jd smEt AllfAZ syC qwy lyh kdh |
| Baseline Trans. | and to say ... very very difficult . that is why i heard |
| Replacement Trans. | and to say ... seriously , i heard a strong bad , why ? |
| Ref. | at least three parties ; check them and read about them in detail |
| Baseline | Ely AlAql three AHzAb cfw +hm w+ AqrA +hm b+ Emq |
| Replacement | Ely AlAql three AHzAb rAy +hm w+ AqrA +hm b+ Emq |
| Baseline Trans. | at least three of the depth of them and with them . |
| Replacement Trans. | at least three parties see them and baqir them in depth |
| Ref. | it is waiting for disagreement between the salafis and the liberals , which engages them in a new battle of nonsense speech similar to |
| Baseline | yntZr An yxtlf Alslfywn mE AllybrAlyyn f+ ydxlwA fy Erkp Ely +k jdydp mn qbyl rmy |
| Replacement | yntZr An yxtlf Alslfywn mE AllybrAlyyn f+ ydxlwA fy mErkp Ely +k jdydp mn qbyl rmy |
| Baseline Trans. | it is expected that the salafis disagrees with liberals , in testing on your new prior to throw |
| Replacement Trans. | waiting for the salafis disagrees with liberals , in the battle for your new prior to throw |
| Ref. | also eradication of poverty and need is very important , toqua |
| Baseline | w+ kmAn AlqDAC Ely Alfqr w+ HAjp mhm jdA yA+ tqa |
| Replacement | w+ kmAn AlqDAC Ely Alfqr w+ Amr kbyr jdA yA+ tqa |
| Baseline Trans. | and also the eradication of poverty and need is very important , |
| Replacement Trans. | and also the eradication of poverty and a very large , |

Table 3.5: Translation examples with and without replacement drawn from Bolt-arz test.

Table 3.5 shows some translation examples with and without any replacement. The replacement is done using our best-performing setup PARL+WC,cor on Bolt-arz test set. The first four examples demonstrate cases that sense OOD (shown in bold) are correctly identified and replaced with a proper MSA equivalent. For instance, the word “zy” in the first example means uniform or clothing in MSA and such as or like in EGY. Thus, replacing the word “zy” with MSA word “mvl” which means like yields to better translation and thereby, improves BLEU score. In the second example, word “nsyb” which means forget in this context is replaced with MSA equivalent “trk” that means leave or forget. As a result, the decoder has translated phrase “trk +nA mn AlAxAf” into a longer phrase let us from the difference.
instead of generating an incoherent translation such as baseline.

Word “wHcp” in the third example is not a pure EGY word. However, it conveys a meaning different from its observed senses in the phrase table. Hence, baseline incorrectly translates this word to *difficult* while the replaced setup generate the correct translation *bad* for the replaced MSA equivalent “syC”. Hence, as shown, our approach has improved SMT lexical choice significantly in this example. Word “cwf” in the fourth example is also correctly identified as a sense OOD according to context. This word is used as noun in MSA with meanings *look* and *appearance* while it is used as a command verb (*order someone to look*) in EGY. As we can see, our disambiguator module has adequately replaced this word with the verb “rAy” which means *to look at or to see*. As a result, the decoder has translated this word into the word *see* in the English sentence which leads to higher BLEU score compared to the baseline translation. Word “Erkp” in the fifth example has English equivalent *battle* in EGY and *test* in MSA context. Similar to the previous example, baseline selects the incorrect translation *testing*. While our replacement setup substitutes this word with MSA equivalent “mErkp” which means *battle* and thereby, improves the translation.

Sixth instance in Table 3.5 demonstrates the example where our identifier has incorrectly identified word “HAjp” (*need* in this context) as sense OOD. This word is then replaced by the word “Amr” (*order*) which does not convey the original word meaning according to context. Hence, the decoder is not able to find a proper translation for the replaced word in the context.

7 Related Work

In this section, we first review related studies performed for sense out-of-domain identification and its usage in SMT. Since we extrinsically evaluate our sense out-of-domain identification and disambiguation model in the context of EGY-to-EN SMT, we also highlight our contributions in the area of dialectal Arabic SMT.
Sense Out-of-domain Identification  There have been several studies for identifying sense out-of-domain words which benefit from parallel data to measure the semantic similarity of words (Frunza and Inkpen, 2006; Nakov et al., 2009; Inkpen et al., 2005; Kondrak, 2001; Mitkov et al., 2007). Some other studies such as (Nakov et al., 2007; Schulz et al., 2004; Nakov et al., 2009; Mulloni et al., 2007) exploit distributional semantics to identify sense out-of-domain words. These methods hypothesize that words occurring in similar contexts tend to be semantically similar. Methods leveraging this idea usually use vector space models to show the local context of the target word. Context can be modeled either with a window of a certain size around the target word (Nakov et al., 2009; Schulz et al., 2004) or words in a particular syntactic relationship with the target word (Mulloni et al., 2007). The most comparable work to our sense out-of-domain identification approach presented in (Aminian et al., 2015) is the work done by Mitkov et al. (2007). Mitkov et al. (2007) use both distributional semantic evidence extracted from monolingual data and bilingual hints obtained from comparable corpora. They eventually use this information as features in a classifier and achieve up to 20% and 37% improvement over the baseline precision and recall, respectively. Our sense out-of-domain identification method, however, is different from the mentioned studies in the sense that we generate a supervised classifier from fully unsupervised labeled data. And unlike previous work that solely focus on the identification task, our model leverages both identification and disambiguation.

Word Sense Disambiguation for SMT  From the sense disambiguation perspective, there have been several attempts to integrate word sense disambiguation (WSD) systems into the SMT framework in recent years. The main goal of these studies is to improve the target translation for an ambiguous word in the source sentence. Most studies in this area incorporate supervised WSD systems which exploit labeled training data. Amongst these studies, Carpuat and Wu (2005) integrate a supervised WSD model trained on the Senseval-3 Chinese lexical sample task data into a standard Chinese-English phrase-based SMT model
with two methodologies: first, at the decode time, they limit set of translation candidates for an ambiguous word to the set of translations mapped to the sense predicted by the WSD model. Second, they replace the translations chosen by SMT with the translation predicted by WSD system. Nevertheless, they show none of these methods improves baseline BLEU score. Vickrey et al. (2005) formulate the task of using WSD for SMT as word translation task. They use parallel data to train their WSD model. They showed that they improve accuracy in both word translation and blank-filling tasks. However, they did not incorporate their word translation setup in an end-to-end SMT system. In another attempt, Carpuat and Wu (2007) transformed the problem into a phrase sense disambiguation task by incorporating state-of-the-art WSD features for selecting a target phrase out of all aligned phrases as the possible senses. Chan et al. (2007) also embedded state-of-the-art WSD system into SMT by adding more features into the SMT model. They showed that they improve Baseline BLEU score using their WSD-based model. Yang and Kirchhoff (2012) use an unsupervised WSD to improve SMT final performance. Similar to previous studies, they add the WSD acquired feature to the SMT model. They could improve the BLEU score by 0.3% compared to the baseline.

All the mentioned studies aim to enhance SMT by identifying the appropriate target translation for a source word in a given context. Our SMT approach (Aminian et al., 2015) is different from previous work in two aspects: first, we try to improve SMT lexical choice by identifying false friends and replacing them with the most adequate equivalent from the standard language. Unlike previous work, all these steps are done on a given input sentence and we can see them as a pre-processing phase, thereby, there is no need to change the SMT model. Second, our approach does not assume availability of any in-domain parallel data. Hence, it is not constrained by the domain and can be extended to any other language variants.
Dialectal Arabic SMT   Leveraging Modern Standard Arabic (MSA) resources and tools to enrich Dialects (DA) for NLP purposes has been explored in several studies. Chiang et al. (2006) exploit the relation between Levantine Arabic (LEV) and MSA to build a syntactic parser on transcribed spoken LEV without using any annotated LEV corpora. Since there are no DA-to-MSA parallel corpora, rule-based methods have been predominantly employed for DA-to-MSA translation. For instance, Bakr et al. (2008) introduce a hybrid approach to transfer a sentence from Egyptian (EGY) into a diacritized MSA form. They use a statistical approach for tokenizing and tagging in addition to a rule-based system for constructing diacritized MSA sentences. Moreover, Al-Sabbagh and Girju (2010) introduce an approach for building a DA-to-MSA lexicon through mining the web. In the context of DA SMT, Sawaf (2010) introduced a hybrid SMT system that uses statistical and rule-based approaches for DA-to-EN SMT. In his study, DA words are normalized to the equivalent MSA using a dialectal morphological analyzer. This approach achieves 2% absolute BLEU enhancement for Web texts and about 1% absolute BLEU improvement over the broadcast transmissions. Furthermore, Salloum and Habash (2012) use a DA morphological analyzer (ADAM) and a list of hand-written morphosyntactic transfer rules (from DA to MSA) to improve DA-to-EN SMT. This approach improves BLEU score on a blind test set by 0.56% absolute BLEU (1.5% relative) on the broadcast conversational and broadcast news data. Test sets used in their study contain a mix of Arabic dialects but Levantine Arabic constitutes the majority variety. Zbib et al. (2012) demonstrate an approach to acquire more DA-to-EN data to improve DA SMT performance by enriching translation models with more DA data. They use Amazon Mechanical Turk to create a DA-to-EN parallel corpus. This parallel data is augmented to the available large MSA-to-EN data and is used to train the SMT system. They showed that their trained SMT model on this DA-to-EN data achieves 6.3% and 7% absolute BLEU enhancement over an SMT system trained on MSA-to-EN data when translating EGY and LEV test sets respectively. Habash (2008) demonstrates four techniques for handling OOV words through modifying phrase tables containing MSA entries. He
also introduces a tool which employs these four techniques for online handling of OOV in SMT (Habash, 2009). Habash et al. (2013) introduces MADA-ARZ, a new system for morphological analysis and disambiguation of EGY based on an MSA morphological analyzer MADA (Habash and Rambow, 2005). They evaluate MADA-ARZ extrinsically in the context of DA-to-EN SMT and show that using MADA-ARZ for tokenization leads to 0.8% absolute BLEU improvement over the baseline which is simply tokenized with MADA. In one of our efforts in this vein, we use MADAMIRA (Pasha et al., 2014a), a system for morphological analysis and disambiguation for both MSA and DA (EGY), to identify DA words and replace MSA equivalents (Aminian et al., 2014). Our approach achieves 0.6% absolute BLEU improvement over the scores reported in Habash et al. (2013).

8 Conclusion

We presented a transfer-based approach for improving cross-lingual SMT performance without any in-domain training data by identifying sense OOD words and replacing them with a semantically similar equivalent from the standard language. We show that our approach improves lexical choice in EGY-to-EN SMT system trained only on MSA data. We demonstrate a fully unsupervised approach for sense OOD identification and disambiguation using the evidences extracted from parallel and monolingual data. We showed that our best-performing setup reduces the baseline WER and PER by the noticeable amount of 0.6% and 0.1% respectively. One interesting line to expand this study is exploring an automatic way to generate the list of possible equivalents for sense OOD words instead of using a predefined inventory of senses. One idea is benefiting from continues word vectors and their similarity to extract possible word senses for a particular sense OOD from the available monolingual corpus. Given the importance of contextual information in addressing sense divergence, we expect that incorporating deep contextualized embeddings such as BERT (Devlin et al., 2019) and XLM (Lample and Conneau, 2019), in the form of monolingual or cross-lingual embeddings, leads to significant enhancements in identification and disambiguation of word
sense divergence. Moreover, sense OOD disambiguation in the model presented in this chapter is limited to the set of word senses observed in the train data or provided by the dialect identification tool. In the next chapter, we try to address this limitation by developing a multilingual correspondence learning algorithm that can be used to expand the set of correspondents used for sense disambiguation.
Chapter 4: Multilingual Correspondence Learning

1 Introduction

Sense OOD disambiguation discussed in Chapter 3 is limited by the particular senses of the target word that are either observed in the train data or provided by the dialect identification tool. To expand the set of correspondents, we devise a multilingual correspondence learning framework using evidence leveraging parallel and monolingual corpora. Machine-readable multilingual lexica are typically created by a combination of manual and automatic (semi-automatic) techniques. This illustrates the need for continuous verification of the quality of the lexica during the development process. Researchers have mainly resorted to using manual evaluation to verify coverage, automatically extend and measure the accuracy of different lexical resources such as multilingual lexica and WordNets (Sagot and Fišer, 2011a,b, 2012; Saleh and Habash, 2009). In this chapter, we devise a framework for learning multilingual correspondents using evidence leveraging parallel and monolingual corpora. The proposed method is capable of detecting inconsistencies in the existing lexicon entries and possibly providing/suggesting candidates to replace them. Accordingly, one can exploit this method to automatically augment multilingual lexica with partially or completely new entries. Naturally the method lends itself to also bootstrapping multilingual lexica from scratch, however, this is outside the scope of the present work.

We demonstrate the efficacy of our proposed framework in the context of verifying and augmenting a publicly available lexicon that is manually created Tharwa (Diab et al., 2014). Tharwa is an electronic three-way lexicon comprising Egyptian Dialectal Arabic (EGY), Modern Standard Arabic (MSA) and English correspondents (EN). The entries in Tharwa are in lemma form. We show that our approach obtains F1-score of 71.71% in generating multilingual correspondents which match with a gold Tharwa set. We further evaluate our approach against the Arabic entries in BabelNet (Navigli and Ponzetto, 2012). We show that
our automated approach reaches F1-score of 54.46% in generating correct correspondents for BabelNet Arabic entries.

2 Our Model

We exploit parallel corpora to generate the initial set of multilingual correspondents. This set is further expanded with correspondents extracted from various monolingual and cross-lingual resources:

Leveraging Parallel Resources Here, we assume that we have access to two parallel corpora with common target side denoted as $P_{1 \rightarrow 2}$ and $P_{3 \rightarrow 2}$, where $P_{l \rightarrow l'}$ shows parallel sentences in the source language $l$ and target language $l'$. Moreover, we have word alignment technology to automatically induce word correspondents from parallel corpora. We derive the initial set of multilingual correspondents $T$ from $P_{1 \rightarrow 2}$ and $P_{3 \rightarrow 2}$ by pivoting through the common language $l_2$ using:

$$T = \{(w_{l_1}^{l_2}, w_{l_2}^{l_2}, w_{l_3}^{l_2}) | w_{l_2}^{l_1} \in e(w_{l_2}^{l_2}, 2, 1), w_{l_3}^{l_2} \in e(w_{l_2}^{l_2}, 2, 3)\}$$

where $e(w, i, j)$ returns the list of all $w' \in l_j$ which have been aligned to $w \in l_i$. We refer to the generated multilingual word level correspondents as multilingual tuples or simply tuples hereafter. Nevertheless, there is always some noise in the automatic word alignment process. We prune a large portion of the noise by applying constraints on part-of-speech tags (POS) correspondence, thereby accepting tuples in $T$ with a certain mapping between POS tag categories. We call the pruned set $T'$ and acquire it using:

$$T' = \{(w_{l_1}^{l_1}, w_{l_2}^{l_2}, w_{l_3}^{l_3}) \in T | M(pos(w_{l_1}^{l_1})) = pos(w_{l_3}^{l_3})\}$$

where $M(pos(w_{l_1}^{l_1}))$ is the POS mapping function. Here $pos(w_{l_1})$ refers to the POS tag of $w_{l_1}$ of either source languages ($l_1$, $l_3$). This mapping function lets us account for some
language-dependent functional divergences that happen when translating a word with a
certain POS tag from source to the target language. For instance, word *jmylp*¹ as an adjective
in EGY could end up being aligned through pivoting on English to the same word in MSA
but functioning in context as a noun. Additionally, to mitigate several limitations posed
by the size and coverage of the parallel corpora, we leverage several monolingual and
cross-lingual resources to expand multilingual correspondents:

Leveraging Monolingual Resources Parallel corpora pose several limitations in size
and coverage for the extracted multilingual correspondents due to the domain and genres
variation of naturally available data. Accordingly, to mitigate these limitations we propose
expanding a target word with all its synonyms. We use the following methods that leverage
different monolingual resources to expand $T'$:

- **WordNet:** One can use synonyms that WordNet generates to expand a word. Before
  expanding monolingual correspondents in $T'$, we perform word sense disambiguation
  using the word sense disambiguation module of Pedersen et al. (2005). If a word
  belongs to more than one WordNet synset, word sense is used to disambiguate the
  correct synset to expand. We additionally use POS tags to filter returned synonyms.

- **Word Clusters:** Not all languages have an extensively developed WordNet. There-
  fore, we leverage monolingual corpora to expand words to their semantically similar
  correspondents. Thereby, having large monolingual corpora in any of the languages
  present in our lexicon, we can generate high quality word clusters. Accordingly, we
  exploit existing methods to obtain vector-space word embeddings. Word vectors
  are then clustered using a hard clustering technique such as K-means. Namely, we
  expand each correspondent in $T'$ with all the words from the same cluster that the
  correspondent belongs to. We also use POS tags to skip irrelevant words. This can be

¹Arabic characters are shown using Buckwalter transliteration scheme throughout this paper. Transliteration
table can be found in [http://www.qamus.org/transliteration.htm](http://www.qamus.org/transliteration.htm)
done for any language in our lexicon conditioned on the fact that the language has enough monolingual data to induce word clusters. We acknowledge, however, that induced clusters do not necessarily contain exclusively semantically similar synonym words. There might be related and irrelevant words altogether.

**Leveraging Cross-lingual embedding** We further incorporate multilingual evidence into monolingual vectors-space word embeddings. Cross-lingual CCA model proposed by Faruqui and Dyer (2014) projects vectors of two different languages into a shared space where they are maximally correlated. Correlation is inferred from an existing bilingual dictionary for the languages. Having projected vectors of a particular language, we expect the synonyms of a word to be found amongst the most similar words in the projected space. Each word is then expanded with the $k$ most similar words acquired from the projected vector-space model.

### 3 Experiments

**Dataset and Tools** We use Bolt-ARZ v3+v4 for EGY-EN parallel data. This data comprises 3.5 million EGY words. For MSA-EN parallel data, we use GALE phase4 data which contains approx. 60 million MSA words. Additionally, we use multiple monolingual EGY corpora collected from Bolt and ATB data sets with approx. 260 million words (EGY$_{mono}$) to generate monolingual word clusters. We furthermore acquire a collection of several MSA LDC data sets from several years with 833 million words (MSA$_{mono}$) to induce monolingual MSA word clusters. We use EGY$_{mono}$ and English Gigaword 5th Edition (Graff and Cieri, 2003) to train the cross-lingual CCA embedding model. We perform some preprocessing steps to clean, lemmatize and diacritize the Arabic side of both parallel data sets and render the resources compatible. For the sake of consistency, the lemmatization step

---

2 MSA and EGY parallel data are collected from 41 LDC catalogs including data prepared for DARPA GALE and BOLT projects.

3 This data is collected from 70 LDC catalogs including Gale, ATB and Arabic Gigawords4 projects.
is replicated on the English data. The tool we use for processing Arabic is MADAMIRA v1.0 (Pasha et al., 2014b), and for English, we use TreeTagger (Schmid, 1994). Hence, all the entries in our resources are rendered in lemma form, with the Arabic components being additionally fully diacritized. The lemmatized-diacritized corpora with the corresponding EN translations are word aligned using GIZA++ (Och and Ney, 2003) producing pairwise EGY-EN and MSA-EN lemma word type alignment files, respectively. We intersected the word alignments on the token level to the type level resulting in a cleaner list of lemma word type alignments per parallel corpus. All correspondents in the form of EGY-EN-MSA are extracted from both alignment files by pivoting on the EN correspondents. We may use TransDict to denote the set of extracted multilingual correspondents throughout this chapter.

We obtain monolingual embeddings using word2vec (Mikolov et al., 2013a). We use the Skip-gram model to build word vectors of size 300 from EGY\textit{mono} and MSA\textit{mono} corpora using a word window of size 8 for both left and right. The number of negative samples for logistic regression is set to 25 and the threshold used for sub-sampling of frequent words is set to $10^5$ in the model with 15 iterations. We also use full softmax to obtain the probability distribution. Word clusters are obtained from word2vec K-means word clustering tool with k=500. We additionally induce clusters with k=9228 corresponding to the number of synsets in the Arabic WordNet (Black et al., 2006). Word2vec is also used to generate vectors of size 300 using a continuous bag of word model from English Gigaword. The generated vectors of a) EGY\textit{mono}-English Gigaword, and b) MSA\textit{mono}-English Gigaword are then used to train the Cross-lingual CCA model. Projected EGY and MSA vector space models are used to get a list of synonyms for the EGY and MSA words in TransDic. For EN expansion, we initially expand all the EN correspondents in the extracted correspondents using synonyms extracted from WordNet3. We further expand the acquired correspondents with EGY and MSA correspondents using either word clusters or cross-lingual synonyms obtained from cross-lingual CCA model.
Experimental Conditions  To assess the effectiveness of leveraging each of the mentioned
resources in the quality of extracted multilingual correspondents, we design the following
experimental setups:

- **PARL**: In this setup, only parallel data is used to generate correspondents. We
  consider this setup to be our baseline.

- **WC**: In this setup, we expand the lemmas in a source language (MSA or EGY) using
  lemma clusters induced over word2vec vectors in addition to PARL.

- **SYN**: In this setup, we expand the lemmas in a source language (MSA or EGY) using
  cross-lingual synonyms by leveraging cross-lingual CCA (SYN) together with PARL.

- **EN-WSD**: In this setup, we expand English lemmas using word sense disambiguation
  to generate WordNet synsets for the pivot language EN.

Having the above setups, we present results for the following experimental conditions
corresponding to various extraction methods: (a) baseline PARL; (b) PARL+EGY-WC
where we expand the EGY lemmas using WC clusters; (c) PARL+EGY-SYN where we
expand EGY lemmas using the SYN expansion method; (d) PARL+MSA-WC where we
expand the MSA lemmas using WC clusters; (e) PARL+EGY-SYN where we expand
MSA lemmas using the SYN expansion method; (f) PARL+EN-WSD where we are only
expanding the English lemmas using WSD; (g) PARL+EN-WSD+EGY-WC+MSA-WC
where all three languages are expanded: EN using WSD, EGY and MSA are expanded using
WC; and, (i) PARL+EN-WSD+EGY-SYN+MSA-SYN, similar to condition (g) but EGY
and MSA are expanded using SYN.

Evaluation Resources:  We measure the quality of the correspondents generated by our
approach using two multilingual resources. The first resource is BabelNet (Navigli and
Ponzetto, 2012), a multilingual semantic network comprising concepts and named entities
lexicalized in different languages including MSA, EGY and En. The second resource is Tharwa, a three-way lexicon containing MSA, EGY and En correspondents. All entries in both resources are in lemma form and marked with a POS tag. BabelNet is comprised of multilingual synsets. Each synset consists of multilingual senses including MSA, EGY and En. First, we iterate over all synsets of type CONCEPT\textsuperscript{4} and extract tuples in the form MSA-En-EGY from each synset that fulfill the following conditions: a) none of MSA, En and EGY words are out of vocabulary with respect to our corpora and, b) MSA, En and EGY, each, are not composed of more than a single word. We acquired 8381 BabelNet tuples applying the above constraints. It is worth emphasizing that this evaluation is limited to measuring quality of the generated multilingual correspondents. The first constraint ensures that no mismatch happens due to domain divergence. Also since generated correspondents contain only single-word elements, we limit the set of extracted BabelNet tuples to the singletons.

We additionally define a particular subset of the Tharwa lexicon as the gold standard to measure the performance of generated correspondents. Similar to BabelNet, gold Tharwa contains MSA-En-EGY tuples from original Tharwa where none of their correspondent words is out of vocabulary with respect to the MSA, En and MSA corpora. Gold Tharwa obtained according to the above conditions contains 19459 rows. We focus on the three major fields in Tharwa, namely: EGY lemma, MSA lemma, and En lemma equivalents and their corresponding POS tags. This condition ensures that none of the mismatches is caused by domain divergence between Tharwa and generated correspondents.

4 Results

Table 4.1 shows precision, recall and F1–score of different correspondent extraction setups against BabelNet and Tharwa. The results reflect full exact match, where extracted correspondents fully matched BabelNet/Tharwa entries including POS tag match. This is

\textsuperscript{4}Named entities are excluded from the comparison.
Table 4.1: Precision, recall and F-score of different correspondence learning methods against BabelNet and Tharwa respectively.

<table>
<thead>
<tr>
<th>Extraction Method</th>
<th>BabelNet</th>
<th>Tharwa</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>PARL</td>
<td>84.9%</td>
<td>21.26%</td>
</tr>
<tr>
<td>PARL+EGY-WC</td>
<td>90.00%</td>
<td>22.54%</td>
</tr>
<tr>
<td>PARL+EGY-SYN</td>
<td>86.61%</td>
<td>21.69%</td>
</tr>
<tr>
<td>PARL+MSA-WC</td>
<td>77.68%</td>
<td>23.79%</td>
</tr>
<tr>
<td>PARL+MSA-SYN</td>
<td>81.08%</td>
<td>22.65%</td>
</tr>
<tr>
<td>PARL+EN-WSD</td>
<td>87.16%</td>
<td>34.34%</td>
</tr>
<tr>
<td>PARL+EN-WSD+EGY-WC+MSA-WC</td>
<td>87.82%</td>
<td>39.47%</td>
</tr>
<tr>
<td>PARL+EN-WSD+EGY-SYN+MSA-SYN</td>
<td>86.26%</td>
<td>36.05%</td>
</tr>
</tbody>
</table>

the harshest metric to evaluate against. We note similar trends across the two evaluation data sets. In general, recall is quite low for BabelNet compared to Tharwa which might be relegated to some domain divergence between our corpora and BabelNet resources where a word might not be out of vocabulary but its sense of a word is, thereby it is not found in the set of extracted correspondents. It should be noted that we only constrained the entries in the gold by being in vocabulary for our corpora without checking if the senses were in vocabulary. This effect is not observed in Tharwa as much due to the relative sizes of BabelNet (almost 9K entries) and Tharwa (almost 20K entries). Expanding EN with WSD significantly improves the results (PARL F1-score is 34.01% vs. 49.27% for PARL+EN-WSD for BabelNet, and 60.63% for PARL vs. 65.3% for PARL+EN-WSD for Tharwa). This is result of the significant increase in recall with little impact on precision. Expansion for MSA and EGY in general yield better results over the baseline in terms of overall F1-score. However, expanding MSA negatively affects precision compared to recall. In general, WC expansion yields better results than SYN for EGY across both evaluation data sets. However, we note that for MSA expansion, for Tharwa, SYN outperforms WC, contrasting with WC outperforming WC for MSA against BabelNet data. For both BabelNet and Tharwa evaluation sets, we note that the same condition PARL+EN-WSD+EGY-WC+MSA-WC yields the highest results of (54.46% and 71.71% F1-score, respectively).
<table>
<thead>
<tr>
<th>Extraction Method</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARL</td>
<td>79.15%</td>
<td>65.14%</td>
<td>71.46%</td>
</tr>
<tr>
<td>PARL+EGY-WC</td>
<td>84.51%</td>
<td>69.55%</td>
<td>76.3%</td>
</tr>
<tr>
<td>PARL+EGY-SYN</td>
<td>80.65%</td>
<td>66.37%</td>
<td>72.79%</td>
</tr>
<tr>
<td>PARL+MSA-WC</td>
<td>76.00%</td>
<td>67.9%</td>
<td>71.72%</td>
</tr>
<tr>
<td>PARL+MSA-SYN</td>
<td>78.31%</td>
<td>66.9%</td>
<td>72.19%</td>
</tr>
<tr>
<td>PARL+EN-WSD</td>
<td>79.30%</td>
<td>73.97%</td>
<td>76.54%</td>
</tr>
<tr>
<td>PARL+EN-WSD+EGY-WC+MSA-WC</td>
<td>82.99%</td>
<td>82.99%</td>
<td>82.99%</td>
</tr>
<tr>
<td>PARL+EN-WSD+EGY-SYN+MSA-SYN</td>
<td>79.95%</td>
<td>76.09%</td>
<td>77.97%</td>
</tr>
</tbody>
</table>

Table 4.2: Precision, Recall and F1-score of TransDict dialectal component EGY against Tharwa

5 Discussion

Evaluating Dialectal Extraction Component Most multilingual lexica are bilingual lexica, but in the current research atmosphere, many researchers would like to have true multilingual resources that go beyond a pair of languages at a time. Hence we evaluate the quality of adding a third language to an already existing bilingual resource. The method can be extended beyond 3 languages, but for sake of exposition, we focus on adding a third language in the scope of this paper. Accordingly, we specifically measure the quality of the extracted EGY correspondents compared to a subset of the Tharwa lexicon. This reference subset must contain EGY-EN-MSA correspondents from our gold Tharwa that satisfy these constraints: 1) EGY correspondent is found in the EGY monolingual corpora, 2) MSA-EN correspondents match with at least one row in the extracted correspondents and 3) POS tag of the Tharwa row matches POS tag of the extracted correspondents. Here, the first constraint avoids domain divergence between Tharwa and extracted correspondents. The second constraint is applied because we focus on measuring the quality of the EGY extraction component, thus fixing MSA-EN. Additionally, the POS constraint is meant to strengthen the match.

Table 4.2 demonstrates the results of comparing the extracted correspondents dialectal entries with Tharwa. Results are assuring that performance of dialectal extraction component is persistently higher than the quality of entire extracted correspondents yielding highest
Table 4.3: Mapping between MSA POS tags and EGY POS tags

<table>
<thead>
<tr>
<th>MSA POS</th>
<th>EGY POS</th>
<th>MSA POS</th>
<th>EGY POS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABBREV</td>
<td>→ABBREV, NOUN</td>
<td>INTERROG-PRON →INTERROG-PRON, PRON</td>
<td></td>
</tr>
<tr>
<td>ADJ</td>
<td>→ADJ, NOUN, NOUN-PROP</td>
<td>NEG-PART →NEG-PART, PART</td>
<td></td>
</tr>
<tr>
<td>ADJ-COMP</td>
<td>→ADJ-COMP, ADJ, NOUN</td>
<td>NOUN →NOUN, ADJ, NOUN-PROP</td>
<td></td>
</tr>
<tr>
<td>ADJ-NUM</td>
<td>→ADJ-NUM, ADJ, NOUN</td>
<td>NOUN-NUM →NOUN-NUM, NOUN</td>
<td></td>
</tr>
<tr>
<td>ADV</td>
<td>→ADV, NOUN</td>
<td>NOUN-PROP →NOUN-PROP, NOUN</td>
<td></td>
</tr>
<tr>
<td>CONJ</td>
<td>→CONJ, PART</td>
<td>NOUN-QUANT →NOUN-QUANT, NOUN</td>
<td></td>
</tr>
<tr>
<td>DEM-PRON</td>
<td>→DEM-PRON, PRON, NOUN</td>
<td>PESUDO-VERB →PESUDO-VERB, PART</td>
<td></td>
</tr>
<tr>
<td>EXCLAM-PRON</td>
<td>→EXCLAM-PRON, PRON</td>
<td>REL-ADV →REL-ADV, ADV, NOUN</td>
<td></td>
</tr>
<tr>
<td>FOCUS-PART</td>
<td>→FOCUS-PART, PART</td>
<td>REL-PRON →REL-PRON, NOUN, PRON</td>
<td></td>
</tr>
<tr>
<td>FUT-PART</td>
<td>→FUT-PART, PART</td>
<td>RESTRIC-PART →RESTRIC-PART, PART</td>
<td></td>
</tr>
<tr>
<td>INTERROG-ADV</td>
<td>→INTERROG-ADV, PART</td>
<td>SUB-CONJ →SUB-CONJ, CONJ, PART</td>
<td></td>
</tr>
<tr>
<td>INTERROG-PART</td>
<td>→INTERROG-PART, PART</td>
<td>VERB →VERB, IV, PV</td>
<td></td>
</tr>
<tr>
<td>VOC-PART</td>
<td>→VOC-PART, PART</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

F1-score of 82.99%. Similar to the trends observed in the overall evaluation, PARL+EN-WSD+EGY-WC+MSA-WC yields the highest performance.

POS Mapping Constraints and Number of Word Clusters We can prune noisy correspondents by applying POS constraints in the process of extracting multilingual correspondents. Table 4.3 shows the mapping used between MSA and EGY POS tags. Results demonstrated in Table 4.1 are obtained when exact POS match constraint is used, meaning only MSA-EN-EGY correspondents are included in the extracted correspondents that their MSA and EGY have the exact same POS tags.

In this section, we pick the best-performing setup from Table 4.1 (PARL+EN-WSD+EGY-WC+MSA-WC) and study effects of different POS matching constraints and also the number of word clusters on the results. The first row of Table 4.4 shows Precision, Recall and F1-score of evaluating PARL+EN-WSD+EGY-WC+MSA-WC against Tharwa when no constraint is applied on POS tags. The second row shows relaxed POS match results when we accept certain POS divergence patterns between MSA and EGY as a valid POS match. Finally, the last row shows the match results for the case where the same POS tags on the EGY and MSA is included in the extracted correspondents.

In addition to different POS constraints, Table 4.4 shows results when different cluster
sizes are exploited for monolingual expansion. The reason we choose k=9228 in addition to k=500 (which has been frequently used for clustering in the literature) is that it encodes the total number of synsets in Arabic WordNet. As shown in Table 4.4, F1-score generally decreases when more POS constraints are used. This mainly happens because system recall gradually drops when stricter POS constraints are applied. Therefore, we might dismiss some of the correct correspondents but we expect correspondents with higher purity in this case. Nonetheless, we notice precision increases in the relaxed mode as we are allowing for more divergence accommodation. On the other hand, we observe that the F1-score drops when the number of clusters increases from 500 to 9228 (regardless of the POS constraint used). This suggests that despite getting purer clusters in the case of the 9228 setting, we are losing significant numbers of synonyms by over-segmenting the semantic space.

In order to measure the quality of the generated EGY candidates and also assess the feasibility of using this component to augment Tharwa with other dialects, we perform two manual assessments of the generated correspondents, assuming a partial match; first, we compile a random sample of size 1000 from the extracted correspondents that match with the gold Tharwa rows, i.e. whose MSA-EN-EGY are found in Tharwa. We also have the corresponding list of other potential EGY candidates generated by our correspondents for each row of this sample as augmented candidates. We obtain this augmented candidate list from two different setups: a) PARL+EN-WSD+EGY-WC+MSA+WC with 500 clusters, and b) PARL+EN-WSD+EGY-WC+MSA-WC with 9228 clusters.

An expert annotator is asked to manually assess the list of augmented EGY candidates
and decide how many candidates in the list are actual synonyms of the gold EGY word. Manual annotation shows that on average 6.6% of EGY candidates provided by TransDict in each row are actual synonyms of the gold EGY word in the 500 cluster setup (a). The match percentage increases to 21.6% for the second setup, the 9228 clusters case (b). This shows that increasing the number of clusters leads to purer clusters in the matching process. The remaining irrelevant (non-synonym) candidates are caused by either erroneous word alignments or lack of efficient pruning criteria in the correspondence learning algorithm.

Second, we carry out an analysis to assess the potential for augmenting Tharwa with generated EGY correspondents. We create a random sample of size 1000 from Tharwa rows where their MSA-EN is found in TransDict (EN expansion setup) but none of TransDict EGY candidates matches with Tharwa gold EGY (non-matched rows, i.e. our errors). Here, the annotator was asked to mark EGY candidates (generated by TransDict) that are synonyms of the TransDict generated EGY word. According to our manual assessment by an expert, 78.1% of the rows in the given sample contained at least one synonym of the gold EGY word. Hence, we expect that the actual matching accuracy over the entire gold Tharwa is 93.8%.

Table 4.5 shows the list of EGY candidates generated by TransDict for different EN senses of two MSA-EGY tuples in Tharwa.\(^5\) For the first tuple, where we found a match with Tharwa, wAd (EGY)-walad (MSA), we show the list of words that were found in TransDict. We note that we for both the EN corresponding senses boy and child, the EGY word wAd is listed and highlighted in boldface. We also note the correspondents yielded in TransDict rendered in red in the Table to indicate that they are different senses that are not correct for the triple. As an example, the word janotalap is slang for polite which is could be pragmatically related to boy as in not a polite way to call on a man for example. The highlighted words in the Table show incorrect sense correspondences given the entire tuple. These could have resulted from sense variations in the pivot EN word such as correspondents

\(^5\)Arabic examples in Table 4.5 are shown according to safe Buckwalter scheme to avoid some of the special characters in the original Buckwalter encoding.
of *child* in the case of *binot*, meaning *girl/child/daughter* and that given our techniques would naturally cluster with *wAd* as in the female of *boy/child/son*. We also see related words such as *daloE* meaning *pampering*. For example, *wAdiy* is a synonym of *wAd* meaning *valley* however, not *child*. Accordingly, errors observed are a result of various sources of noise: misalignments, sense divergences for any of the three languages, differences in vowelization between the EGY resources. The second tuple in Table 4.5 shows cases where no matches are found with Tharwa in *TransDict*, yet the resulting *TransDict* entries comprise correct correspondents but they are not covered in Tharwa hence they are viable candidates for augmentation. The third tuple in the Table shows cases where the entry in Tharwa is incorrect and would need to be corrected. For example, the English word should have been *workshop* not *operator*. Thereby highlighting these partial matches allows for a faster turn around in fixing the underlying lexicon Tharwa.

We finally attempt to assess the amount of possible augmentation of whole entries to Tharwa for completely unseen triplets and verify their validity. We compile a list of a 1000 triplets generated in *TransDict* where none of the word types (EN, EGY, MSA) is seen in any entry in Tharwa. 85% of these entries are considered correct by the expert lexicographer.
6 Related Work

Machine-readable multilingual lexica are typically created by a combination of manual and automatic (semi-automatic) techniques. This illustrates the need for continuous verification of the quality of the lexica during the development process. Approaches exploited for lexicon evaluation and verification mainly comprise manual assessment and human verification. This process is expensive and poses several limitations in terms of domain coverage as well as the amount of data that can be manually evaluated. Hence, it is often desired to automate the evaluation process and reduce manual annotation expenses.

Researchers have mainly resorted to using manual evaluation to verify coverage, automatically extend and measure the accuracy of different lexical resources such as multilingual lexica and WordNets (Sagot and Fišer, 2011a,b, 2012; Saleh, 2009). For instance, Saleh (2009) proposes an approach for extracting an Arabic-English dictionary while exploiting different human-annotated samples to measure the accuracy of the extracted dictionary. De Melo and Weikum (2009) use human-annotated samples to measure the accuracy of the multilingual dictionary they extract. More recently, Navigli and Ponzetto (2012) benefit from manual evaluation by expert annotators to assess the coverage of additional lexicalizations provided by their resource and not covered in existing lexical knowledge bases.

In our work (Aminian et al., 2016), we devise a framework for automatic augmentation of multilingual correspondents using evidence leveraging parallel and monolingual corpora. The proposed method is capable of detecting inconsistencies in the lexicon entries and possibly providing/suggesting candidates to replace them. Accordingly, one can exploit this method to automatically augment multilingual lexica with partially or completely new entries. Naturally, the method lends itself to also bootstrapping multilingual lexica from scratch, however, this is outside the scope of our work. We demonstrate the efficacy of our proposed framework in the context of augmenting a publicly available lexicon that is manually created, Tharwa (Diab et al., 2014). Tharwa is an electronic three-way lexicon
comprising Egyptian Dialectal Arabic, Modern Standard Arabic and English correspondents. The entries in Tharwa are in lemma form.

7 Conclusion

In this chapter, we presented a new approach for automatic extraction of multilingual correspondents leveraging evidence extracted from parallel and monolingual corpora. Extracted multilingual correspondents can be used to verify lexicon converge and detect errors. We showed that our approach reaches F1-score of 71.71% in generating correct correspondents for a gold subset of a three-way lexicon (Tharwa) without any human intervention in the cycle. We also demonstrated that our approach reaches F1-score of 54.46% in generating correct correspondents for Arabic entries in BabelNet.
Chapter 5: Cross-Lingual SRL with Limited Resources

1 Introduction

Semantic role labeling (SRL) is the task of automatically labeling predicates and arguments of a sentence with shallow semantic labels characterizing “Who did What to Whom, How, When and Where?” (Palmer et al., 2010). These rich semantic representations are useful in many applications such as question answering (Shen and Lapata, 2007b) and information extraction (Christensen et al., 2011), hence gaining a lot of attention in recent years (Zhou and Xu, 2015a; Täckström et al., 2015; Roth and Lapata, 2016; Marcheggiani et al., 2017). Since the process of creating annotated resources needs significant manual effort, SRL resources are available for a relatively small number of languages such as English (Palmer et al., 2005), German (Erk et al., 2003), Arabic (Zaghouani et al., 2010) and Hindi (Vaidya et al., 2011) and majority of languages still lack SRL systems. There have been some efforts to use information from a resource-rich language to develop SRL systems for resource-poor languages. Transfer methods address this problem by transferring information from a resource-rich language (e.g. English) to a resource-poor language. Annotation projection is a popular transfer method that transfers supervised annotations from a source language to a target language through parallel data. However, one can encounter various issues while exploiting annotation projection for building the target systems: translation shifts are one class of issues that can lead to erroneous projections, thereby affecting the performance of the SRL system trained on the projections. Translation shifts are usually the result of differences in word order and semantic divergences that usually exist between the source and target language. Errors in the source annotations and automatic word alignments are other sources of the noise in projected annotations, which can lead to a cascade of error in the target side. Moreover, the annotations in different datasets are heterogeneous, making the evaluation more challenging.
In this chapter, we introduce a new approach for a dependency-based SRL system based on annotation projection without any semantically annotated data for a target language. We primarily focus on improving the quality of annotation projection by using translation cues automatically discovered from word alignments. We show that exclusively relying on partially projected data does not yield good performance. We improve over the baseline by filtering irrelevant projections, iterative bootstrapping with relabeling, and weighting each projection instance differently with data-dependent cost-sensitive training.

In short, contributions of this chapter can be summarized as follows: we introduce a weighting algorithm to improve annotation projection based on cues obtained from syntactic and translation information. In other words, instead of utilizing manually-defined rules to filter projections, we define and use a customized cost function to train over noisy projected instances. This newly defined cost function helps the system weight some projections over other instances. We then utilize this algorithm in a bootstrapping framework. Unlike previous bootstrapping methods (Akbik et al., 2015), ours relabels every training instance (including labeled data) in every self-training round. Our final model on transferring from English to German yields a 3.5 absolute improvement labeled F-score over a standard annotation projection method.

2 Baseline Model

We aim to develop a dependency-based SRL system which makes use of training instances projected from a source language onto a target language through parallel data (cf. §6 for more details about annotation projection and dependency-based SRL). Our SRL system is a pipeline of classifiers consisting of: (a) predicate identification and disambiguation module, (b) argument identification module, and (c) argument classification module. Our basic SRL module is a reimplementation of the greedy (local) model of (Björkelund et al., 2009) with a slight modification in the main classifier used; we substitute the logistic regression model used in the original model with an averaged perceptron algorithm (Freund
and Schapire, 1999b) as the learning algorithm. Our preliminary experiments showed that averaged perception leads to equivalent results and much faster training process.

During projection, we apply automatic word alignment on parallel data and preserve the intersected alignments from the source-to-target and target-to-source directions. To rule out projected sentences with sparse alignments, we define a projection density criterion: given a target sentence with \( n \) words, in which \( f \) words have alignments \((f \leq n)\) and a source sentence with \( p \) predicates, in which \( p' \) have alignments \((p' \leq p)\), we define projection density as \(\frac{p' \times f}{p \times n}\) and prune out sentences with a density value less than the threshold \(\delta\). We determine \(\delta\) during tuning experiments performed on the development data. In this criterion, the denominator shows the maximum number of training instances that could be obtained by projection and the nominator shows the actual number of relevant instances that are used in our model. In addition to speeding up the training process, our observations show that filtering sparse alignments helps remove sentence pairs with significant translation shifts. We train our supervised model on the resulting projections.

3 Model Improvements

As already mentioned, the quality of projected roles is highly dependent on different factors including translation shifts, errors in automatic word alignments and the source supervised SRL system. To address these issues, we explore using the following methodologies to improve learning from partial and noisy projections, thereby enhancing the performance of our model:

- **Bootstrapping**: we make use of unlabeled data to enhance our cross-lingual model during an iterative bootstrapping algorithm;

- **Data-dependent cost function**: we define a customized cost function to determine quality of a particular projected semantic dependency during training. We resort to two factors in order to define this cost function: 1) source-target syntactic correspondence;
and, 2) projection completeness. By utilizing these constraints as the training objective, the classifier is enabled to actively distinguish the translation shifts and erroneous instances during training, thereby enhancing the overall performance of the system.

### 3.1 Bootstrapping

Bootstrapping (or self-training) is a semi-supervised technique that gradually uses unlabeled data during an iterative process to improve the training process. A traditional self-training method (McClosky et al., 2006) labels unlabeled data (in our case, fill in missing SRL decisions) and adds that data to the labeled data for further training. We refer to this variation of self-training as *fill–in* in this chapter.

In the field of low-resource SRL, Akbik et al. (2015) showed the effectiveness of the fill–in approach for building a cross-lingual SRL model. As an alternative to the fill–in approach, one can relabel all the training instances (including the already labeled instances) during each iteration of self-training. In our experiments, we tried both strategies and interestingly found it more efficient to relabel all training instances instead of labeling unlabeled instances. This way, our classifier is empowered to discover outliers (resulted from erroneous projections) and change their labels during the training process. Figure 5.1 shows our training algorithm. It starts with training on the labeled instances and uses the trained model to label the unlabeled data instances as well as relabeling the already labeled data. The algorithm iterates over this process for a certain number of epochs until the model converges and the performance is no longer improved.

### 3.2 Data-dependent Cost Function

In our baseline approach, we utilize the standard perceptron training: at each iteration $t$, the weight vector $\theta^t$ is updated after observing a training instance $(x_i, y_i)$ using:

$$\theta^t = \theta^{t-1} + \phi(x_i, y_i) - \phi(x_i, y_i^*)$$
**Inputs:** 1) Projected data \( D = D^L \cup D^U \) where \( D^L \) and \( D^U \) indicate labeled and unlabeled instances in the projected data; 2) Number of self-training iterations \( m \).

**Algorithm:**

1. Train model \( \theta^0 \) on \( D^L \)
2. for \( i = 1 \) to \( m \) do
   1. Label data \( D^U \) with model \( \theta^{i-1} \).
   2. Relabel data \( D^L \) with model \( \theta^{i-1} \).
   3. Train model \( \theta^i \) on \( D^L \cup D^U \)
3. Output: The model parameters \( \theta^m \).

Figure 5.1: The iterative bootstrapping algorithm for training SRL on partially projected data

where \( \phi(x, y) \) denotes the function used to return the feature vector and \( y^* \) denotes the gold label.

It was previously assumed that every data point \( x_i \) in training data has the same importance and cost of wrongly predicting the best label for each training instance is uniformly distributed. This is while different projected instances have different qualities and this is not a realistic assumption while training on projections comprising considerable noise. To confront this issue, we defined and used a customized cost function \( \lambda_i \) for each training instance \( x_i \). Equation 3.2 shows the new update rule of the perceptron with the modified cost function:

\[
\theta^i = \theta^{i-1} + \lambda_i (\phi(x_i, y_i) - \phi(x_i, y_i^*))
\]

Thus the penalty of making a mistake by the classifier for each training instance depends on the importance of that instance defined by a certain cost. The main challenge, however, is how to define an effective cost function, especially in our framework where we don’t have supervision. We used the following criteria to define \( \lambda_i \):

- **Projection completeness**: intuited by our observations, projection density or sparsity is a very important indicator of projection quality and can be seen as a rough indicator of translation shifts: the more alignments from source to target, the less we have a chance of having translation shifts. As an example, consider the sentence pair
extracted from English–German Europarl corpus: “I sit here between a rock and a hard place” and its German translation “Zwei Herzen wohnen ach in meiner Brust” which literally reads as “Two hearts dwell in my chest”. The only words that are aligned (based on the output of Giza++) are the English word “between” and the German word “in”. The German sentence is in fact an idiomatic translation of the English sentence. Consequently predicate–argument structure of these sentences vary tremendously; The word “sit” is the predicate of the English sentence while “wohnen (dwell)” is the predicate of the German sentence. In our work, we devise the definition of completeness already used by Akbik et al. (2015) to define the sparsity cost ($\lambda^\text{comp}$): this definition uses the proportion of verbs and direct dependents of verbs in a sentence that are labeled as an indicator of projection completeness.

- **Source-target syntactic dependency match**: Based on our observations, when the dependency label of a target word is different from its aligned source word, there is a higher chance of a projection mistake. However, given the high frequency of source-target dependency mismatches, it is harmful to blindly prune all projections with a syntactic mismatch; instead, we consider a penalty for observing a training instance with syntactic mismatch. For an argument $x_i$ that is projected from source argument $s_{x_i}$, we define the cost $\lambda^\text{dep}_i$ based on the source-target dependency labels $\text{dep}(x_i)$ and $\text{dep}(s_{x_i})$:

$$
\lambda^\text{dep}_i = \begin{cases} 
1 & \text{if } \text{dep}(x_i) = \text{dep}(s_{x_i}) \\
0.5 & \text{otherwise}
\end{cases}
$$

Figure 5.2a shows an English-German parallel sentence from the Eurparl (Koehn, 2005) and their dependency parse trees generated using a state-of-the-art dependency parser (Rasooli and Tetreault, 2015). Similar dependencies between the source and target sentence are denoted with similar colors. Supervised semantic role structure
of the source sentence (top), word alignment links (dashed lines in the middle) and
projected semantic roles (bottom) are demonstrated in Figure 5.2a. As shown, words
with similar syntactic dependencies have received similar semantic dependencies in
almost all cases. The German sentence shown in this figure is a literal translation of
the English sentence, i.e. we observe no translation shift.

As another example, consider Fig. 5.3 that demonstrates another English-German
sentence pair. The German translation literally reads as “I ask for your approval”. As
we can see, there is a shift in the translation of English clausal complement “to endorse
this” into German equivalent “um Zustimmung (your approval)” which accordingly
has led to different syntactic structures. Therefore, neither the predicate label of the
English verb “endorse” nor its argument “A2” must be projected to the German noun
“Zustimmung”. Here we expect that the customized penalty term used to modify the
training objective prevents the model from learning from an incorrect projection by
down-weighting this instance.

- Completeness + syntactic match: we additinoally employ the average of $\lambda^{dep}$ and
$\lambda^{comp}$ values as defined above. This way, we simultaneously model both the complete-
ness and syntactic similarity information.

4 Experiments

In the experiments presented in this section, we consider English as the source language
and demonstrate the performance of our annotation projection model for building an SRL
system for German as the target language. We consider the low-resource scenario in which
no training data with SRL annotation is available for the target language, thereby we acquire
it by projecting annotations from English. However, we have access to other supervised
models such as part-of-speech (POS) tagging and dependency parsing models for both
the source and target language. Our ideal setting is to transfer to more languages but
We are proposing a system which considerably improves legal certainty for SEMs.

Wir schlagen ein System vor, das die Rechtssicherheit für KMU beträchtlich erhöht.

(a)

We are proposing a system which considerably improves legal certainty for SEMs.

Wir schlagen ein System vor, das die Rechtssicherheit für KMU beträchtlich erhöht.

(b)

Figure 5.2: (a) English-German parallel sentences from Europarl data (Koehn, 2005) with their dependency parse trees. The dependency tree is generated using the Yara Parser (Rasooli and Tetreault, 2015). Similar colors denote groups of similar dependencies in the English and the German sentences. (b) Transferring semantic roles from English to German. The English sentence with semantic roles is shown on top: the semantic roles are generated using our supervised system. The German sentence with its projected semantic roles is shown with dashed lines at the bottom. Dashed lines in the middle show intersected word alignments generated using GIZA++.

Figure 5.3: Example of a parallel English-German sentence with their respective dependency structures. Different dependencies are shown with dashed arcs. The predicate–argument structure of the English sentence is shown below each word in blue.
inconsistencies existing across labeling schemes used to annotate different languages in the CoNLL dataset prevents us from evaluating our model on more languages.

**Dataset and Tools** In our experiments, we use the Universal POS tagset (Petrov et al., 2011) and the Google Universal dependencies (McDonald et al., 2013) for POS tagging and dependency parsing. We ignore the projection of adjunct roles (roles starting with AM prefix) since this particular set of roles does not appear in the German annotations. We use the standard data splits used in the CoNLL2009 shared task on SRL (Hajič et al., 2009) for training and evaluation. We replace the original POS and dependencies provided by the shared task with automatic POS and dependencies produced by the Yara parser (Rasooli and Tetreault, 2015). We trained the parser on the Google Universal Treebank. We use the Europarl corpus (Koehn, 2005) as our parallel data and Giza++ (Och and Ney, 2003) for extracting automatic word alignments. Since predicate senses are projected from English to German, comparing projected senses with the gold German predicate sense is impossible. To address this, all evaluations are conducted using the Gold predicate sense. After filtering projections with density criteria described in §2, almost 29K of the sentences are preserved. The number of preserved sentences after filtering sparse alignments is roughly one percent of the original parallel data (29K sentences out of 2.2M sentences). Density threshold is set to 0.4 and determined based on our tuning experiments on the German development data.

5 Results

Table 5.1 shows the results of different models on the German evaluation data compared to the baseline: the model trained on projected German annotations without incorporating either iterative bootstrapping or cost-sensitive training. As shown, bootstrapping outperforms the baseline. Interestingly, relabelling (Bootstrap–relabel) yields 0.8 absolute improvement in the labeled F1-score compared to the model that supplies predictions to fill in empty slots.

---

1We tried other automatic alignment tools such as fast align as well but did not observe any gain in the overall performance of our model.
<table>
<thead>
<tr>
<th>Model</th>
<th>Cost</th>
<th>Lab. F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>×</td>
<td>60.3</td>
</tr>
<tr>
<td>Bootstrap–fill-in</td>
<td>×</td>
<td>61.6</td>
</tr>
<tr>
<td>Bootstrap–relabel</td>
<td>×</td>
<td>62.4</td>
</tr>
<tr>
<td>Bootstrap–relabel comp.</td>
<td>comp.</td>
<td>63.0(+1.0)</td>
</tr>
<tr>
<td>Bootstrap–relabel dep.</td>
<td>dep.</td>
<td>63.4(+1.8)</td>
</tr>
<tr>
<td>Bootstrap–relabel comp.+dep.</td>
<td>comp.+dep.</td>
<td>63.8(+1.3)</td>
</tr>
<tr>
<td>Supervised</td>
<td>–</td>
<td>79.5</td>
</tr>
</tbody>
</table>

Table 5.1: Labeled F1-score of our model evaluated on the CoNLL2009 German test set. Cost column shows the type and usage of different cost-sensitive training objectives: projection completeness (comp.), source-target dependency match (dep.) and their sum (comp.+dep.). The numbers in parenthesis show the absolute improvement over the Bootstrap-fill-in method. All evaluations are performed using the gold sense of predicates.

...i.e. instances without a projected label (Bootstrap–fill-in). To further clarify the relabeling process, consider the example shown in Figure 5.3, the fill-in approach would label only the German word "um" that does not have any projected label from the English side, while relabeling will overwrite all projected labels with less noisy predictions. We additionally observe that the combination of the two cost functions yields in further improvements and reaches the best results. Overall, the best model yields 3.5 absolute improvement in the labeled F1-score over the baseline. As expected, none of the approaches improves over supervised performance.

6 Analysis

In this section, we provide a detailed analysis of the results. We first look into the effect of applying more constraints (through POS tags) for projecting annotations from English to German. We then investigate how relabeling affects the identification and classification of different non–root semantic dependencies.

6.1 Effect of POS Filter

Rasooli and Collins (2015) demonstrated the usefulness of applying a set of manually defined rules to filter projections for transferring syntactic dependencies. Motivated by
their work, we also explore the effectiveness of POS filters to prune noisy projections and improve the quality of obtained data to train target SRL model. Table 5.2 shows the set of POS equivalence rules defined in (Rasooli and Collins, 2015). The way we perform the process of filtering is as follows: a projected semantic dependency is preserved if the POS tag of the projected predicate and its argument in the source and target language are equivalent i.e. they either match or belongs to the same equivalence class based on Table 5.2. Table 5.3 demonstrates labeled F1–score of different setups on German evaluation data after applying POS filter for projection. As shown in the table, using POS filter leads to a slight drop in the labeled F1–score compared to the results of Table 5.1. Based on our analysis, the significant shrinkage of the size of projections after applying the POS filter is the main reason for this performance degradation.

### 6.2 Effect of Relabeling

We further analyzed the performance of our system on identification and classification of different non–root semantic dependencies (i.e. argument identification and classification) during relabeling iterations. Error analysis is conducted using the best performing setup:

Table 5.2: POS equivalent classes from (Rasooli and Collins, 2015)

<table>
<thead>
<tr>
<th>Model</th>
<th>Cost</th>
<th>Labelled F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bootstrapping–fill-in</td>
<td>×</td>
<td>60.9</td>
</tr>
<tr>
<td>Bootstrapping–relabel</td>
<td>×</td>
<td>62.2</td>
</tr>
<tr>
<td>Bootstrapping–relabel</td>
<td>completeness</td>
<td>61.8 (+0.9)</td>
</tr>
<tr>
<td>Bootstrapping–relabel</td>
<td>dependency match</td>
<td>63.2 (+2.0)</td>
</tr>
<tr>
<td>Bootstrapping–relabel</td>
<td>completeness +dependency match</td>
<td>63.5 (+2.4)</td>
</tr>
</tbody>
</table>

Table 5.3: Labeled F-score for different models in SRL transfer from English to German when POS filter is used for projections. Cost column shows type of cost function used in cost-sensitive training including: projection completeness, source-target dependency match and both. The numbers in parenthesis show the absolute improvement over the fill-in method.
Figure 5.4: Precision, recall and F1-score of VERB+A0, VERB+A1 and ADJ+A1 during relabeling iterations on the German development data. Horizontal axis shows the number of iterations and vertical axis shows the results.

Bootstrap–relabel with cost-sensitive training based on projection completeness and source–target dependency match (last row in Table 5.1) on the German development data.

Figure 5.4 shows precision, recall and F1–score (all labeled) for three semantic dependencies (predicate pos + argument label) throughout relabeling iterations. We assess the effect of relabeling on the performance of our model during different iterations for the two most frequent verbal semantic dependencies (VERB+A0 and VERB+A1) and one semantic dependency with an adjective as the predicate (ADJ+A1).

As demonstrated in the graph, both precision and recall improve by cost-sensitive relabeling for VERB+A0. In fact, cost-sensitive training helps the system refine irrelevant projections at each iteration of relabeling and assign more weight to less noisy projections, hence enhancing precision. Our analysis on VERB+A0 instances shows that source–target dependency match percentage also increases during iterations leading to increase the recall. In other words, weighting projection instances based on dependency match helps the classifier label some of the instances which were ignored during projection, thereby yields improvements in the system recall. While similar improvement in precision is observed for VERB+A1, Figure 5.4 shows that the recall is almost decreasing by relabeling. The detailed analysis revealed that unlike VERB+A0, percentage of source–target dependency match more or less remains steady for VERB+A1. This means that cost-sensitive relabeling for
Figure 5.5: German semantic role labels of the sentence demonstrated in Figure 5.3 after the first iteration of relabeling.

this particular semantic dependency has not been very successful in labeling unlabeled data. The last graph in Figure 5.4 shows changes in precision, recall, and F–score for ADJ+A1. Even though all three measures decrease in the first 6 iterations, we observe they almost grow afterward. This mainly happens due to the small frequency of this particular semantic dependency in the projection data which makes it difficult for relabeling procedure to refine projection instances. Therefore, it needs more training iterations to find noisy instances.

To further illustrate the effect of relabeling in correcting projection mistakes, consider Figure 5.5 which shows the output of the first relabeling iteration over projection done in Figure 5.3. As we can see, applying the source–target dependency match constraint for training helps our system identify the translation shift and filter out the argument “A2”.

7 Related Work

There has been a great deal of interest in using transfer methods for SRL by different techniques such as enhancing the quality of projections (Padó and Lapata, 2005, 2009), joint learning of syntax and semantics (Van der Plas et al., 2011; Kozhevnikov and Titov, 2013b), and iterative bootstrapping to learn a robust model from erroneous projections (Akbik et al., 2015). Padó and Lapata (2005) as one of the earliest studies on annotation projection for SRL using parallel resources, apply different heuristics and techniques to improve the quality of their model by focusing on having a better word and constituent alignments. Using a similar method, Padó and Lapata (2009) find the best alignments between source and target constituents by solving a graph optimization problem. In another study, Van der Plas et al. (2011) improve an annotation projection model by jointly training a transfer
system for parsing and SRL that can discover syntactic and semantic correlations in the
target language, thereby, correcting noisy projections. They solely focus on fully projected
annotations and train only on verbs. In our work (Aminian et al., 2017), we train on all
predicates as well as exploit partial annotation. Kozhevnikov and Titov (2013b) use a shared
feature representation between source and target language, however, they only focus on
verbal predicates and ignore the predicate disambiguation step. The benefit of using shared
representations is complementary to our work, encouraging us to use it in future work.
Kozhevnikov and Titov (2013a) also introduce a cross-lingual model that aims to improve
the SRL system by automatically learning semantic role divergences that occur between the
source and target annotation schemes.

In a more recent study, Akbik et al. (2015) introduce an iterative self-training approach
using different types of linguistic heuristics and alignment filters to improve the quality of
projected roles. These rules make use of syntax as well as bilingual dictionaries to filter
mismatches that happen during annotation projection. Unlike (Akbik et al., 2015) that use
bilingual dictionaries in the process of filtering noisy projections, our work (Aminian et al.,
2017) does not use any external resources. We also leverage self-training with a different
approach: first of all, ours does not apply any heuristics to filter out projections. Second,
it trains and relabels all projected instances, either labeled or unlabeled, at every epoch
and does not gradually introduce new unlabeled data. Instead, we find it more useful to
let the target language SRL system rule out noisy projections via relabeling. We further
utilize a special cost function derived from source-target syntactic correspondences as well
as projection sparsity to enhance target side SRL system (Aminian et al., 2017).

8 Conclusion

We described a method to improve the performance of annotation projection in the
dependency-based SRL task utilizing a data-dependent cost-sensitive training. Unlike
previous studies that use manually-defined rules to filter projections, we benefit from
information obtained from projection sparsity and syntactic similarity to weigh projections. We utilize a bootstrapping algorithm to train an SRL system over projections. We showed that our model yields in particular improvements if we devise entire data relabeling throughout bootstrapping iterations, as opposed to using model predictions for instances that miss a projected label.
Chapter 6: Cross-lingual SRL: From Raw Text to Semantic Roles

1 Introduction

In the previous chapter, we assumed that we have access to some of the enabling supervised tools and annotated resources for the target language and the only missing part is an accurate supervised SRL system, however, it is not usually a realistic assumption for truly low-resource scenarios. In this chapter, we aim to expand our work to a more realistic low-resource setting by building a cross-lingual SRL system that does not rely on any supervised linguistic feature for making predictions in the target language.

Despite considerable efforts on developing semantically annotated resources for semantic role labeling (SRL) (Palmer et al., 2005; Erk et al., 2003; Zaghouani et al., 2010), the majority of languages do not have such annotated resources. The lack of annotated resources for SRL has led to a growing interest in transfer methods for developing semantic role labeling systems. The ultimate goal of transfer methods is to transfer supervised linguistic information from a rich-resource language to a target language of interest. Amongst transfer methods, annotation projection is a method that projects supervised annotation from a rich-resource language to a low-resource language through automatic word alignments in parallel data (Hwa et al., 2002; Padó and Lapata, 2009). Recent work on annotation projection for SRL (Kozhevnikov and Titov, 2013a; van der Plas et al., 2014; Akbik et al., 2015; Aminian et al., 2017) presumes the availability of accurate supervised features such as lemmas, part-of-speech (POS) tags and syntactic parse trees. However, this is not a realistic assumption for truly low-resource languages, for which (accurate) supervised features are hardly available.

In the first part of this chapter, we consider the problem of annotation projection of dependency-based SRL in a scenario for which only parallel data is available for the target language. Recent state-of-the-art SRL systems have shown a significant reliance on the
predicate lemma information while in a low-resource language, a lemmatizer might not be available. We first demonstrate that unsupervised stems can be used as an alternative to supervised lemma features. We further show that we can obtain a robust and simple SRL model for the target language without relying on any explicit linguistic feature (including lemmas), either supervised or unsupervised. We achieve this goal by changing the structure of a state-of-the-art deep SRL system (Marcheggiani et al., 2017) to make it independent of supervised features. Our model solely relies on the word and character level features in the target language. As the second part of this chapter, we challenge the assumption of having access to a large parallel data by using the Bible, a sizable parallel corpus available for the majority of languages. We explore the usage of two direct transfer methods: (a) polyglot training, and (b) aggregating related source embeddings extracted from Wiktionary.

The main contribution of this work is on applying annotation projection without relying on supervised features in the target language of interest. To the best of our knowledge, this is the first study that builds a cross-lingual SRL transfer model in the absence of any explicit linguistic information in the target language. We make use of the recently released Universal Proposition Banks (Akbik et al., 2016)\(^1\), a semi-automatically annotated data that unifies the annotation scheme for all languages. We show the effectiveness of our method on a range of languages, namely German, Spanish, Finnish, French, Italian, Portuguese, and Chinese. We compare our model to a state-of-the-art baseline that uses a rich set of supervised features and show that our model outperforms on six out of seven languages in the Universal Proposition Banks. Furthermore, for Finnish, a morphologically rich language, our model with unsupervised features improves over the model that relies on a supervised lemmatizer.

\(^{1}\)https://github.com/System-T/UniversalPropositions
2 Annotation Projection Model

Our goal is to train an SRL system on the projected predicate-argument structures without having supervised features such as supervised lemmas, dependency parse trees, and part-of-speech tags. Our model has two main components: 1) joint argument identification and classification which we simply refer to as argument classifier, and 2) predicate sense disambiguation. Our argument classifier is independent of sense information, thus we can train the two components independently. Our argument classifier is inspired by the model of (Marcheggiani et al., 2017): we use predicate-specific BiLSTM encoders, and a role+predicate-specific decoder. However, unlike the model of (Marcheggiani et al., 2017), which relies heavily on POS tags and predicate lemmas, we do not use a supervised lemmatizer and POS tagger in any layer. Instead, we benefit from character representations and unsupervised stems to bring in unsupervised features to our model.

2.1 Joint Argument Identification and Classification

Given a sentence $s = [s_i]_{i=1}^{n}$ that contains $n$ tokens with $m$ predicates in the predicate set $\mathbb{P}$, we run $m$ separate predicate-specific deep BiLSTM encoders $[E_j]_{j=1}^{m}$ to extract contextualized representations for each token given a predicate index $p_j$.

**Input Representation** For each encoder $[E_j]_{j=1}^{m}$, we represent each token $s_i$ as the concatenation of its word embedding ($x_i^{re}$ and $x_i^{pe}$), character embedding ($x_i^{char}$) and predicate lemma embedding ($x_i^{lem}$):\footnote{We use [:] notation to show vector concatenation.}

$$x_{i,j} = [x_i^{re}; x_i^{pe}; x_{i}^{char}; x_{i,j}^{le}] \quad \forall i \in [1, \cdots, n]; \quad j \in [1, \cdots, m]$$

where:

- $x_i^{re} \in \mathbb{R}^{d_w}$ is a randomly initialized word embedding vector;
• $x_i^{pe} \in \mathbb{R}^{d_w}$ is an external pre-trained word embedding that is fixed during training;

• $x_i^{\text{char}} \in \mathbb{R}^{d_{ch}}$ is character representation of each token $s_i$. For every token, we obtain $x_i^{\text{char}}$ by running a deep bidirectional LSTM (Hochreiter and Schmidhuber, 1997) on top of each word. We use the concatenation of the final backward representation of the first character, and final forward representation of the last character to represent each token:

$$x_i^{\text{char}} = \text{BiLSTM}(x_i^c[1:|s_i|];|s_i|)$$

where $x_i^c \in \mathbb{R}^{d_c}$ is a randomly initialized character embedding and $|s_i|$ is the number of characters in token $s_i$;

• $x_{i,j}^{\text{le}} \in \mathbb{R}^{d_{le}}$ is a lemma vector for each word $s_i$ with respect to the predicate that is targeted in $E_j$. $x_{i,j}^{\text{le}}$ is active if $s_i$ is the predicate word, otherwise, a zero vector is used to represent the lemma embedding:

$$x_{i,j}^{\text{le}} = \begin{cases} [x_i^{\text{le}};1] & \text{if } i = p_j \\ [\overrightarrow{0};0] & \text{otherwise} \end{cases}$$

In the above equation, the concatenated zero/one value is a flag to indicate if the current token is the targeted lemma. In our model, we use one of the following options to represent predicate lemma:

- Represent each lemma by a deep character BiLSTM. This BiLSTM is different from the character BiLSTM used for acquiring $x_i^{\text{char}}$.

- Use an unsupervised morphological analyzer to give the surface-form stem of each word. This way, we can use a lemma embedding dictionary without requiring a lemmatizer.
**Predicate-Specific Encoder**  A deep BiLSTM is used to get the final representation for each token in a sentence. In the following notation, $h_{i,j}$ is the final hidden state from the deep BiLSTM model for the $i$th token with respect to the $j$th predicate:

$$h_{i,j} = \text{BiLSTM}(s_{1:n,j}; i) \in \mathbb{R}^{d_h}$$

**Role+Predicate-Specific Decoder**  Given the BiLSTM representations, we perform an affine transformation on the concatenation of $h_{p_j,j}$ (predicate representation) and $h_{i,j}$ (argument representation) to find the probability of having the $i$th token as the argument of predicate $p_j$ with role $r$ (including the NULL role):

$$p(r|h_{p_j,j}, h_{i,j}) = \text{softmax}_r(W_{j,r} [h_{p_j,j}; h_{i,j}])$$

where $W_{j,r}$ is a parameter matrix that encodes the information of role $r$ and the $j$th predicate. This matrix is calculated as follows:

$$W_{j,r} = \text{RELU}(U[u^l_j, v_r])$$

where $u^l_j \in \mathbb{R}^{d_l}$ is another predicate lemma embedding parameter which is specifically used for the decoder layer, $v_r \in \mathbb{R}^{d_r}$ is a randomly initialized role embedding, $U$ is a parameter matrix, and $\text{RELU}$ is the rectified linear units activation function (Nair and Hinton, 2010). Similar to the input layer, we represent $u^l_j$ by 1) a different deep character BiLSTM, or 2) a surface-form stem obtained from an unsupervised morphological analyzer.

A graphical depiction of the network in a case for which lemmas are represented by character BiLSTMs is shown in figure 6.1. As shown in the figure, we use two different character BiLSTMs to represent lemmas: one for the input representation and the other for the decoder representation.
Figure 6.1: Graphical depiction of our joint argument identification and classification model without using part-of-speech tags, lemmas, and syntax. In this example, the predicate-specific encoder considers word eats as the sentence predicate and the goal is to score the assignment of argument apple with label A₀. Our model contains three different character BiLSTMs; at the bottom, a character BiLSTM is run to acquire a character-based representation for all the words in the sentence in the absence of POS tags. There are two character BiLSTMs for predicate lemma: one in the encoder level (next to the second word) to model predicate lemma in the input layer and the other in the decoder level (top left). In this example, we just show one layer of BiLSTM but we use a deep BiLSTM in our experiments.
3 Annotation Projection Experiments

**Datasets and Tools** We use English as the source language and project SRL annotations to the following languages: German, Spanish, Finnish, French, Italian, Portuguese, and Chinese. We use the Europarl parallel corpus (Koehn, 2005) for the European languages and a random sample of 2 million sentence pairs from the MultiUN corpus (Eisele and Chen, 2010) for Chinese. We use the Giza++ tool\(^3\) (Och and Ney, 2003) with its default setting for word alignment. We run Giza++ in source-to-target and the reverse direction and get the intersection of alignment links. The intersected alignments are less noisy and guaranteed to be one-to-one. For English, we use the pre-trained embedding vectors generated using the structured skip-gram model of (Ling et al., 2015). For the target languages, we train Word2vec (Mikolov et al., 2013b) on Wikipedia data to generate embedding vectors.

**SRL parameters** We implement our deep network using the Dynet library (Neubig et al., 2017). We use the dimension of 100 for word embeddings, 50 for characters, 512 for LSTM encoders, 128 for role and lemma embeddings in the decoder, and 100 for decoder lemma embedding. We pick random minibatches of size 1000 with a fixed learning rate of 0.001 for learning the parameter values with the Adam optimizer (Kingma and Ba, 2014). The depth of BiLSTM network is set to one for character representation \((x^{char})\) and three for predicate-specific representations \((x^{le}, u^l)\).

**Predicate Disambiguation** Our model is agnostic to predicate senses but since our automatic evaluation relies on automatic predicate senses, we need a disambiguation module. Predicate disambiguation systems typically consist of separate classifiers for each predicate lemma. Since we assume that we do not have a reliable lemmatizer in the target language, we train a single classifier for all predicates. We encode a sentence with a three-layer deep BiLSTM and run a softmax layer on top of each predicate to disambiguate the predicate.

\(^3\)We tried other automatic alignment tools such as fast align as well but did not observe any gain in the overall performance of our model.
sense of each predicate.

**Predicate identification on the source side**  For projection experiments, first of all, we need to identify predicates in the source language. Input to our predicate identifier is the concatenation of word embedding, pre-trained fixed word embedding, POS embedding\(^4\), and character representation (obtained from a character BiLSTM) for every token in the sentence. We use a deep BiLSTM to get the final representation for each token. The ultimate predictions are made by performing an affine transform on the BiLSTM hidden output.

### 3.1 Projection Experiments

Our supervised SRL system is a reimplementation of the model of Marcheggiani et al. (2017). We generate automatic English predicate senses using a system similar to the predicate disambiguation module of Björkelund et al. (2009) except that we replace the logistic regression classifier with the averaged Perceptron algorithm (Collins, 2002). To comply with the Universal Proposition Bank annotation scheme, we convert the argument spans in the English PropBank v3 (Palmer et al., 2005) to dependency-based arguments by labeling the syntactic head of each span.

For annotation projection, we use the density of alignments to find sentences with relatively-dense alignments using:

\[
\text{density}^{(i)} = \frac{\sum_{j=1}^{l'_i} \mathbb{I}(a_j^{(i)} > 0)}{l'_i}
\]

where \(l'_i\) is the length of the \(i\)th target sentence in parallel data, \(a_j^{(i)}\) is the alignment index for the \(j\)th word in the target sentence, and \(\mathbb{I}(a_j^{(i)} > 0)\) is an indicator for a non-NULL alignment.

We prune the target sentence pairs with the density less than 80% for all European languages. We set this threshold to 60% for Chinese to obtain a comparable number of sentences to the European languages. Table 6.1 summarizes the sizes of projected dataset after applying the

\(^4\)Since this is only used for a supervised setting, we can use POS features.
Table 6.1: Size of projected data for languages used in our experiments: German (de), Spanish (es), Finnish (fi), French (fr), Italian (it), Portuguese (pt) and Chinese (zh).

density filter for each language in our experiments. We set the number of training epochs to 2 for all languages based on development results obtained from the English to German projections.

Since the original model of Marcheggiani et al. (2017) heavily relies on the predicate lemma information for making a robust prediction, we further assess the influence of using explicit linguistic features in our model by using a) supervised lemma from the UDPipe pre-trained models (Straka and Straková, 2017), and b) unsupervised stems obtained from the unsupervised morphological analyzer. We use the target side of parallel data to train the unsupervised morphological analyzer of Virpioja et al. (2013). The output from the analyzer does not provide morpheme classes (affixes and stems). To obtain morpheme classes, we train Morfessor FlatCat (Grönroos et al., 2014) on the output of the analyzer. We run the fixed-affix model of Rasooli et al. (2014) to get a unique stem for in-vocabulary words and get unsupervised stems for words outside the vocabulary. The fixed-affix model simplifies the complex output from Morfessor FlatCat to a simple prefix-stem-suffix sequence for which affixes can be empty. In this model, unsupervised stem for a new word is obtained by feeding the word to a tri-state finite state transducer (FST). The FST model segments an input word to a prefix-stem-suffix sequence.
### 3.2 Results

We compare our character-based approach (CModel) with three different models: 1) The cross-lingual model of Aminian et al. (2017) (Bootstrap) that uses a rich set of supervised features including supervised lemmas, POS tags, and dependency parse information, 2) a variant of our model that uses supervised lemmas (SLem) generated by a lemmatizer to represent predicate lemmas in the input and the decode layer, and 3) a model similar to the second model but using unsupervised stems (UStem) generated by an unsupervised morphological analyzer to represent predicate lemmas. Here, we aim to assess the minimal degree of supervision required for the target model by incorporating different levels of explicit linguistic features ranging from fully specified supervised features to unsupervised features in our model. The Bootstrap model uses an iterative bootstrapping approach by utilizing a special cost function and benefiting from a rich set of supervised lexico-syntactic features, thereby, it is considered a hard baseline. Since Bootstrap has a large number of features, the model is not memory-wise scalable to our projection data sizes. Therefore we train the Bootstrap model on a random sample of 20K sentences. This number is similar to the number of sentences used in the original experiments (Aminian et al., 2017).

Table 6.2 shows labeled F1–scores using both gold and automatic predicate senses on the test portion of the Universal PropBank test sets compared to different baselines: the SRL system of Aminian et al. (2017) (Bootstrap), SLem demonstrating the results of the model when supervised lemma is used and UStem that shows the results of our model with unsupervised stem. Numbers in parenthesis denote results with automatic predicate senses.

<table>
<thead>
<tr>
<th>System</th>
<th>de</th>
<th>es</th>
<th>fi</th>
<th>fr</th>
<th>it</th>
<th>pt</th>
<th>zh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bootstrap</td>
<td>59.8 (55.0)</td>
<td>60.6 (52.2)</td>
<td>59.0 (53.1)</td>
<td>71.0 (63.4)</td>
<td>59.2 (52.3)</td>
<td>61.2 (53.9)</td>
<td>50.3 (42.5)</td>
</tr>
<tr>
<td>SLem</td>
<td>61.7 (57.0)</td>
<td>62.4 (55.7)</td>
<td>62.5 (59.2)</td>
<td>65.0 (58.9)</td>
<td>61.8 (56.4)</td>
<td>63.0 (56.8)</td>
<td>52.1 (43.7)</td>
</tr>
<tr>
<td>UStem</td>
<td>62.0 (57.4)</td>
<td>63.0 (56.0)</td>
<td>64.5 (58.8)</td>
<td>65.3 (59.2)</td>
<td>61.3 (55.4)</td>
<td>62.8 (56.8)</td>
<td>52.6 (43.2)</td>
</tr>
<tr>
<td>CModel</td>
<td>61.0 (57.0)</td>
<td>62.5 (56.0)</td>
<td>64.6 (58.9)</td>
<td>65.1 (58.5)</td>
<td>61.0 (55.5)</td>
<td>62.9 (56.5)</td>
<td>52.7 (42.7)</td>
</tr>
<tr>
<td>Supervised</td>
<td>74.5 (72.0)</td>
<td>77.8 (75.2)</td>
<td>74.0 (69.6)</td>
<td>88.9 (87.5)</td>
<td>77.9 (75.9)</td>
<td>66.6 (62.4)</td>
<td>68.8 (68.6)</td>
</tr>
</tbody>
</table>

Table 6.2: Results of projection experiments using our character based model (CModel) on the Universal PropBank test sets compared to different baselines: the SRL system of Aminian et al. (2017) (Bootstrap), SLem demonstrating the results of the model when supervised lemma is used and UStem that shows the results of our model with unsupervised stem. Numbers in parenthesis denote results with automatic predicate senses.
Proposition Banks for each language, thereby can serve as an upper bound for our model. As shown in Table 6.2, our model (CModel) outperforms the Bootstrap model for all languages except French. Additionally, our model performs on par to the supervised lemma and unsupervised stem models. This demonstrates the power of our approach even though our model has access to fewer linguistic features in the target language. Using unsupervised stems outperforms supervised lemma on all languages except Portuguese and Italian. This further highlights the model reliance on the accuracy of lemmatizer. Additionally, results show that except German and Spanish, CModel performs on par to the UStem.

3.3 Analysis

As shown in Table 6.2, using automatic predicate senses for annotation projection experiments leads to almost 4% to 20% reduction in the labeled F1–score depending on the language. Predicate disambiguation performance degradation mainly happens because we have to build a single classifier for all predicate senses, in contrary to most of the previous work that builds separate classifiers for each predicate lemma (and sometimes POS tags). For instance, the predicate disambiguation module of Björkelund et al. (2009) (one of the top-performing systems in the CoNLL 2009 shared task) builds different classifiers for each predicate lemma. They also train different classifiers for nominal and verbal predicates. Since we assume that predicate lemmas and POS tags are not available, we cannot build separate classifiers for each predicate lemma. Consequently, we have a large number of labels for which training instances are scarce, resulting in low precision. For example, the number of unique predicate senses in the German projection data is 7113, from which 60% are observed fewer than 5 times. Moreover, the Universal PropBank dataset contains unified predicate senses for all languages which leads to a lower precision for out-of-vocabulary words. This is while, in previous predicate disambiguation modules, out-of-vocabulary words are simply labeled with their default sense (lemma.01) but we can not use the default predicate sense in a system that uses English sense for all languages. It is also
worth noting that 73% of predicates in English PropBank 3.0 have their default sense, which specifies the lower bound for predicate disambiguation accuracy in English. Similarly, we expect that the lower bound for predicate disambiguation accuracy on projection data without having access to the default predicate senses would be lower than the case that we are allowed to use default senses.

### 3.4 CoNLL Experiments

We additionally conducted experiments on the CoNLL2009 shared task dataset (Hajič et al., 2009) to assess the effectiveness of our character-based model on standard datasets. Table 6.3 shows the results compared to the recent work. In these experiments, we use the predicate senses from (Björkelund et al., 2009) for English, German and Chinese. For Czech and Spanish, we use the predicate senses generated by (Zhao et al., 2009). Empty cells in Table 6.3 corresponds to the cases that we don’t have access to a comparable system result for a particular language. As shown in the table, our model with supervised lemmas (SLem) outperforms state-of-the-art systems in all languages except for German. We further observe that our character-based model (CModel) results in comparable performance to the supervised model that uses a range of linguistic features on English, Czech, and Chinese.
4 Direct Transfer Model

Since projected annotations often comprise noisy and partial labels, the performance of the target SRL model built using these projections heavily depends on the size of the parallel data used for annotation projection. Figure 6.2 demonstrates the performance drop (∼0.5% to 26% depending on the language) resulted from down-sampling projections (332k → 50k). Lack of access to a sizable parallel corpus such as Europarl for the majority of languages motivated us to move toward a more realistic low-resource setup where a smaller parallel corpus is used for annotation projection. In the second part of this chapter, we explore the efficacy of our model when a large parallel corpus, in the scale of Europarl, is not available. We use Bible which is available for a wide range of languages, even though limited by the size, to build the target SRL model. We try to compensate for the lack of data by employing two direct transfer techniques in our annotation projection model:

- **Polyglot training:** In a recent study, Mulcaire et al. (2018) showed that polyglot training is a useful technique for building multilingual SRL systems, where a single model is trained on the union of (heterogeneous) annotated data for different languages. Mulcaire et al. (2018) demonstrated that sharing statistical strengths of different languages through parameter sharing helps to build a multilingual model...
that outperforms the monolingual baselines. Similar to Mulcaire et al. (2018), we train our SRL model on the combination of the source and target corpora and build a cross-lingual model.

- **Related Source Embedding:** We further boost the power of cross-lingual features in our model by incorporating word-level information from the source language in the aforementioned polyglot model. Basically, we extend the input layer of the target SRL model by appending an extra embedding vector which is averaged over source embeddings that are considered semantically related to the target word. In other words, instead of directly using target predicate embedding for decoding, we use the embedding of its semantically related words from the source language. This way we guide the model towards statistical information provided by the source language annotations. Therefore, it can be seen as a way to overcome the noise existing in projections.

4.1 **Polyglot Training**

If we denote the source and target language as *en* and *de*, we train our SRL model on the combination of the following datasets: 1) $D^{en}$, the *en* corpus with manual SRL annotations, and 2) $P^{de}$ as the German corpus with projected SRL annotations. Cross-lingual features are injected to the model through cross-lingual pretrained word embeddings. We further experiment with a code-switched variation of $D^{en}$ that we call $D^{en}_{cs}$. For code-switching, given an *en-de* bilingual dictionary $B$, we replace the word by its translation from $B$. We leave the source word intact if it is not found in $B$. We then train the target SRL model on the concatenation of $P^{de}$ and $D^{en}_{cs}$. 
4.2 Related Source Embedding

If $w^{(l)}$ denotes a particular word in language $l$, we define set of translation pairs $\mathcal{W}_{de\rightarrow en}$ as:

$$\mathcal{W}_{de\rightarrow en} = \{(w^{(de)}, w^{(en)}) \mid w^{(en)} \in t(w^{(de)}, en)\}$$

where $t(w^{(l)}, l_2)$ returns the set of all words in language $l_2$ which are considered a translation of $w^{(l)}$ according to $B$. We use the translation section of Wiktionary \(^5\) to acquire $B$ using the translation parser of Acs et al. (2013). Wiktionary is a multilingual dictionary which ultimately aims to define all words across all languages. The Wiktionary parser of Acs et al. (2013) returns the list of all translation entries in Wiktionary. They further expand this list using a triangulation method which adds new translation pairs based on a common word in the bridge language. Given $E_{(en)}$ as the set of monolingual pretrained embeddings for language $en$ and $\mathcal{W}_{de\rightarrow en}$ as the set of $de\rightarrow en$ translation pairs, we generate $E_{RS}^{(de)}$ which we call related source embeddings. Each entry in $E_{RS}^{(de)}$ contains an embedding vector for a particular $w_{(de)}$. This embedding is generated by averaging over the monolingual embeddings of all $w_{(en)}$ such that $(w_{(de)}, w_{(en)})$ has been a member of $\mathcal{W}_{de\rightarrow en}$.

Finally, we represent each token to our network as the concatenation of the following embedding vectors (cf. §2):

$$x_{i,j} = [x_{i}^{re}; x_{i}^{pe}; x_{i}^{char}; x_{i,j}^{te}; x_{i,j}^{rs}] \forall i \in [1, \cdots, n]; j \in [1, \cdots, m]$$

where $x_{i,j}^{rs}$ shows the related source embedding vector found in $E_{RS}^{(de)}$. All other vectors are similar to §2. In case a $de$ word is not found in $E_{RS}^{(de)}$ (due to missing $en$ entry in $E_{(en)}$), we represent $x_{i,j}^{rs}$ with its corresponding monolingual $de$ embedding.

In addition to the above modification of the input layer in the SRL model of §2, we use the generated related source embeddings in the decoder layer as well. We modify $W_{j,r}$ (the parameter matrix used to jointly encode the information of the role $r$ and the $j$th predicate)

\(^5\)https://www.wiktionary.org
as follows:

\[ W_{p,r} = \text{RELU}(U[u_{p}^{s}, v_{r}]) \]

where \( U \) is a parameter matrix, \( v_{r} \in \mathbb{R}^{d_r} \) is a randomly initialized role embedding and \( u_{p}^{s} \in \mathbb{R}^{d_w} \) is the related source embedding of predicate \( p \). In other words, instead of directly using target predicate representation for decoding, we use its related source embedding. This way we guide the model towards statistical information provided by the source language annotations, thereby, it can been seen as a way to overcome the noise existing in projections.

### 4.3 Pretrained Cross-lingual Word Embeddings

The backbone of a cross-lingual model is the pretrained embeddings that map words from distinct vocabularies, here tokens of different languages, to a shared representation space. This provides the opportunity for a cross-lingual model to learn from shared linguistic and statistical patterns in different languages. We produce cross-lingual embeddings by code-switching monolingual corpora in different languages: input to our method are 1) two monolingual corpora for the source (en) and target language (de) and 2) bilingual dictionary. For each word in the monolingual corpora, we generate a random number. If the random number is less than a certain threshold (0.3 in our experiments), we replace the word by its translation provided by the bilingual dictionary. We then run word2vec (Mikolov et al., 2013a) on the code-switched corpora to obtain cross-lingual embeddings. We used word2vec with its default parameter configuration to generate vectors of size 100.

### 4.4 Experiments

**Dataset and Tools**  In our experiments, we consider English as the source language and build SRL model for the following target languages: German (de), Spanish (es), Finnish (fi), French (fr), Italian (it), Portuguese (pt) and Chinese (zh). The English supervised SRL system used in our experiments is similar to the supervised model described in our annotation projection experiments. For brevity, we skip details of that model and refer the reader to
§2 for more details. Likewise, the predicate identification and disambiguation modules used for direct transfer experiments are similar to the ones used for annotation projection experiments (§3). As our parallel data, we use the Bible corpus of Christodouloupoulos and Steedman (2015). We use Giza++ (Och and Ney, 2003) to generate intersected word alignments from English to all target languages. For English pre-trained embeddings, we use the publicly available embeddings of Ling et al. (2015). Cross-lingual embeddings are generated using the latest Wikipedia dump for each source and target language pair. The bilingual dictionary used for code-switching is generated from Bible alignments. We use the Wiktionary translation parser tool introduces in Acs et al. (2013) to extract $W_{de \rightarrow en}$ for all target languages. We additionally use the provided triangulation utility to combine the extracted translation pairs and generate new ones.

We use Proposition Bank 3.0 (PropBank for brevity) Palmer et al. (2005) to train the English SRL system. We convert the span-based frame annotations of PropBank into dependency-based annotation by assigning the label of each argument span to the syntactic head of that argument. English PropBank is further used for polyglot training as well. ProbBank corpus has 272K sentences which is considered considerably larger than the Bible corpora. Thus, to keep data balance across the source and target language, we use a random sample of PropBank sentences for polyglot training models: for each target language, the sample size is set to $1.5 \times$ (size of projected corpus). We use the Bible alignment dictionary to generate code-switched ProbBank data. We use UDPipe (Straka and Straková, 2017) pre-trained models to generate lemma, POS tags and dependency trees for all supervised SRL systems throughout this paper. For annotation projection, we set the alignment density to 0.8 for all languages which results in the statistics shown in Table 6.4. This helps maintain sentences with higher quality alignments in the projected corpus.

**SRL Parameters** All model parameters are tuned on the development set of German. We employ the same parameters for other languages. We use Dynet library Neubig et al. (2017)
<table>
<thead>
<tr>
<th>Lang.</th>
<th>#Sent.</th>
<th>#Tokens</th>
<th>#Types</th>
<th>#Pred.</th>
</tr>
</thead>
<tbody>
<tr>
<td>de</td>
<td>6K</td>
<td>132K</td>
<td>6K</td>
<td>16K</td>
</tr>
<tr>
<td>es</td>
<td>4K</td>
<td>86K</td>
<td>6K</td>
<td>8K</td>
</tr>
<tr>
<td>fi</td>
<td>9K</td>
<td>188K</td>
<td>13K</td>
<td>24K</td>
</tr>
<tr>
<td>fr</td>
<td>8K</td>
<td>202K</td>
<td>9K</td>
<td>20K</td>
</tr>
<tr>
<td>it</td>
<td>20K</td>
<td>528K</td>
<td>16K</td>
<td>46K</td>
</tr>
<tr>
<td>pt</td>
<td>2K</td>
<td>50K</td>
<td>5K</td>
<td>1K</td>
</tr>
<tr>
<td>zh</td>
<td>7K</td>
<td>212K</td>
<td>2K</td>
<td>16K</td>
</tr>
</tbody>
</table>

Table 6.4: Size of the Bible projected data for different languages

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word embedding ((d_w))</td>
<td>100</td>
</tr>
<tr>
<td>Char. embedding ((d_c))</td>
<td>50</td>
</tr>
<tr>
<td>LSTM hidden state ((d_h))</td>
<td>512</td>
</tr>
<tr>
<td>Input pred. char. vector ((d_{ic}))</td>
<td>100</td>
</tr>
<tr>
<td>Role embedding ((d_r))</td>
<td>128</td>
</tr>
<tr>
<td>Decoder pred. char. vector ((d_{d}))</td>
<td>128</td>
</tr>
<tr>
<td>Char LSTM hidden state ((d_{ch}))</td>
<td>100</td>
</tr>
<tr>
<td>Minibatch size</td>
<td>(~1K) tokens</td>
</tr>
<tr>
<td>Learning rate (Adam)</td>
<td>0.001</td>
</tr>
</tbody>
</table>

Table 6.5: List of CModel parameters

...to implement our neural model. The objective function of our model is log-likelihood and is optimized using the Adam algorithm (Kingma and Ba, 2014). Table 6.5 shows parameters of our model. The depth of BiLSTM network used to generate character models is set to 1. All other BiLSTM networks in our model have 3 layers.

### 4.5 Results

We evaluate our model using the test sets provided in the Universal PropBank. All languages in the Universal PropBank comprises the same annotation scheme and frameset that further facilitates uniform evaluation across all languages. Table 6.6 shows labeled F1–scores of different direct transfer models for different languages. First of all, we observe that direct transfer (polyglot training, related source embeddings or a combination of them) outperforms the basic SRL model trained on the Bible projections (first row) for all languages.
Table 6.6: Labeled F1-scores of our direct transfer model for different target languages: de, es, fi, fr, it, pt, zh on the Universal PropBank test set compared to the model trained on Bible projected annotations.

Results further demonstrate that the direct transfer yields to larger improvements for some languages compared to others (∼8% for Chinese vs. ∼1.5% for Italian). Interestingly, results show that code-switching does not yield to noticeable improvements across languages: we just observe a slight increase for Finish and Portuguese, however code-switching results in almost equivalent results as the baseline model for all other languages, such as German, Spanish, French and Chinese. Furthermore, we observe that related source embeddings outperform the baseline model, as well as other variations, when combined with polyglot training in Finish, Portuguese and Chinese.

5 Related Work

There has been a great deal of interest in using transfer methods for SRL by different techniques such as enhancing the quality of projections (Padó and Lapata, 2005, 2009), joint learning of syntax and semantics (Van der Plas et al., 2011; Kozhevnikov and Titov, 2013b), and iterative bootstrapping to learn a robust model from erroneous projections (Akbik et al., 2015; Aminian et al., 2017). Previous work presumes the availability of a wide range of supervised lexico-syntactic features for the target language. Consequently, their performance heavily relies on the accuracy of the available tagging tools. For instance, Akbik et al. (2015) reports lower argument precision for languages that do not have accurate syntactic parsers such as Arabic and Hindi. We take one more step toward a more realistic low-resource scenario and build a cross-lingual SRL system that does not need any supervised features for making robust predictions in the target language (Aminian et al., 2019). In contrary to the
previous studies, our work considers building a cross-lingual SRL system without having any supervised features for the target language.

It is worth emphasizing, in the realm of supervised SRL methods, however, there have been several efforts to build SRL models that do not need a wide range of linguistic features (specifically syntactic features) (Marcheggiani et al., 2017; Zhou and Xu, 2015b; He et al., 2017, 2018; Cai et al., 2018; Mulcaire et al., 2018). In a more recent study, Mulcaire et al. (2018) proposed a polyglot SRL system that benefits from the similarities between the semantic structures of different languages to improve monolingual SRL. All these studies, however, assume the availability of semantically annotated datasets for the target language, thus making them non-applicable to low-resource languages. Additionally, one major bottleneck in annotation projection is the lack of universal annotation schemes. This problem has been solved for POS tagging (Petrov et al., 2011) and dependency parsing (McDonald et al., 2013; Nivre et al., 2017) by providing gold-standard universal annotation. Recent work (Akbik et al., 2015) have considered developing a universal proposition bank by using semi-automatic methods. Although relying on semi-automatic annotations is not ideal, the universal proposition bank is the best available dataset for conducting experiments on annotation projection. We use this dataset in our experiments presented in (Aminian et al., 2019).

6 Conclusion

We described a method for cross-lingual transfer of dependency-based SRL systems via annotation projection. Our model is agnostic to linguistic features leading to a robust model that can be trained on projected text on a target language without annotated data. We have shown that our model achieves comparable performance in annotation projection and also supervised SRL. In addition to improving the performance of our model with the current setting, we explored two direct transfer techniques, polyglot training and employing related source embeddings in the absence of large parallel corpora.
Chapter 7: Multitasking for Cross-lingual Semantic Dependency Parsing

1 Introduction

Broad-coverage semantic dependency parsing (SDP) discovers the underlying semantic structure of a sentence in the form of predicate-argument dependencies. Unlike dependency parsing that resembles a tree over the syntactic structure of the sentence, SDP creates a directed acyclic graph in which each word can be the semantic argument of multiple predicates in the sentence. In addition to the relation to syntactic dependency parsing, SDP is closely related to semantic role labeling. However, in contrary to SRL that mainly focuses on verbal and nominal predicates, SDP produces semantic analysis for all content words in the sentence, thereby some semantic phenomena such as comparatives, possessives and various types of modification that convey the sentence meaning and have not been addressed in SRL, are extensively analyzed by the SDP target representations.

Semantically annotated datasets with SDP structure are limited to the dataset introduced in the SemEval-2014 and then extended in SemEval 2015 shared tasks. SemEval dataset provides annotated SDP corpora for only three languages namely English, Czech and Chinese. SDP data scarcity leads us to explore cross-lingual methods for improving existing supervised systems or building SDP models for languages that do not have any annotated resources. The ultimate goal of this chapter is building SDP model for languages that do not have any SDP dataset. In continuation to the approach we pursued in previous chapters to address word sense divergence and low-resource SRL, we use annotation projection for transferring supervised SDP annotations from a high-resource language namely English to the target language of interest, which is Czech in our work.

Here, we assume that we have access to a sizable parallel data that can be used to provide links between the high-resource language and a low-resource one. In this chapter, we consider the low-resource setting where we may have access to a supervised POS tagger and
dependency parser for the target language, however, no manually annotated data with SDP labels is provided. In a vanilla system, the projected SDP annotations from the high-resource language are used to train the target SDP model. In this chapter, we aim to further enhance the vanilla SDP model trained on projections by using multitasking to aggregate syntax in the target SDP model. Given the large amount of structural and statistical similarities existing between syntactic and semantic dependency parsing, we explore the effectiveness of aggregating supervised syntax, as an auxiliary task, in a multitasking framework. We hypothesize that multitasking, even though simple, will yield in particular improvements in the target SDP model. In other words, we expect that similarities between syntactic and semantic dependencies act as a useful inductive bias and help the target model addresses the noisy/missing projections. We additionally explore the efficacy of contextualized word representations extracted from a pre-trained BERT model to further improve our model.

In this chapter, we first describe the details of the supervised semantic dependency parser module used in our experiments. We then describe the details of our model, experimental setting and results for each of the above directions.

2 Supervised Semantic Dependency Parsing

The supervised semantic dependency parser used in our cross-lingual model is similar to the system introduces in (Dozat and Manning, 2018). In order to make the original model scalable to our computational resources, we performed modifications in the input representation and parameters used in the model. Preliminary experiments confirmed equivalent performance on the development data to the original model. Following we describe the supervised SDP model used in our experiments with details.

Given a sentence \(s = [s_i]_{i=1}^n\) with \(n\) tokens, we represent each token \(s_i\) with the vector \(x_i\) by concatenating its word \((x^w_i \in \mathbb{R}^{d_w})\) and POS tag \((x^t_i \in \mathbb{R}^{d_t})\) embeddings:

\[
x_i = [x^w_i; x^t_i] \quad \forall i \in [1, \cdots, n]
\]
$x_i^{we}$ itself is computed by summing over the following embedding vectors:

$$x_i^{we} = x_i^{re} + x_i^{pe} + x_i^{ce}$$

where:

- $x_i^{re} \in \mathbb{R}^{d_w}$ is a randomly initialized word embedding vector;
- $x_i^{pe} \in \mathbb{R}^{d_w}$ is a fixed external pre-trained word embedding vector;
- $x_i^{char} \in \mathbb{R}^{d_w}$ is character level representation of each token $s_i$ and is obtained by running a one-layer BiLSTM on top of $s_i$ characters.

The recurrent representation of each token in $s$ is obtained by running a deep three-layered BiLSTM on top of the sentence tokens. We represent hidden representation of $s_i$ with $h_i$ which is the last hidden state of the deep BiLSTM for $s_i$:

$$h_i = \text{BiLSTM}(s_1^n; i) \in \mathbb{R}^{d_h}$$

The hidden representation of each token ($h_i$) is then factorized into four vectors by feeding $h_i$ into four distinctly parameterized feed-forward neural networks:

$$h_i^{(e,d)} = \text{FNN}^{(e,d)}(h_i) \quad \forall e \in \{\text{edge, label}\}; d \in \{\text{dep, head}\}$$

The factorized hidden vectors are then used to compute bilinear scores for the edges and labels separately:

$$score_{i,j}^{(e)} = h_i^{T(e,dep)}W_{e}h_j^{(e,head)} \quad \forall e \in \{\text{edge, label}\}$$

To decide if there is an edge between $s_i$ and $s_j$, the unlabeled parser uses a sigmoid function that simply labels $s_j$ as the head of $s_i$ if $score_{i,j}^{(edge)} \geq 0$. Given an edge between $s_i$ and $s_j$,
the probability of having a certain label for the edge is then computed using:

\[
p(label|(i,j)) = \text{Softmax}_{label}(\text{score}_{i,j}^{(label)})
\]

The model is trained by back-propagating the linear interpolation of the edge and label loss values computed as:

\[
loss = \lambda loss_{label} + (1 - \lambda) loss_{edge}
\]

Figure 7.1 gives the graphical depiction of the supervised system we just described. Moreover, we use word dropout for dropping word and POS tag embeddings (independently) from the input layer. We further utilize dropout in the main BiLSTM network as well as edge/label FNN models.

3 Multitasking with Syntax

The main goal of multitask learning is benefiting from structural or statistical similarities found in one or more auxiliary tasks to improve the model learned for the target task. Often
times, auxiliary tasks are chosen amongst closely related tasks. This is indeed intuitively reasonable because related tasks can provide stronger bias that supports a specific pattern in the main task. Since SDP target representations are closely tied to the underlying syntactic analysis of the sentence, syntax can be considered as an effective auxiliary task. In other words, relevant syntactic features help the SDP model focus its attention on the features that matter the most. Multitasking with syntax would even make more sense for building an SDP model based on annotation projection because of the noise existing in the projections (due to noisy/missing alignment). In other words, we expect that the relevant syntactic features, provided by a supervised syntactic parser, help the SDP model rule out noisy instances and focuses on more relevant features. Although multitasking with syntax seems to be a simple solution to improve the target SDP system, to the best of our knowledge, none of the previous studies considered aggregating syntactic dependencies in a multitasking framework to improve the target SDP model.

![MTL models with shared FNN](image)

Figure 7.2: MTL models with shared FNN: (a) MTL model with a single BiLSTM shared across tasks, (b) MTL model with a task-specific BiLSTM beside the shared BiLSTM

In order to find out the exact parameter sharing structure that leads to the best SDP model for the target language, we tried different parameter sharing variations in our model from shared recurrent layer (BiLSTM) to shared feed-forward neural networks (FNN). All parameter sharing variations have their own task-specific BiLinear and classification layers.
Figure 7.2a shows the model that both BiLSTM and FNNs are shared across the tasks (sharedFNN). Figure 7.2b shows another variation that consists of a task-specific BiLSTM beside the shared BiLSTM (SharedFNN+SepRNN). The ultimate representation of each token in this model is obtained by concatenating the output of the shared and task-specific BiLSTMs. The FNN layer has remained intact and is shared across syntax and semantics. Figure 7.3a shows the network with separate FNNs for each task (SepFNN). Lastly, Figure 7.3b shows the model with task-specific BiLSTM in the recurrent computation layer, in addition to separate FNNs for the main and auxiliary task (SepFNN+SepRNN).

We compute the loss value for the semantic module ($loss^{sem}$) by interpolating semantic label loss and semantic edge loss using:

$$loss^{sem} = \lambda loss^{sem.label} + (1 - \lambda) loss^{sem.edge}$$

Moreover, syntactic loss ($loss^{syn}$) is computed by averaging syntactic edge loss and syntactic label loss values:

$$loss^{syn} = \frac{loss^{syn.label} + loss^{syn.edge}}{2}$$

The overall loss value for the MTL model is computed by interpolating semantic and syntactic loss by defining a particular MTL interpolation coefficient called $\lambda^{MTL}$ and tuned
<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word embedding ($d_w$)</td>
<td>100</td>
</tr>
<tr>
<td>POS tag embedding ($d_t$)</td>
<td>100</td>
</tr>
<tr>
<td>BERT features ($d_b$)</td>
<td>768</td>
</tr>
<tr>
<td>LSTM hidden state ($d_h$)</td>
<td>600</td>
</tr>
<tr>
<td>LSTM hidden layer</td>
<td>3</td>
</tr>
<tr>
<td>FNN edge/label size</td>
<td>600</td>
</tr>
<tr>
<td>Word/POS dropout</td>
<td>20%</td>
</tr>
<tr>
<td>BiLSTM dropout</td>
<td>25%</td>
</tr>
<tr>
<td>FNN Edge dropout</td>
<td>25%</td>
</tr>
<tr>
<td>FNN Label dropout</td>
<td>33%</td>
</tr>
<tr>
<td>Inter. coef. ($\lambda$)</td>
<td>0.025</td>
</tr>
<tr>
<td>MTL Interpolation coef. ($\lambda_{MTL}$)</td>
<td>0.975</td>
</tr>
<tr>
<td>Learning rate (Adam)</td>
<td>0.001</td>
</tr>
<tr>
<td>Minibatch</td>
<td>~1K tok.</td>
</tr>
</tbody>
</table>

Table 7.1: Our SDP model parameters

during development experiments:

$$\text{loss}_{MTL} = \lambda_{MTL}\text{loss}_{sem} + (1 - \lambda_{MTL})\text{loss}_{syn}$$

3.1 Adding BERT Features

We further investigate the efficacy of using contextualized word embeddings in our model by expanding the input representation by adding dynamic embeddings extracted from a pre-trained BERT model ($x^{be} \in \mathbb{R}^{d_b}$). Thus, the modified representation of the word $s_i$ will be:

$$x_i = [x_i^{we}, x_i^{te}, x_i^{be}] \quad \forall i \in [1, \ldots, n]$$

where $x_i^{we}$ and $x_i^{te}$ are calculated as described in §2.

4 Experiments

In our experiments, we consider English as the source language and Czech as the target language. We use evaluation sets (in-domain and out-of-domain) provided by the SemEval
2015 to evaluate our models. Both English and Czech annotation are based on PAS semantic dependency formalism. As parallel data, we use Europarl parallel corpus (Koehn, 2005). We use Giza++ \(^1\) (Och and Ney, 2003) with its default configuration to obtain word alignments. One-to-one alignments are obtained by intersecting source-to-target and target-to-source directions. For English pretrained embeddings, we use embedding vectors generated by the structural skip-gram model of (Ling et al., 2015). Czech pretrained embeddings are generated by running Word2vec continues skip-gram model on the latest Wikipedia dump. We use UDpipe (Straka and Straková, 2017) pretrained models to produce automatic POS tags in our experiments. We use the biaffine dependency parser of (Dozat and Manning, 2016) to generate supervised syntactic parses in our MTL experiments. Table 7.1 shows the parameters of our supervised SDP model which mainly follows the hyperparameters used in (Dozat and Manning, 2018) except the character LSTM and linear transformation layers\(^2\). For extracting English contextualized embeddings, we used the cased pretrainend BERT model with 12 hidden layers and 12 heads. Czech contextualized embeddings also were extracted using the cased pretrained multilingual BERT model with 12-layers and 12-heads.

5 Results

Graphs depicted in Figure 7.4 show labeled and unlabeled F1 scores of different MTL models trained on projections with different densities. In order to assess the minimal degree of supervision required by different MTL models, we conduct experiments using projections samples with different density medians. The x-axis in Figure 7.4 denotes the median density of the sample used for training the model. Therefore, median 0.9 for a sample containing 80K sentence denotes that 40k sentences in the training data have density less than 0.9 and

---

1 We tried other automatic alignment tools such as fast align as well but did not observe any gain in the overall performance of our model.

2 During development experiments, we found out that linear transformation of characters does not play a significant role in the performance, thereby we excluded this part for simplicity.
the other 40k sentences comprises sentences whose projection density is more than 0.9. We use the density criteria defined in §3 to obtain projection density using:

$$\text{density}^{(i)} = \frac{\sum_{j=1}^{l'_{i}} I(a_{j}^{(i)} > 0)}{l'_{i}}$$

where $l'_{i}$ is the length of the $i$th target sentence in parallel data, $a_{j}^{(i)}$ is the alignment index for the $j$th word in the target sentence, and $I(a_{j}^{(i)} > 0)$ is an indicator for a non-NULL alignment.

We perform evaluations against both in-domain (left column) and out-of-domain (right column) test sets provided by SemEval 2015 shared task (Oepen et al., 2015). All evaluations are performed using the official scoring script provided by the shared task. We compare the
MTL results with the Single baseline model: the SDP model that is directly trained on the projected annotations, without aggregating syntactic information as in the MTL setting. Our preliminary experiments confirmed that using the full set of projections (612k sentences) to train the model results in equivalent performance to the model trained on a sample with reasonable size (80K in our experiments). Therefore, given the fact that the training process is much faster when the 80K sample is used, we train all models (including the Single baseline) on a random sample of 80K sentences from the projected corpus.

Comparing the labeled F1–scores for different MTL models, we observe that all MTL models outperform the Single baseline, regardless of the architecture and data density used to train the model. We additionally observe that MTL models yield in a larger increase on the out-of-domain test set compared to in-domain test set. This illustrates the particular power of MTL to improve the target SDP model in truly low-resource settings where (enough) in-domain training data might not be available. We further observe that increasing projection density have different effects on different MTL models. For instance, we note a sharp decrease of labeled scores after density 0.7 on in-domain test for shared FNN model, while the same density change results in a particular improvement for the task-specific FNN model. This is while we observe a modest increase of the labeled scores on out-of-domain test for 3 out of 4 MTL models. We also observe that model structure and size of parameters specifies the optimal density to avoid overfitting, thereby the optimal density value is higher for models with fewer parameters (such as Single). SepFNN+SepRNN model almost always underperforms other MTL models due to the over-parameterized model. Overall, comparing labeled F1–score results in Figure 7.4 show that MTL model with task-specific FNN and shared BiLSTM (SepFNN) across tasks outperforms all other MTL architectures on both in-domain and out-of-domain evaluation. Intuited by our observations, we surmise that the particular characteristic of the PAS target representation is one of the main reasons for the superior performance of SepFNN model; some of the semantic edges in the PAS representation are directed in the opposite direction of the syntactic dependency existing
between those words. For instance, in the PAS labeling scheme, the word *position* in the noun clause “*the position*” is labeled as the child for the determiner *the* with SDP label *det_ARGL* while the syntactic edge between these two words is in the exact opposite direction. Therefore, having separate FNNs leads to superior performance compared to the model that comprises shared FNNs.

Graphs depicted on the second row in Figure 7.4 show unlabeled F1 scores for different MTL models compared to the Single baseline. Unlike labeled results, increasing projection density results in either a slight increase or equivalent unlabeled F1 scores. But similar to the conclusion made for labeled F1–scores, the MTL model with shared BiLSTM and task-specific FNN (SepFNN) gives the best performance compared to other MTL models.

**Adding BERT Features** We additionally conducted some experiments to assess the efficacy of using contextualized word embeddings extracted from a pre-trained BERT model.
in our MTL model. Graphs depicted in Figure 7.5 show performance of the Single baseline in addition to different MTL models on in-domain and out-of-domain test sets when BERT features are added for input representation. All models are trained on a sample with density median of 0.8. We closely assess the efficacy of aggregating BERT features in the target SRL model by comparing performance of the following models:

- **no BERT**: this model involves the vanilla SDP model (with multitasking) that is trained on projected annotations without aggregating any BERT features in the source or target model,

- **source BERT**: in this model, we only use BERT features to train the source SDP model. The target model is trained on the projected annotations but no BERT feature is used on the target side,

- **source + target BERT**: in this model, we incorporate representations extracted using BERT for building source and target SDP model.

As shown in Figure 7.5, utilizing contextual features extracted from BERT pre-trained model on the source + target model results in dropping model’s labeled F1–score for the majority of MTL models on both in-domain and out-of-domain test set compared to the no BERT model. The only exception is sharedFNN model with source + target BERT which performs slightly better than the no Bert model on the in-domain test with. Surprisingly, the same trend is not observed when BERT features are used only on the source side (source BERT). In fact, in some MTL models, source BERT improves labeled F1–score while aggregating source + target BERT underperforms the no BERT model. We surmise that this particularly happens because of the lower quality of features extracted from pre-trained BERT model for the target language (Czech) compared to the source language (English). Although aggregating source + target BERT does not succeed in improving the labeled F1–scores, we observe particular improvements in the unlabeled F1–scores resulted from using source + target BERT in 3 out of 4 MTL models (second row in Figure 7.5). We also
observe that source BERT when utilized in the single model significantly outperforms the no BERT and source + target BERT models, however lags behind the MTL models.

6 Analysis

System Performance on Different Semantic Dependency Lengths We further analyzed the performance of our best performing MTL model (SepFNN) on different semantic dependencies. Figure 7.6 illustrates labeled precision of SepFNN model compared to the Single and Supervised model for different semantic dependency lengths. Length of a semantic dependency is generally shown with the number of tokens located between the semantic head and its dependent. Numbers shown above each plot denotes the improvement resulted from the SepFNN model compared to the Single model. Interestingly, we observe that the MTL model yields in particular improvements in the longer semantic dependencies compared to the shorter ones, such that the MTL precision for semantic dependencies with length $\geq 10$ is noticeably close to the supervised results.

Correlation with Different Syntactic Relations Figures 7.7 and 7.8 show the percentage of improved and degraded semantic dependencies resulted from the best performing MTL setup (SepFNN) compared to the baseline (Single). The x-axis denote the syntactic relation
corresponding to the semantic dependencies, either improved or degraded. As shown in the figures, some syntactic relations yield to considerable improvements in semantic dependencies. The sources of this improvement is two folds: first, there is a direct correspondence between the syntactic and semantic role in some dependencies such as `conj` and `iobj`, thereby, injecting information about these relations enhances predictions made by the SDP model. Second, enhancement obtained from other syntactic relations such as `nmod`, `nsubj`, `obj` and `amod` is in fact because these relations are more frequently observed during training. Therefore, the most degraded results are observed in these syntactic relations as well. Intuited by these observations, we surmise that using a weighting algorithm to prioritize some syntactic relations over the others will help the MTL model obtain even better results.

7 Related Work

Datasets released during SemEval shared tasks (Oepen et al., 2014, 2015; Che et al., 2016) provides training data for building supervised semantic dependency parsing (SDP)
systems for three languages namely English, Czech and Chinese. Since this is the only SDP dataset released so far, studies performed to build SDP models mainly limited to semantic dependency parsers for these languages (Du et al., 2015; Che et al., 2016; Chen et al., 2018; Almeida and Martins, 2015; Wang et al., 2018; Dozat and Manning, 2018; Kurita and Søgaard, 2019; Stanovsky and Dagan, 2018). This SDP dataset consists of three main semantic formalisms, also known as target representations, that are slightly different in the way they determine and label semantic relations. These representations are usually referred to as DM (DELPH-IN minimal recursion), PSD (Prague semantic dependencies) and PAS (Enju predicate–argument structures) (c.f. §3 for more details). Besides these three target representations, there are other semantic formalisms, all of which intend to produce sentential semantic analysis by generating a graph covering (content) words in the sentence (such as Universal Conceptual Cognitive Annotation (UCCA) (Abend and Rappoport, 2013), Abstract Meaning Representation (AMR) (Banarescu et al., 2013a) and Frame-semantic parsing (Baker et al., 1998)).

Motivated by the fact that each of these formalisms cover different aspects of sentential
there has been a line of studies to use multi-task learning in order to improve single-task baselines (Peng et al., 2017, 2018; Hershcovich et al., 2018; Kurita and Søgaard, 2019). Peng et al. (2017) is one of the early studies in this task that try to improve single-task SDP baseline by exploiting data provided for different SDP target representations via two multi-task setups: in the first setup, they use a multi-task setup with shared recurrent layer across different target representations while each task has its own classifier. As the second setup, they use higher-order shared structures to model interactions between different tasks. In continuation of this line of studies, Peng et al. (2018) introduce a multi-task system that is jointly trained on disjoint datasets: one with SDP annotations and the other with frame semantic annotations. They model uncommon semantic structures as latent variables. In another work, Hershcovich et al. (2018) applies multi-task learning over different semantic annotation formalisms including UCCA, AMR, and DM as well as universal dependencies. Their model consists of a shared BiLSTM in addition to the task-specific BiLSTMs in the recurrent computation layer. In a more recent study, Kurita and Søgaard (2019) describe an SDP model that combines the arc-factored and graph-based parsing algorithms. They further show that multi-task learning across three SDP formalisms outperforms the single-task results.

Multitasking with syntax has been extensively studied in other sentential semantic parsing tasks such as semantic role labeling (Lluís et al., 2013; Swayamdipta et al., 2018). Lluís et al. (2013) describe a joint arc-factored parser that uses dual decomposition for jointly decoding syntactic and semantic structures. In another work, Swayamdipta et al. (2018) introduce a system for discovering sentential semantic structure by incorporating syntactic information in a multi-task framework. They demonstrate their performance in SRL (both frame SRL and PropBank style SRL) and coreference resolution. To the best of our knowledge, none of the previous work considers integrating syntactic inductive bias in a multitasking framework in any cross-lingual SDP models and our work is the first work that aggregates syntax to enhance a cross-lingual SDP model. Moreover, to the best of our
knowledge, this is the first work that aims to build an SDP model using cross-lingual transfer systems with the assumption that there is no annotated data available for the target language of interest.

8 Conclusion

In this chapter, we described a cross-lingual SDP model based on annotation projection that does not use any annotated SDP data in the target language. We enhance the target SDP model trained on projected SDP annotations by incorporating syntax in a multitasking framework. We show that our multitasking model outperforms the baseline trained on the target SDP projections on both in-domain and out-of-domain SemEval test sets for Czech. Our detailed analysis shows that the proposed multitasking approach yields in particular improvements for long-distance semantic dependencies.
Chapter 8: Conclusion

In this dissertation, we addressed the problem of developing semantic analysis models in different low-resource scenarios using the semantic knowledge transferred from a high-resource language. We developed cross-lingual semantic models to address both word-level and sentence-level semantics. At the word-level, we tackled the issue of word sense divergence. We particularly targeted the sense divergence phenomenon in one of the informal spoken varieties of Arabic, Egyptian, as a low-resource language variant. At the sentence-level, we explored different aspects of building and enhancing a cross-lingual semantic role labeling (SRL) and semantic dependency parsing (SDP) model using annotation projection under different low-resource scenarios.

At the word-level, we developed an unsupervised model for identifying and disambiguating words with sense divergence using lexical semantic evidence provided by parallel corpus. We particularly used an Egyptian-English SMT system to extrinsically evaluate our model and demonstrated the efficacy of transfer approaches to improve the SMT lexical choice for words with sense divergence. We additionally demonstrated that our transfer-based approach yields in the performance that is fairly close to a knowledge-oriented approach which has access to annotated data and supervised tools. As intuited by our analysis, incorporating contextual similarity information yields in superior performance in sense divergence identification and disambiguation compared to the case that we only rely on parallel cues for making predictions. Notwithstanding the significant improvements yielded by aggregating the sense divergence module in the SMT system, we observed sentences with correct sense OOD identification and disambiguation which did not lead to an improved translation output, basically due to the absence of a proper entry in the pre-generated phrase tables. Based on our analysis, contextual information provided by the disambiguation module played an important role in many difficult examples of sense divergence including words whose
diacritics have been omitted. This further highlights the importance of contextual hints in identifying and disambiguating words with sense divergence.

From another point of view, the proposed model can be conveniently used as a pre-processing step before the decoding phase of the SMT model, thereby it does not add any computational complexity to the SMT model. Since the model does not depend on the availability of any in-domain training data, it is not limited to any domain and can be extended to any other language variants with minimal effort. Furthermore, we developed an algorithm that can be used to expand the multilingual correspondents exploited in the disambiguate phase. Our multilingual correspondent learning algorithm can be used for automatic augmentation of multilingual lexica and is capable of detecting inconsistencies in the lexicon entries and possibly providing or suggesting candidates to replace them.

Performance of semantic analysis models built using annotation projection is usually affected by the considerable amount of noise in projections. We developed a cross-lingual SRL model based on annotation projection that particularly focuses on identifying noisy projections during the training phase. We utilized a customized objective function that is sensitive to the noise identified through translation and syntactic information in projections. We used the newly defined objective functions to train the target model. We demonstrated that our model yields in particular improvements in filtering noisy training instances and overall performance of the target SRL model. Unlike previous studies that use manually-defined rules to filter projections, we benefit from information obtained from projection sparsity and syntactic similarity to weight projections. As inferred by the results, this customized cost function reaches its best performance when aggregated in an iterative bootstrapping framework which relabels all training instances at each iteration (instead of making predictions for instances without a projected label). In fact, relabeling helps the model gradually reduces the noise existing in projections. Based on our analysis, using syntactic cues to weight projection instances helps the model labels some of the instances that were dismissed during projection, thereby increasing the system recall.
We also developed a cross-lingual SRL model based on annotation projection that is agnostic to linguistic features and solely relies on the character representation of words. We showed that our character-based model results in superior performance compared to the baselines that use a range of fully-fledged supervised features from supervised lemma and POS tags to unsupervised stems. Results also demonstrate that the power of character models highly depends on the morphological characteristics of the language. For instance, our character-based model outperforms the SRL model that incorporate supervised lemma and unsupervised stems on all languages except German and Spanish. We further explored the assumption of having access to a large parallel data by using the Bible, a sizable parallel corpus available for the majority of languages. Our experiments reaffirmed that the performance of cross-lingual SRL model built using annotation projection heavily depends on size of the parallel corpus used for projection, however, in the absence of a large parallel corpus, we can devise different direct transfer techniques to further enhance the target model. In our work, we used two techniques to benefit from statistical strengths of the source language to guide the target model; (a) polyglot training, and (b) aggregating related source embeddings in the model. As expected, direct transfer, in the form of polyglot training, related source embeddings or a combination of these two, leads to superior performance compared to the single model trained on projections. However, the degree to which each of these techniques is useful highly depends on the language (\(\sim 8\%\) improvement for Chinese vs. \(\sim 1.5\%\) improvement for Italian).

We explored the effects of aggregating supervised syntactic information in a multitask learning (MTL) framework to improve a cross-lingual semantic dependency parser (SDP) based on annotation projection. Our results show that the MTL model outperforms the baseline without multitasking, regardless of the architecture and data density used to train the model. We additionally observe that all MTL models yield in larger increase on the out-of-domain test set compared to the in-domain test set. This illustrates the particular power of MTL to improve the target SDP model in truly low-resource settings where (enough)
in-domain training data might not be available. Our observations show that the specific MTL structure that is used and the size of the parameters used in the model is an important factor that specifies the optimal projection density to avoid overfitting. Therefore, the optimal density value is higher for models with fewer parameters. Additionally, our analysis shows that the MTL model yields in particular positive effect on the longer semantic dependencies compared to the shorter ones, such that the MTL precision for semantic dependencies with length $\geq 10$ is noticeably close to the supervised precision.

Intuited by the observations derived from various transfer experiments on different languages, we draw the following insights about different language families and its role in the level of supervision required for modeling semantics in each language:

- We observe that cross-lingual transfer models yield in superior performance on the languages belonging to the Romance subtree of the Indo-European language family including Spanish, French, and Portuguese compared to the Germanic languages such as German. As inferred by our results, the former languages particularly respond well to character models and require less supervision (in terms of annotated resources and features) for comparable performance. We further observe that, despite distinct morphological characteristics of the Chinese, its performance is very similar to the Romance languages.

- Accordingly, languages such as Spanish, French and Portuguese (Romance family) are less sensitive to the amount of parallel data used for projection while languages such as German (Germanic) need larger parallel resources. Therefore, we observe that the performance of cross-lingual models built for the later languages significantly drops when smaller parallel data is used in the process of annotation projection.

Guided by the results and analysis conducted in this dissertation, we propose the following directions for future work:

- Given the importance of contextual information in addressing sense divergence, we
expect that incorporating deep contextualized embeddings such as BERT (Devlin et al., 2019) and XLM (Lample and Conneau, 2019), in the form of monolingual or cross-lingual embeddings, leads to significant enhancements in identification and disambiguation of word sense divergence. Likewise, we expect that contextual embedding features, either monolingual or multilingual, further improve cross-lingual SRL or SDP models. Therefore, one interesting line of research to continue the current work is to explore the effectiveness of contextualized features to address issues particularly related to low-resource scenarios.

• We explored transferring semantics from one high-resource language, namely English, to different low-resource languages. One interesting direction to peruse is to transfer semantic knowledge embedded in multiple source languages to build the target model. The target model can use various polling strategies or ranking criteria to rank these projected annotations and use them to further enhance the model. Another way to incorporate annotations transferred from multiple source languages is to use one language as the bridge to transfer annotations from one source language to another language.

• All methods discussed so far are bound by the availability of parallel corpora. One of the future directions to combat this limitation is to look into the usage of comparable corpora for projecting annotations across languages. Semantic information can be transferred across languages by utilizing sentence or phrase similarity measures. In a more realistic scenario, one can use a small set of parallel sentences, which is accessible for almost all languages nowadays, to train the similarity measures and then exploit them on comparable resources.
Bibliography


Kondrak, G. (2001). Identifying cognates by phonetic and semantic similarity. In *Proceedings of the second meeting of the North American Chapter of the Association for*


Saleh, I. and Habash, N. (2009). Automatic extraction of lemma-based bilingual dictionaries for morphologically rich languages. In Third Workshop on Computational Approaches to Arabic Script-based Languages at the MT Summit XII, Ottawa, Canada.


