THE UNIVERSITY
of EDINBURGH

Neural Semantic Role Labeling with More or Less Supervision

Rui Cai

Submitted for the degree of Doctor of Philosophy

THE UNIVERSITY OF EDINBURGH

2021

Abstract

In recent years, thanks to the relative maturity of neural network models, the task of automatically identifying and labeling the semantic roles has been the focus of renewed interest. These models have the capacity to learn continuous representations automatically and thereby forgo the need for extensive feature engineering. Semantic role labeling (SRL) has generally been recognized as a core task in natural language processing (NLP) and has been shown to benefit a range of NLP applications such as machine translation, information extraction and summarization.

Recent SRL systems have usually been trained on datasets whose semantic role annotations have been produced on the top of tree-banked corpora. This reflects the intimate relationship between syntactic information and semantic roles. In order to effectively incorporate syntactic information into neural network models, we train the semantic role labeler jointly with two auxiliary tasks: predicting the dependency label of a word, and determining whether there exists an arc linking it to the predicate. The auxiliary tasks provide syntactic information that is specific to SRL and can be learnt from training data (dependency annotations). This liberates our SRL system from the dependence on external parsers, which is believed to be noisy (e.g., on out-of-domain data or infrequent constructions).

Supervised neural SRL models, which derive their efficacy via sufficient annotated data, are driven by data. Nonetheless, the reliance on high-quality annotations obscures the development of SRL systems in low-resource scenarios (e.g., rare languages or domains). In order to reduce the annotation effort involved, we have rendered semi-supervised learning for SRL as simple as possible. More specifically, we propose an end-to-end SRL model and demonstrate it could effectively leverage unlabeled data within the cross-view training modeling paradigm. Our semantic role labeler is jointly trained with auxiliary tasks subsidiary to SRL. Consequently, our system may be applied directly to plain text, and it is essentially self-sufficient.

For true low-resource languages, we cannot even expect to perform semi-supervised learning for them, as SRL annotations are only available for a handful of the world's languages. To build a competitive semantic role labeler for these low-resource languages, we have resorted to cross-lingual semantic role labeling, which can transfer

supervision in a source language to target languages (low-resource languages). The backbone of our model is an LSTM-based semantic role labeler jointly trained with a semantic role compressor and multilingual word embeddings. The compressor collects useful information from the output of semantic role labeler and compress it into fixed-size cross-lingual representations. Our model (in contrast to earlier efforts, which deployed automatic alignments in order to transfer annotations) exists in a space of multilingual embeddings. For the target language, moreover, it affords direct supervision for the prediction of semantic roles. For model evaluation, we have also contributed two quality-controlled datasets, which we hope will be useful for the development of cross-lingual models.

Acknowledgements

First and foremost, I would like to express my gratitude to my supervisor, Mirella Lapata, for her support in my PhD study, professional development, and my personal life. Thanks to her knowledgeable and professional expertise, I learned a great deal during those weekly meetings. Besides all her help for my study, I am also grateful for her suggestions and continuous encouragement during my hard times. It will always be my honour and pleasure to be her student, and I cannot imagine having a better mentor and a PhD supervisor.

I am also grateful to all the excellent researchers in Edinburgh NLP group. Many thanks to Hao Zheng, Yumo Xu, Jiangming Liu , Ratish Puduppully, and Yang Liu, I learned a lot and was inspired during the weekly group discussions. Special thanks to my office mates Li Dong, Jiangming Liu, and Yanpeng Zhao, the discussion with them helped me to think from different angles.

Finally, none of my effort would be possible without the help of my family. Thanks to my parents for their love and support and thanks to my wife, Ling, for taking care of our baby.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Rui Cai*)

Contents

Chapter 1

Introduction

The study of the semantic information comprised within plain text is one of the key objectives of Natural Language Processing (NLP). Semantic role labeling (SRL), also known as shallow semantic parsing, aims automatically to identify and analyse the situations described by natural language sentences. Semantic role labeling, to be more precise, strives to locate arguments within a sentence for certain predicates; for each argument, moreover, it allocates a set of predefined relations (e.g.,"who" did "what" to "whom", "when", and "where"). For example, in the sentence *He no longer crowds the plate.*, the semantic information is conveyed by the predicate *crowds*, and the entities participating the event such as *He* and *the plate*. These entities are presented in terms that describe their involvement with the situation, i.e., in terms of semantic roles. The latter capture predicate-argument structures from sentences, which can then be used as semantic features by a wide spectrum of downstream tasks, ranging from machine translation (Aziz et al., 2011; Marcheggiani et al., 2018) to question answering (Zheng and Kordjamshidi, 2020) and summarization (Khan et al., 2015).

In recent years, a considerable amount of work (He et al., 2018; Lang and Lapata, 2014; Marcheggiani et al., 2017; Roth and Lapata, 2016) has been devoted to the task of automatically labeling semantic roles for a given input sentence. Due to the complexity generated by the variations in the syntactic realization of semantic roles, data-driven models have become the method of choice for this task and are typically trained in a supervised fashion. Among various data-driven models, neural networks

have been successfully applied to semantic role labeling, forging the need for extensive feature engineering. Syntactic features, including dependency labels and POS tags, are also fed as input into SRL models to enhance the performance of SRL systems. Such features are typically obtained via prepossessing tools that are also trained with labeled data. Within recent studies, the question of how best to incorporate these syntactic features has become a subject of much interest (Marcheggiani and Titov, 2017; Roth and Lapata, 2016) on SRL. Nonetheless, these works suffer from information loss (Roth and Lapata, 2016) and noise in syntactic features (Marcheggiani and Titov, 2017). In this thesis, we enable the SRL model to extract dependency information on their own, and (simultaneously) to learn syntax-aware hidden representations.

Unfortunately, the success of neural semantic role labeling is based on large amounts of semantically-annotated data, which are costly to obtain and mostly unavailable for rare domains and languages. As a result, it is common to see the performance of semantic role labelers (Cai et al., 2018; He et al., 2019; Marcheggiani and Titov, 2017) decrease dramatically when they are tested on out-of-domain test sets (i.e., when the training data and test data belong to different domains). This thesis also explores how to utilize unlabeled data, effectively to improve the performance of SRL models.

For low-resource languages without any labeled data, cross-lingual SRL offers a promising but also challenging alternative. If it is successful, we can obtain SRL systems for low-resource languages while only requiring labeled data regarding the source language. Current cross-lingual models rely on external tools such as word-alignment tools, and then perform annotation projection for target languages (Aminian et al., 2019; Fei et al., 2020). The present study demonstrates the feasibility of carrying out cross-lingual training *without* the need to depend on tools of external alignment.

1.1 Mono-lingual Semantic Role Labeling

The first basic question for semantic role labeling is: *How to obtain a semantic role labeler for a certain language?* For mono-lingual semantic role labelers, both the training data and the testing data belong to the same language, and they cannot usually be applied directly on other languages. Many previous mono-lingual SRL systems

trained their models in a supervised fashion, utilizing popular labeled datasets such as Propbank (Palmer et al., 2005). The performance of supervised SRL systems has been enhanced by the successful development of neural networks. Nonetheless, there are at least two challenges that hinder further performance gains.

The first problem is how to incorporate dependency information into SRL systems. Since SRL systems are trained on datasets whose semantic role annotations have been produced on top of treebanked corpora, semantic roles are closely tied to syntactic information. In addition, some predicates are typically associated with a standard linking, which is a deterministic mapping from syntactic roles to semantic ones (Lang and Lapata, 2010). Nonetheless, syntactic information usually exists in the form of trees (e.g., a dependency tree), and it cannot be directly fed to neural encoders such as LSTMs. A further obstacle is the availability of gold dependency information for some, but not all, text genres and domains. Dependency trees produced by external parsing tools are usually noisy; consequently, relying on these noisy dependency features could hurt the performances of SRL systems. In order to resolve this issue, the present study advances a new form of SRL model with the capacity to learn dependency information autonomously (see Chapter 4).

Another challenge is the scarcity of gold semantic role annotations. Manual annotation efforts for SRL resources like PropBank (Palmer et al., 2005) and FrameNet[1] amount to multi-million US-Dollar expenditures, and consequently they tend to be confined to certain specific domains. CoNLL2009 benchmark datasets, for instance, tend to be closely associated with texts regarding financial news, since they comprise sentences from the Wall Street Journal. Models trained only with labeled data do not generalize well on other data (or domains), especially when the size of the labeled samples is small (or when these samples all come from a single domain). In comparison with labeled data, the size of which is often restricted by manual-annotation costs, it is far easier to assemble large volumes of unlabeled data, i.e., data without annotations. In this thesis, we propose a semi-supervised framework for SRL, which leverages unlabeled data to improve the performance of SRL systems on different domains (see Chapter 5) .

[1] https://framenet.icsi.berkeley.edu/fndrupal/

1.2 Cross-lingual Semantic Role Labeling

For popular languages such as English and Chinese, there exists sufficient amounts of labeled data. Nonetheless, semantic role annotations are not available for low-resource languages, which of course prohibits the application of supervised learning to these low-resource languages. This raises the question: *How, when only annotated data from popular languages are used, may models be learnt for low-resource languages?*

Cross-lingual training offers a viable alternative to supervised methods. Previous work on cross-lingual SRL can generally be divided into two categories, namely model transfer (Ahmad et al., 2019a) and annotation projection (Aminian et al., 2019; Fei et al., 2020). The former strives to learn language-independent representations while taking multilingual features as the input (e.g., multilingual word embeddings and universal part-of-speech (POS) tags). Large-scale parallel corpora between source and target language, meanwhile, provide the basis for the performance of annotation projection. Source-side sentences in parallel corpora are automatically annotated by pre-trained source language SRL systems; then, these annotations are projected to target-side sentences based on word-alignment. An obvious drawback of annotation projection is that both automatic annotation and word alignment introduce noise, which may damage the performance of target-language SRL systems.

With the development of pre-trained language models, multilingual contextualized word embeddings such as multilingual BERT (mBERT; Devlin et al., 2018) have been widely used in cross-lingual (Ahmad et al., 2019a) and multi-lingual models (He et al., 2019). Recent work (Ahmad et al., 2019a; Fei et al., 2020) has demonstrated the surprising cross-lingual abilities of multilingual BERT, given that it is trained without any cross-lingual objective and with no aligned data. Consequently, multilingual BERT is now available for both low-resource and popular languages. Taking inspiration from cross-lingual representations of words (mBERT), we propose to learn cross-lingual representations of semantic roles (see Chapter 6). Instead of relying on word-alignment tools, we perform cross-lingual training based on cross-lingual representations learned with a semantic information compressor. Current human-labeled SRL datasets for different languages lack of a homogeneous labeling style, which increases the difficulty of the evaluation of cross-lingual SRL systems. With this in mind, the present the-

sis presents Chinese and German datasets (with annotation performed by experts) that evince a unified scheme of annotation.

1.3 Thesis Statement

In this thesis, we investigate a series of hypotheses relating to the labeling of semantic roles in natural language sentences, which we test through extensive evaluation and analysis of our methods.

> HYPOTHESIS I: There are tight connections between dependency information and semantic roles; therefore, enabling SRL models to extract dependency information can improve their performance.

Previous research (Marcheggiani and Titov, 2017; Roth and Lapata, 2016) has hinted at the benefits of incorporating dependency features for SRL. Here, we propose a syntax-aware SRL model, which can learn syntax-aware representations autonomously, without the reliance on external parsers.

> HYPOTHESIS II: Within the paradigm of *cross-view training* modeling, unlabeled data can be effectively leveraged to enhance SRL performance.

Cross-view training (CVT; Clark et al., 2018) has been successfully applied to several NLP tasks, unfortunately, application of CVT to semantic role labeling is fraught with difficulty. To solve this problem, we adapt the cross-view training for the semantic role labeling task and build an end-to-end SRL system that can directly be applied to plain text.

> HYPOTHESIS III: It is possible to obtain semantic role labelers for target (low-resource) languages without target-language annotations and word-alignment tools.

While word-alignment tools have been widely utilized for cross-lingual semantic role labeling (Aminian et al., 2019; Fei et al., 2020), our work is the first to learn cross-

lingual role representations that can provide direct supervision for the prediction of semantic roles in the target language, thereby avoiding noise introduced by external tools.

1.4 Contributions

The main contributions of this thesis are as follows:

- We propose a multi-task model that learns dependency-aware representations for semantic role labeling without using any external parser. Previous work has concentrated on utilizing dependency features produced by external parsers which can be noisy. We design a dependency information extractor to learn two types of predicate-specific syntactic features: dependency labels and link types. Enabling the model autonomously to extract dependency information removes the reliance on external parsers, and — by exploiting concealed representations within the extractor — enhances the performance of the model.

- In order to apply cross-view training (CVT; Clark et al., 2018) to the semantic role labeling task, which relies on various syntactic features, we develop a sentence learner that is able to perform all tasks subsidiary to semantic role labeling (e.g., predicate identification, POS tagging, dependency parsing). With the help of the sentence learner, we build an end-to-end SRL system, which can directly be applied to plain text without using preprocessing tools.

- We adapt the CVT proposed by Clark et al. (2018) for semantic role labeling task. The sentence is segmented, and words prior to and following the predicate are addressed differently, which ensures that information derived from the predicates percolates to each word within the sentence. Furthermore, we show that the strategy of selecting the target predicate (in sentences containing multiple candidates) influences performance; we also advance a strategy whereby predicates are randomly selected from potential candidates.

- We propose a cross-lingual SRL model that can effectively leverage unlabeled

parallel data without relying on alignment tools. Previous work concentrated on aligning words in parallel sentences, and then filtered low-confidence alignments empirically to reduce the impact of erroneous alignments. We design a semantic role compressor that collects useful information from the output of the semantic role labeler, thus producing cross-lingual representations of semantic roles. With the help of contextualized multilingual word embeddings (mBERT; Devlin et al., 2018), these cross-lingual role representations can provide direct supervision for the prediction of semantic roles in the target language. Intermediaries, such as machine translation and word-level alignments, are thereby avoided.

- For a better evaluation of cross-lingual models, we have contributed two manually annotated datasets, following the PropBank-style guidelines (Palmer et al., 2005). These two datasets contains 258 German and 304 Chinese sentences, which are smaller than UPB but evince a higher accuracy. For Chinese and German, these two datasets are (so the author believes) the first published, human-labeled SRL resources to conform to the PropBank annotative style.

1.5 Thesis Outline

Chapter 2 provides as an overview of two important linguistic concepts, namely *frames* and *semantic roles*, and the linguistic theories that lay the foundation for the empirical resources described in Section 2.2. We first discuss the historical development of frames and semantic roles, respectively, and then introduce some popular SRL datasets, which have been widely used in the research of semantic role labeling. We then present some basic steps of semantic role labeling: predicate identification, predicate disambiguation, argument identification, argument classification, and global inference. This chapter aims primarily to familiarize the reader with some of the basic concepts that underpin semantic role labeling — concepts that are deployed as building blocks within a variety of learning environments.

Chapter 3 describes methodological issues accompanying the task. It also delineates a range of learning environments within which our models are implemented. We start by giving definitions of supervised and semi-supervised semantic role labeling, both

of which are monolingual tasks with access to labeled data. The difference between them is that semi-supervised semantic role labeling takes advantage of large amount of unlabeled data to improve the generalization ability of models. Then, the environment of unsupervised semantic role labeling, with no access to SRL annotations, is briefly outlined. Finally, we formalize the learning setting of cross-lingual SRL, which is quite different from mono-lingual SRL and more appropriate to the low-resource languages.

Chapter 4 presents our syntax-aware supervised neural model for semantic role labeling. First, the close connection between dependency labels and semantic roles is demonstrated. Because of this relationship, dependency information can helpfully be deployed in improving the performance of SRL systems. After discussing the drawbacks of previous work, we present our neural module *dependency information extractor*, which can learn dependency features autonomously and provides syntax-aware representations for the semantic role labeler. We evaluate our model in the context of the CoNLL2009 benchmark datasets, revealing significant improvements over previous approaches.

Chapter 5 presents our end-to-end semi-supervised neural model for semantic role labeling. We first introduce cross-view training, a recently proposed semi-supervised learning algorithm that improves the hidden representations learning of a Bi-LSTM sentence encoder using a mix of labeled and unlabeled data. Subsequently, we apply CVT to the task of semantic role labeling, which entails reliance on a range of syntactic features. To this end, we develop a *sentence learner* which is able to perform all tasks subsidiary to semantic role labeling (i.e., predicate identification, POS tagging, dependency parsing). We evaluate our model in the context of different domains and multiple languages using the CoNLL-2009 benchmark dataset, which shows that our approach outperforms baseline models in English, Chinese, Czech, and Spanish.

Chapter 6 addresses cross-lingual semantic role labeling, and here, while exclusively employing unlabeled parallel data and source-language labeled data, SRL systems for low-resource languages are developed. We first introduce the semantic role compressor and decompressor, which are trained to learn cross-lingual semantic role representations. The method of performing cross-lingual training is then demonstrated. Unlike previous work which relied on word-alignment tools, the present study leverages cross-lingual semantic role representations to improve the performance of the target-

language semantic role labeler. For evaluation, we make use of several multi-lingual SRL benchmarks, and experimental results show that our method is highly effective across languages and annotation schemes, even compared with systems that make use of supervised features.

Chapter 7 presents a summary of the principal findings of the thesis, and, while acknowledging certain limitations of the study, also indicates potential avenues for future research.

1.6 Published WorK

Some of the material presented in this thesis has been previously published. The work presented in Chapter 4 is a refinement of the work published in Cai and Lapata (2019b).The work in Chapter 5 was published in Cai and Lapata (2019a), and the work in Chapter 6 was published in Cai and Lapata (2020).

Chapter 2

Background

How is language used to convey knowledge? There are different answers towards this question, and these answers have given birth to classic linguistic theories (Fillmore, 1968; Minsky, 1974) regarding frames. Before we move on to describe the learning settings, methodology, and experiments of our work, this chapter provides an overview of theories of frames and semantic roles. These theories have provided the foundations for recently constructed empirical resources (e.g., FrameNet[1]) built lately, which have in turn been widely used in current research (Carreras and Màrquez, 2005; Hajič et al., 2009a).

The material in this chapter is presented in four parts. First, we introduce the basic concepts and terminology of frames and semantic roles. The second part introduces three large-scale resources, namely: FrameNet, PropBank, and Nombank. In the third part, we introduce two different annotation styles (span- and dependency-based), as well as the main steps for building semantic role labelers. Finally, we introduce the evaluation measure of semantic role labeling systems.

[1] https://framenet.icsi.berkeley.edu/fndrupal/

2.1 Frames and Semantic Roles

2.1.1 Frames

Minsky (1974) introduced frames as, "a data-structure for representing a stereotyped situation, like being in a certain kind of living room, or going to a child's birthday party". According to the theory advanced by Minsky (1974), the frame is essentially a structure selected from memory, generated people encounter a new situation or substantially change their view of a present problem. Several different kinds of information are attached to frames, including how to use the frame, what to expect next and what to do if these expectations are not met. If we think of a frame as a network of nodes and relations, the "top levels" of a frame are fixed (some information in the frame is generally unchanged). Other information, stored in lower-level "terminals", usually changes with the new situation encountered.

Before Minsky (1974), Fillmore (1968) proposed that each verb selects a number of *deep cases* (i.e., semantic roles, such as Agent, Patient, Location, and Instrument) to form its *case frame*. Thus, case frames can be regarded as structures holding together the deep cases (arguments) bound to a particular verb (predicate). For example, in the sentence, *Matilde fried the catfish in a skillet*, *fried* is a verbal predicate, and the three nominal elements *Matilde*, *catfish* and *a skillet* are bound together by the verbal predicate *fried*. The bounded structure in the example sentence forms a case frame, and each element evinces a specific prototypical relationship with the verbal predicate. For example, the case frame for *fried* specifies the entity that instigates the action as an Agent (here *Matilde*). Such prototypical relationships are referred to as semantic roles and will be discussed in Section 2.1.2. Some semantic roles are obligatory, while others are optional. Obligatory roles should not be deleted, otherwise one would produce ungrammatical sentences. For example, if the Patient (*the catfish*) of *fried* is deleted, the sentence becomes ungrammatical, as the Patient is obligatory. Nonetheless, if we delete the optional role Instrument (*a skillet*), the sentence becomes *Matilde fried the catfish*, which is still grammatical.

2.1.2 Semantic Roles

Being quite similar to those frames that describe events and the participants in them, *semantic roles* characterize how an entity is involved in an event or action. *Agent* is a common role that describes the instigator of an action, which in turn might affect the target entity (*Patient*). Other common roles include *Location*, i.e., the place where an event or action takes place, *Instrument*, i.e., the entity utilized during an action, and so on. The aforementioned roles belong to a set of general roles, as they are globally defined for any action or event. In contrast, situation-specific roles are defined individually, and detailed examples will be given in the next section.

Fillmore (1968) investigates the phenomenon of *deep cases*, which are also known as *semantic roles* and consist of *Agent*, *Patient*, *Result*, *Instrument*, *Location*, and *Neutral*. As distinct from other roles, the explanation of *Neutral* is determined by the current verbal predicate. These semantic roles, except *Neutral*, are apparently sufficient in general to characterize the arguments of any predicates. In other words, they are globally defined. This attribute of general roles is important from a linguistic standpoint because it leads to a concise linguistic theory. Nonetheless, this also causes disagreement about which roles should be included in such a general role set (Dowty, 1991). Other similar issues include the questions of which roles are necessary and sufficient, how general or specific these should be, and what roles are present in all languages.

In contrast to general roles, situation-specific roles does not have a general meaning and their definition depends on the situation they describe. For example, the situation-specific roles *Buyer* and *Seller* describe two important participants during a trading event, and they are unlikely to appear in other kinds of situation, such as cooking or explosions. Such situation-specific roles have a more precise meaning, and therefore support a more detailed representation of the situation, but at the cost of increased complexity and labeling workload.

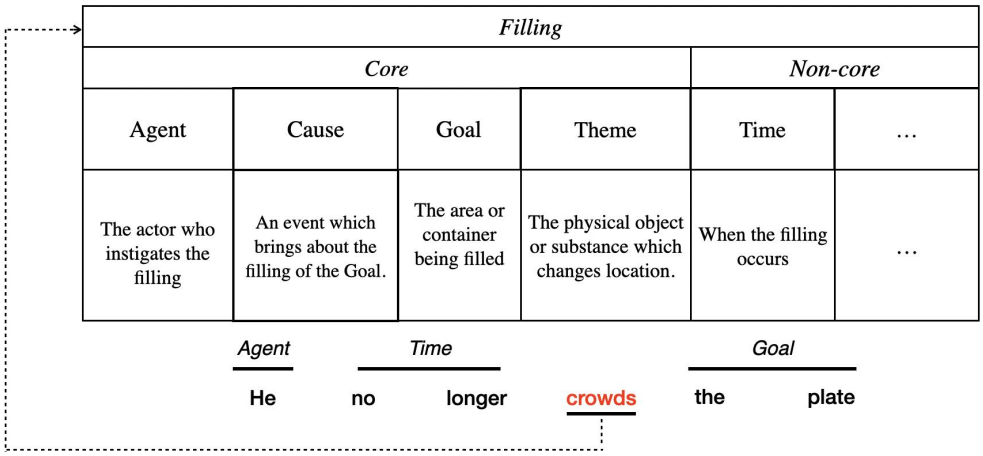

Figure 2.1: *Frame elements* in the sentence *He no longer crowds the plate*, where *crowds* triggers the frame *filling* and is designated a lexical unit.

2.2 Empirical Resources

For data-driven SRL models, empirical resources are vital for the development of semantic analysis systems. In addition to serving as the basis for training supervised models, they can also promote empirical linguistic research regarding frame semantics. This section introduces FrameNet (Baker et al., 1998), PropBank (Palmer et al., 2005) and NomBank (Meyers et al., 2004): three large-scale English role-semantic resources. In each resource sentences, are semantically analyzed, and are accompanied by lexical dictionaries such as the pre-defined frame lexicon in PropBank.

2.2.1 FrameNet

FrameNet is a lexical database of English built around a linguistic theory called *frame semantics* (Fillmore, 1968). The meaning of each word in this database is represented on the basis of a *semantic frame*, which is a description of a type of event, relation, or entity and the participants within it.

For example, as shown in Figure 2.1, the concept of "filling" might involve the person who instigates the filling (*Agent*), the container or area being filled (*Goal*), the event that brings about the filling (*Cause*), the physical object or substance that changes location (*Theme*), and when the filling occurs (*Time*). In the FrameNet project, this is

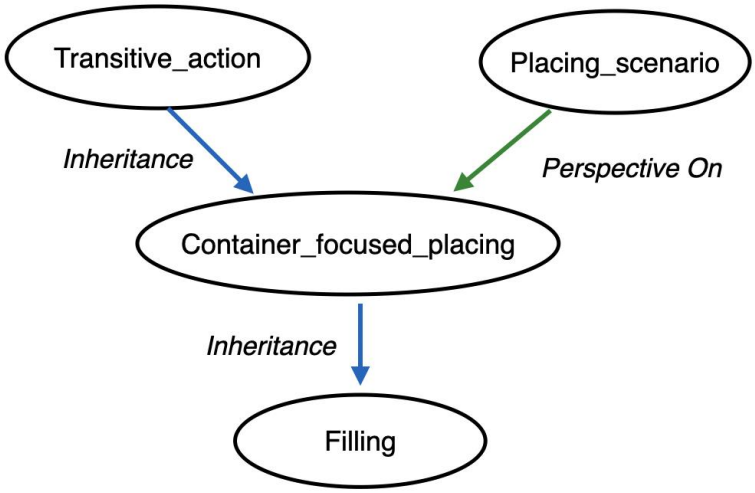

Figure 2.2: Relations between the frame *Filling* and various other frames.

represented as a frame designated *filling*, while *Agent, Goal, Cause, Theme*, and *Time* are referred to *frame elements* (FE). Some *frame elements*, such as *Buyer* and *Seller*, are regarded as *core*, which means that they uniquely define a particular frame. *Non-core frame elements* are peripheral, as they provide information for general aspects of events, such as *Time* and *Place*. The words that evoke the frame (such as crowd and cover) are the *lexical units* (LU) of the frame. For instance, in the example in Figure 2.1, *crowds* is the lexical unit of the frame *filling*.

FrameNet also defines relations between frames such as inheritance (one situation is a special case of another), perspective (two frames describe the same situation from different perspectives), composition (one situation contains the other) and temporal precedence (one situation happens before the other). Figure 2.2 illustrates the relations obtaining between the frame *Filling*, the *Container_focused_placing*, the *Transitive_action* and the *Placing_scenario*.

Formally, FrameNet[2] annotations are sets of triples that represent the FE realizations for each annotated sentence, with each including a frame element name (for example, Food), a grammatical function (say, Object) and a phrase type (say, noun phrase (NP)). Figure 2.3 gives an example from FrameNet, where *Bake* is lexical unit of the frame *Apply_heat*. As of today, FrameNet has provided more than 200,000 manually annotated sentences linked to more than 1,200 semantic frames, and has been used

[2] https://framenet.icsi.berkeley.edu/fndrupal/

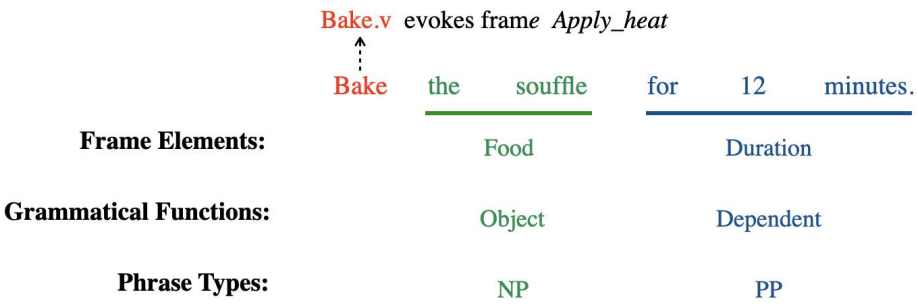

Figure 2.3: An example sentence in FrameNet, with Frame elements, grammatical functions, and phrase types.

extensively to provide training data for semantic role labeling (FitzGerald et al., 2015; Giuglea and Moschitti, 2006; Thompson et al., 2003). In addition to the annotated sentences, nearly 1,800 frame-to-frame relations constitute a hierarchy of semantic frames, and FrameNet has developed a visualization tool, *FrameGrapher*[3], for viewing the relations between frames and their frame elements.

2.2.2 PropBank

Unlike FrameNet, PropBank (Palmer et al., 2005) is a verb-oriented database that associates each verbal predicate with one frame, which in turn captures the possible configurations of semantic roles. PropBank consists of two portions, namely a lexicon of frames and an annotated corpus. The annotated corpus provides predicate-argument annotations for the entire Penn Treebank (Marcus et al., 1993a), and each verb in the latter is linked to a specific frame in the pre-defined frame lexicon. The semantic roles annotated in PropBank are divided into core roles and adjunct roles. Adjunct roles, as their name indicates, are realized as adjuncts. They can be part of the frame of any predicate, and are therefore globally defined for all predicates. By contrast, core roles are defined individually for each predicate

A0 and A1 are the two most common core roles in PropBank. A0 is assigned to arguments that are understood as agents, causers or experiencers, while A1 is usually assigned to patient arguments, i.e., the argument that undergoes a change of state or is in the state of being affected by an action. Unlike A0 and A1, other core ar-

[3] https://framenet.icsi.berkeley.edu/fndrupal/FrameGrapher

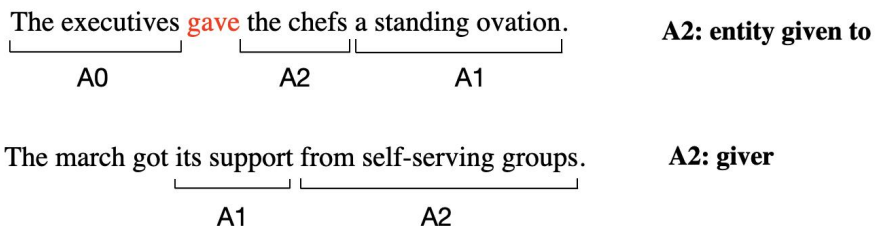

Figure 2.4: Two sentences from PropBank annotated with span-based labels (definitions of A2 are given on the right).

guments can be specific to verbs. Figure 2.4 shows two examples in PropBank, and the definition of A2 in each sample differs with the current predicate (*gave* and *got*) within the sample. AA is a secondary agent tag that will only be used when the argument structure outlined in the roleset indicates a proto-agent role. For instance, in the sentence *John walked his dog.*, *John* is the secondary agent that causes the dog to walk but is not (necessarily) walking himself. Adjunct arguments are universal to all verbs, usually labelled with one of the following roles: location (LOC), extent (EXT), cause (CAU), time (TMP), purpose (PRC), manner (MAN), direction (DIR), expressions (REC), predicatives (PRD), discourse connectives (DIS), negation (NEG), modal verbs (MOD), and a general purpose Adverbial (ADV).

Throughout this thesis, we deploy PropBank v3.0 for experiments. The latter was built as an additional annotation layer on the top of the Wall Street Journal portion of the Penn Treebank (Marcus et al., 1993a), and it contains around 110,000 annotated frame instantiations. The latest release of Propbank[4] contains additional annotations for Ontonotes[5] and the English Web Treebank. An additional 160,000 predicates have been annotated with Prop-bank style roles in the BOLT corpora[6], and these will be made publicly available when the LDC releases BOLT to the general catalogue.

2.2.3 NomBank

NomBank is an annotation project at New York University and is related to PropBank. Its goal is to mark the sets of arguments that cooccur with nouns in the PropBank Corpus, as PropBank records such information only for verbs. Nombank shares the

[4] https://github.com/propbank/propbank-release
[5] https://catalog.ldc.upenn.edu/LDC2013T19
[6] https://catalog.ldc.upenn.edu/LDC2013T19

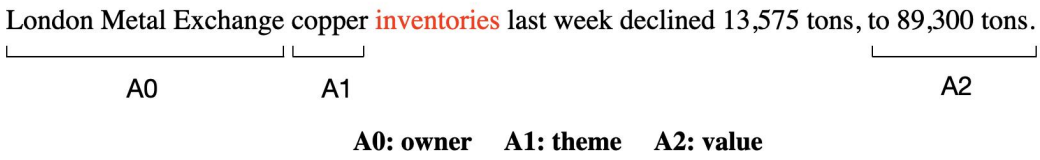

Figure 2.5: A sample in NomBank annotated with span-based labels, and definitions of semantic roles are given in the bottom.

same semantic roles (e.g., A0 and A1), and it also contains a PropBank-style lexicon (frame files). Figure 2.5 provides an example from NomBank, the predicate of which is the noun *inventories*. These resources allow one to label the various arguments and adjuncts of head nouns with semantic roles. Researchers are now making a coordinated effort to ensure that, when possible, role definitions are consistent across parts of speech. For example, PropBank's frame file for the verb *decide* is used during the annotation of the noun *decision* in NomBank.

NomBank is constructed on the basis of Catherine Macleod's Nomlex[7] project and related work on support verbs, since half of the argument-taking nouns in the corpus are nominalizations or nouns that have nominalization-like properties. For example, *aggression* and *agenda* have argument structures similar to the verbs *destroy* and *schedule*. In December 2007, NomBank.1.0[8] was released, it covered all the argument-taking nouns in the Penn Treebank Wall Street Journal corpus. NomBank.1.0 includes a total of 114,576 propositions derived from the checking of 202,965 noun instances and the selection only of those nouns whose arguments occurred in the same sentence. (Hence, approximately 35,000 nouns were excluded because these nouns have no corresponding arguments.)

[7] https://nlp.cs.nyu.edu/nomlex/index.html
[8] https://nlp.cs.nyu.edu/meyers/NomBank.html

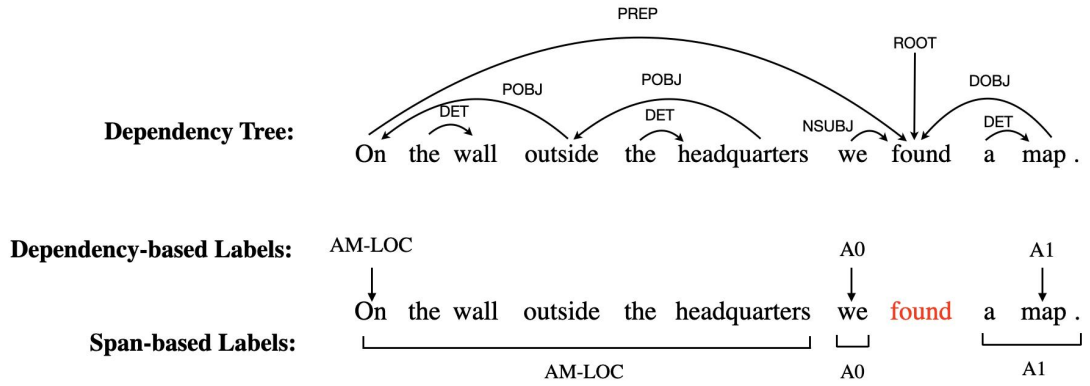

Figure 2.6: A sample sentence from PropBank: *On the wall outside the headquarters we found a map* (where *found* is the current predicate). The upper of the Figure shows the dependency tree for this sentence, while the lower part presents semantic roles following dependency and span-based SRL conventions.

2.3 Modeling Semantic Roles

2.3.1 Span vs. Dependency

Currently, there are two popular formalisms to represent the semantic predicate-argument structure: dependency-based or span-based. The dependency-style formalism annotates the syntactic heads of arguments, while span-based SRL annotates the entire argument span. In span-style annotations, the representation of the predicate-argument structure is quadruple in form, including: the current predicate, the start and the end of the argument span, and the corresponding relation between the argument and the current predicate. For dependency-based annotations, only the dependency head, rather than the start and end of each argument span, is labeled. As a result, the annotation for each argument is a "triple", including: the predicate, the syntactic head of the argument span, and their relationships.

As illustrated in Figure 2.6, span-style annotation provides information for the entire span, while dependency-based SRL merely concentrates on the dependency head of each argument. For example, in Figure 2.6, the span *On the wall outside the headquarters* provides information regarding the location of an event; thus, it is annotated with AM-LOC. For a dependency-based SRL, *On* is the dependency root of the subtree derived by syntactically parsing *On the wall outside the headquarters*. As a result, only *On* is labeled with AM-LOC.

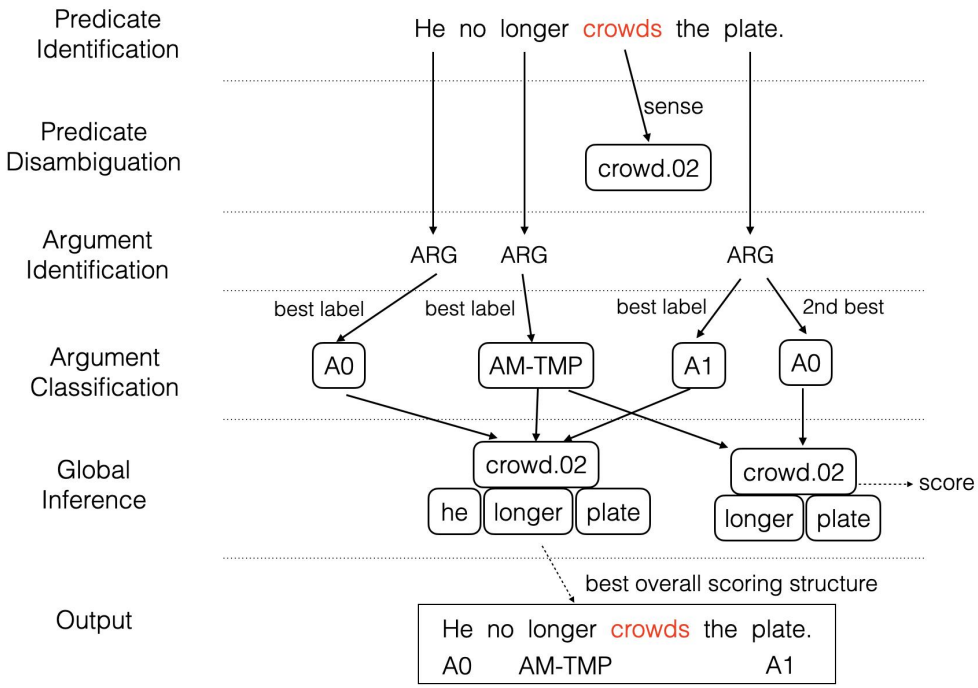

Figure 2.7: Steps in an SRL system (dependency-based).

Both span- and dependency-based annotations can represent semantics effectively, and one annotation type can be transformed into the other style on the basis of a gold syntactic structure. The discussion regarding the potential superiority of one form or the other has persisted for several years. In Johansson and Nugues (2008) and Li et al. (2019b), for example this topic has been discussed, and both papers reach the same conclusion: the best dependency-based SRL model available at the time outperformed the best span-based model, and significantly so.. Recently, Li et al. (2019a) proposed a new cross-style semantic role labeling convention, which annotates the entire span and dependency head simultaneously. In this thesis, we mainly concentrate on dependency-based SRL, although our methods are also compatible with span-style SRL.

2.3.2 SRL, Step by Step

The input for an SRL system is typically raw text. One must usually take a few steps to process it and label the semantic roles (Björkelund et al., 2009; Toutanova et al., 2008). The text is analysed sentence by sentence. Figure 2.7 provides an overview of the main steps in a dependency-based semantic role labeling system. First, sentences are syntactically analysed; subsequently, predicate identification and disambiguation

Sense ID	find.01	find.02
Definition	*discovery*	*adjudge*
Core Role Set	**Arg0-PAG**: *finder* **Arg1-PPT**: *thing found* **Arg2-GOL**: *benefactive, found for*	**Arg0-PAG**: *finder* **Arg1-PPT**: *thing found*
Example	Carter and Herbert's finding of the tomb of Tutankhamun was significant.	State courts found the rates illegal.

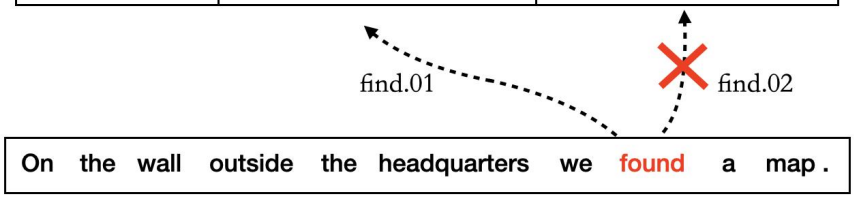

Figure 2.8: Predicate disambiguation for the predicate *found* in a sample sentence. The table in the upper part of the Figure shows the content of frame files defined for the verb *find*. Its first sense (i.e., *find.01*) agrees with the context of *found* in the sample sentence.

take place. The job of a predicate identifier is to identify all predicates in a sentence. These predicates are typically expressed as verbs. Nonetheless, some other syntactic categories (e.g., nouns and adjectives) can also play the role of a predicate. Each predicate in a sentence might have multiple senses. For example, in PropBank, usages of each predicate are stored in the frame set[9], and predicate disambiguation must take place to determine to which frame the current predicate belongs. A detailed example is given in Figure 2.8.

Given a predicate and its sense, the next step is filtering (or pruning) candidate arguments. Since a continuous or discontinuous sequence of words might be an argument, extensive exploration of such a candidate space is not feasible (besides being unnecessary), because it is large and unbalanced (i.e., most sequences are not actual arguments of the predicate). Xue and Palmer (2004) produced a simple algorithm to prune arguments, relying on syntactic parse trees of input sentences. The pruning algorithm in Xue and Palmer (2004) is described as follows:

[9] http://verbs.colorado.edu/propbank/framesets-english-aliases/

1. Designate the predicate as the current node and collect its sisters (constituents attached at the same level as the predicate) unless its sisters are coordinated with the predicate. If a sister is a PP, also collect its immediate children.
2. Reset the current node to its parent and repeat Step 1 till it reaches the top level node.

The third step consists of scoring candidate arguments *locally* by learning a function that outputs scores for all possible role labels. Note that there is an extra "no-argument" label to indicate candidates that are not considered to be arguments. In some SRL systems, scoring is divided into two sub-processes, namely argument identification and argument classification. During argument identification, the system only scores between "argument" and "non-argument" labels. Only candidates labeled as "argument" participate in argument classification, which then assigns each argument a particular semantic role.

In local scoring, decisions for each candidate are independent of one another. In the final step, the predictions of local scorers are combined to obtain a global structure of labeled arguments for the current predicate. This process is referred to as global scoring or joint scoring, and it is often implemented by ensuring that a labeling result satisfies a set of structural and SRL-dependent constraints (He et al., 2017; Punyakanok et al., 2008). We list some example constraints as follows:

1. BIO Constraints for span-based SRL: these constraints reject any sequence that does not produce valid BIO transitions, such as B_{ARG0} followed by I_{ARG1}.
2. Punyakanok et al. (2008) describe a list of SRL specific global constraints:
   - Unique core roles: each core role (ARG0-ARG5, ARGA) should appear once, at most, for each predicate.
   - Continuation roles: a continuation role C-X can exist only when its base role, X, is realized before it.
   - Reference roles: a reference role R-X can exist only when its base role X is realized (not necessarily before R-X).

Currently, there are many variations (Cai and Lapata, 2019b; Daza and Frank, 2019; Marcheggiani et al., 2017) of the four-step SRL system. Some systems, especially those built for dependency-based SRL (Cai and Lapata, 2019b; Marcheggiani

et al., 2017), might omint either one or a few steps, e.g., they may adopt local scoring only, or they may skip candidate argument pruning. In dependency-based SRL, candidate arguments consist of single words, instead of word sequences. Hence the exploration of the candidate space becomes feasible and argument pruning can be bypassed (Marcheggiani et al., 2017). A similar situation occurs during scoring, as dependency-based SRL systems do not need to label out the entire span. This in turn greatly reduces the constraints on the labeling result and the necessity of global scoring (Marcheggiani et al., 2017).

2.4 Evaluation

A standard automatic SRL model generally works as follows: given a sentence and a target predicate, it finds the arguments of the predicate and marks them with a semantic role. Evaluation for SRL systems tends to be performed in terms of precision, recall, and F1 of the labeled arguments. For span-based SRL, during evaluation, an argument is only considered to be correctly labeled when its boundaries and semantic role labels meet the gold standard. For dependency-based SRL, the role labels of each word are considered independently; this is similar to common classification tasks.

Formally, let G denote the set of gold arguments, C the set of arguments predicted by the SRL system to be evaluated. Precision can be defined as:

$$Precision = \frac{|G \cap C|}{|C|} \tag{2.1}$$

and recall is calculated as follows:

$$Recall = \frac{|G \cap C|}{|G|} \tag{2.2}$$

The F_1 score is defined as:

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{2.3}$$

Equations (2.1) and (2.2) apply only to argument labeling. Nonetheless, some

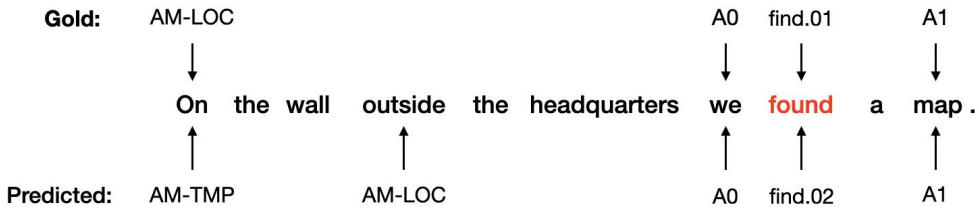

Figure 2.9: Dependency-based labeling results of a sample sentence.

evaluators also take predicate disambiguation into consideration. The result (word sense ids) of predicate disambiguation is treated in the same fashion as role labels, and participates in the calculation of precision and recall. Therefore, precision and recall are redefined as:

$$Precision = \frac{|G \cap C| + N_{correct}}{|C| + N_{predicate}} \tag{2.4}$$

$$Recall = \frac{|G \cap C| + N_{correct}}{|C| + N_{predicate}} \tag{2.5}$$

where $N_{correct}$ is the number of predicates, the sense of which is correctly assigned by the end-to-end SRL system, and $N_{predicate}$ is the total number of predicates in the test set.

Given the example provided in Figure 2.9, when one performs evaluation without predicate disambiguation, precision is $\frac{2}{4}$ because two out of four semantic roles are incorrect. *On* is incorrectly labeled as AM-TMP and *outside* is incorrectly labeled with AM-LOC. Recall is $\frac{2}{3}$, since there are only 3 arguments in gold. If predicate disambiguation is taken into consideration, precision becomes $\frac{2}{5}$ and recall becomes $\frac{2}{4}$. In practice, predicate disambiguation is easier than argument labeling. Hence, evaluation results tend to be improved when disambiguation results are taken into account.

2.5 Summary

Frames convey knowledge of events by specifying participants and their semantic roles. Semantic role labeling tries automatically to extract frames from input sentences and to label participating entities with semantic roles. Large-scale empirical resources such as FrameNet and PropBank provide training data for data-driven SRL systems.

These resources, follow different linguistic theories, and as a result, employ different predefined role sets and labeling schema.

Apart from the labeling of inventories, semantic role annotations also differ in their style (i.e., they may be dependency-based or span-based). Both span- and dependency-based annotations can represent semantics effectively, and one annotation type can easily be transformed into another style easily. Despite differences in labeling and annotation style, SRL systems typically a adopt a set of common steps to process input sentences and label the semantic roles. With the development of neural SRL systems, certain few steps (e.g., argument pruning and global inference) have become less critical and are often skipped (Daza and Frank, 2019; Marcheggiani et al., 2017). For evaluation, we use precision, recall and the F_1 score to measure the performance of SRL systems. The final result depends on whether predicate disambiguation is taken into consideration or not.

Chapter 3

Semantic Role Learning Settings

Before presenting our models for semantic role labeling, it is important to describe the methodological issues accompanying the task. Below, we will establish various learning settings in which our models are applied.

We start by giving a definition of *supervised* semantic role labeling, which is monolingual and has access to labeled data. Supervised semantic role labeling, as introduced in Section 3.1, relies on high-quality annotations that are costly to obtain, and mostly unavailable in low-resource scenarios (e.g., rare languages or domains). *Semi-supervised* semantic role labeling aims to reduce the annotation effort involved via semi-supervised learning, which is introduced in Section 3.2. When labeled data are not available, *unsupervised* semantic role labeling (see Section 3.3) performs semantic analysis without annotations for predicates, arguments, or argument roles.

The above settings are all monolingual, and in Section 3.4, we introduce the setting of *cross-lingual* semantic role labeling. This aims at leveraging existing annotations in a source language to minimize the effort required to construct a model or labels for a new target language.

3.1 Supervised Semantic Role Labeling

Semantic role labeling, i.e., the detection of semantic arguments in natural language text for a specific predicate, is most commonly modeled as a fully supervised sequence labeling task (He et al., 2018; Marcheggiani et al., 2017). As such, it requires a dataset C consisting of labeled training instances $\{(s_i, g_i)\}_{i=1}^{|C|}$, where s_i is a sentence in raw text and g_i are the gold labels for sentence s_i. Assuming that w_j is the j-th word in s_i, its corresponding gold label l_j also appears at the j-th position in g_i. Therefore, s_i and g_i usually share the same sequence length n, and s_i can be viewed as a sequence of words $(w_1, w_2, ..., w_n)$. Meanwhile, g_i is a sequence of labels $(l_1, l_2, ..., l_n)$, each of which can be found in a pre-defined set r of semantic roles.

As discussed in Section 2.2, empirical resources lack a unified annotation standard, and as a result, training datasets built on the basis of these semantic resources tend to follow different annotation conventions. As a collection of possible semantic roles, set R also varies with different datasets. Figure 3.1 shows a toy example of a semantic role labeling training set (top left), which consists of sentences paired with gold semantic labels. The sentences in Figure 3.1 were selected from the CoNLL-2009 shared task English SRL dataset. The goal is to train a labeler (bottom) and use it to predict the semantic roles of unseen test instances (top right). Predicate disambiguation is not shown in Figure 3.1, and a classifier for word senses can be learnt in a similar manner to the semantic role labeler.

A semantic role labeler, parameterized by θ, will produce a score over all labels in a role set. Formally, for i-th word w_i in input sentence s, given current predicate w_p, the score for j-th semantic role r_j produced by the trained SRL system is $s(r_j|w_i, w_p, s)$. The probability distribution over all roles can be obtained by normalizing the collection of role scores:

$$P_\theta(r|w_i, w_p, s) = \text{softmax}\{s(r_1|w_i, w_p, s), \\ ..., s(r_{n_r}|w_i, w_p, s)\} \tag{3.1}$$

where n_r is the size of the role set. SRL systems, without a global inference step, directly output role labels with the highest scores or probabilities as the final result. Otherwise, scores or probabilities are fed to another model, which tries to find a good global role assignment for the identified arguments.

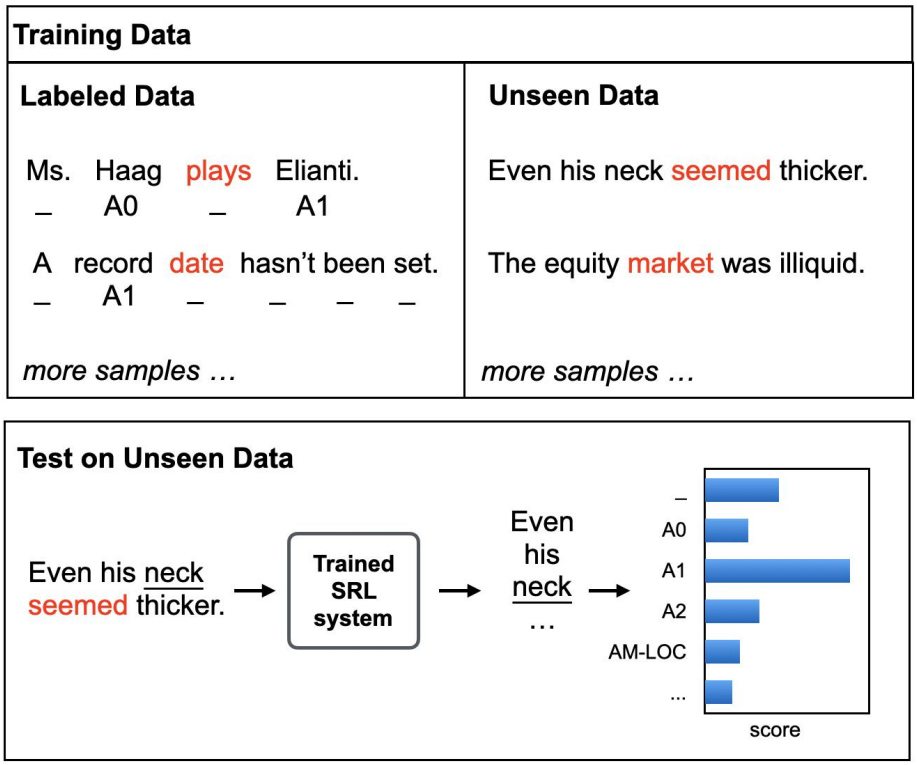

Figure 3.1: A toy example of supervised semantic role labeling. Words in red are predicates and _ indicates non-arguments. A training set (top left) is used to train a semantic role labeler. The trained model (bottom) outputs scores of all roles for each word in unseen sentences.

3.2 Semi-supervised SRL

Compared with labeled data, the size of which tends to be limited by the cost of manual annotation, it is much easier to collect a large quantity of data without annotations (unlabeled data). Models trained with only labeled data often lack generalization ability for other data (or domains), especially when the size of labeled samples is small (or these samples all come from a single domain). Semi-supervised learning (Clark et al., 2018; McClosky et al., 2006) tries to leverage both labeled and unlabeled data to improve the performance of models vis-à-vis test data . For semi-supervised learning to work, certain assumptions will have to hold. One common assumption is the *cluster* assumption: if data points are in the same cluster, they are likely to be of the same class. Another common assumption is the *manifold* assumption: the (high-dimensional) data lie (roughly) on a low-dimensional manifold. In fact, no matter the clustering assumption or the manifold assumption, its essence is the basic assumption that "similar samples have similar outputs". Without such assumptions, it would never

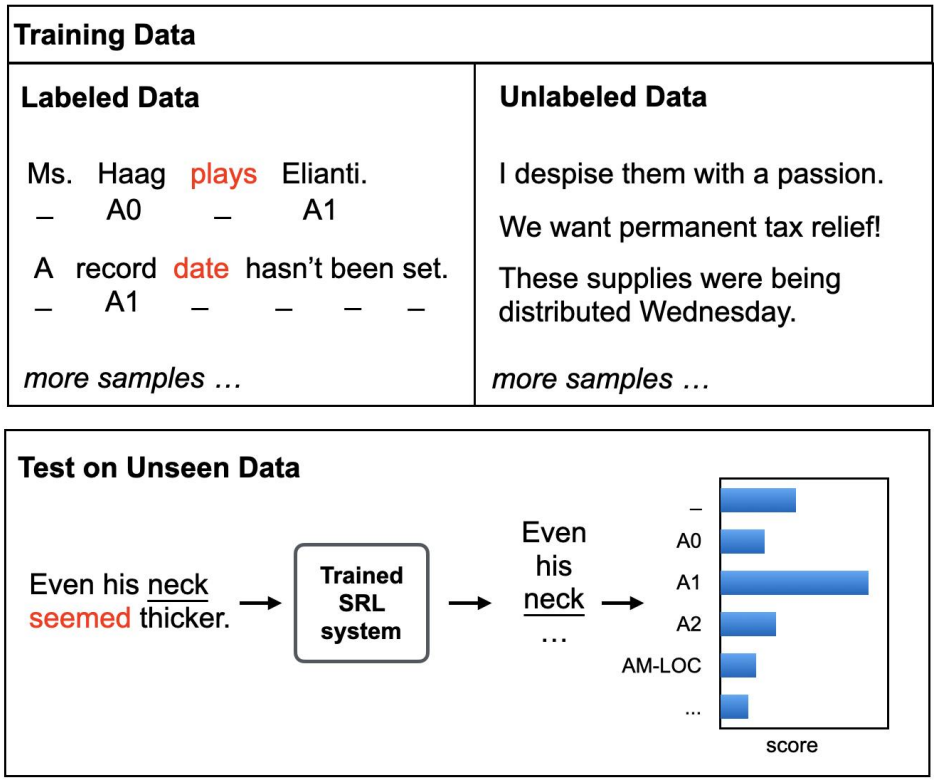

Figure 3.2: A toy example of semi-supervised semantic role labeling. Samples in the unlabeled dataset are plain text without annotations of predicates and arguments.

be possible to generalize from a finite training set to a set of possibly infinite unseen test cases.

Semi-supervised learning can be further divided into *inductive* learning and *transductive* learning. The former aims to output a prediction function, which is defined on the entire data space, while *transductive* learning tries to perform predictions only for the test points. In other words, pure semi-supervised learning is based on an "open world" assumption, entailing the hope that the learnt model can be applied to unobserved data during the training process. Conversely, *transductive* learning is based on a "closed world" assumption, which only tries to understand the unlabeled data observed during the learning process. In this thesis, we adopt the setting of *inductive* learning.

In the context of semantic role labeling, high-quality annotations (of semantic predicates and their arguments) are expensive, and mostly unavailable in low resource scenarios (e.g., rare languages or domains). This underpins the need for effective semi-supervised methods that leverage unlabeled examples. Compared with labeled data,

unlabeled data (e.g., Wikipedia articles that are freely available online) are widely diffused on websites and thus can be easily obtained. Sizes of unlabeled datasets tend to be much larger than those of labeled datasets, and have been successfully used in various NLP tasks (e.g., pretrained language models in Mikolov et al. (2013) and Devlin et al. (2018)).

Formally, semi-supervised SRL requires two datasets: a labeled dataset C consisting of labeled training instances $\{(s_i, g_i)\}_{i=1}^{|C|}$ and an unlabeled dataset U consisting of unlabeled training instances $\{s_i\}_{i=1}^{|U|}$. Definitions of s_i and g_i are the same as in Section 3.1. The output of the semi-supervised SRL system is the normalized probability distribution over all roles, which is defined in Equation (3.1).

3.3 Unsupervised Semantic Role Labeling

Both supervised and semi-supervised semantic role labeling require a predefined role set and a dataset with gold annotations. Therefore, these methods are limited by their reliance on the manually role-tagged corpora such as FrameNet or PropBank, which are expensive to produce and limited in size. To avoid the need for expensive manual labeling of text, unsupervised semantic role labeling (Lang and Lapata, 2010, 2014) performs semantic analysis without annotations that indicate predicates, arguments, or semantic roles.

The process of unsupervised semantic role labeling can be divided into three steps. The first step is predicate identification, which has been described in Section 2.3.2. The second step is argument identification, which aims to filter non-argument tokens and preserve all candidates that are likely to be an argument. Some non-argument candidates might pass the filtering and be reserved. This is permissible, as these non-argument candidates can be grouped into a separate cluster during argument classification.

The final step of unsupervised semantic role labeling is argument classification, and this differs fundamentally from the supervised setting. Since in the unsupervised setting there is no predefined role set, semantic roles must be induced from the data

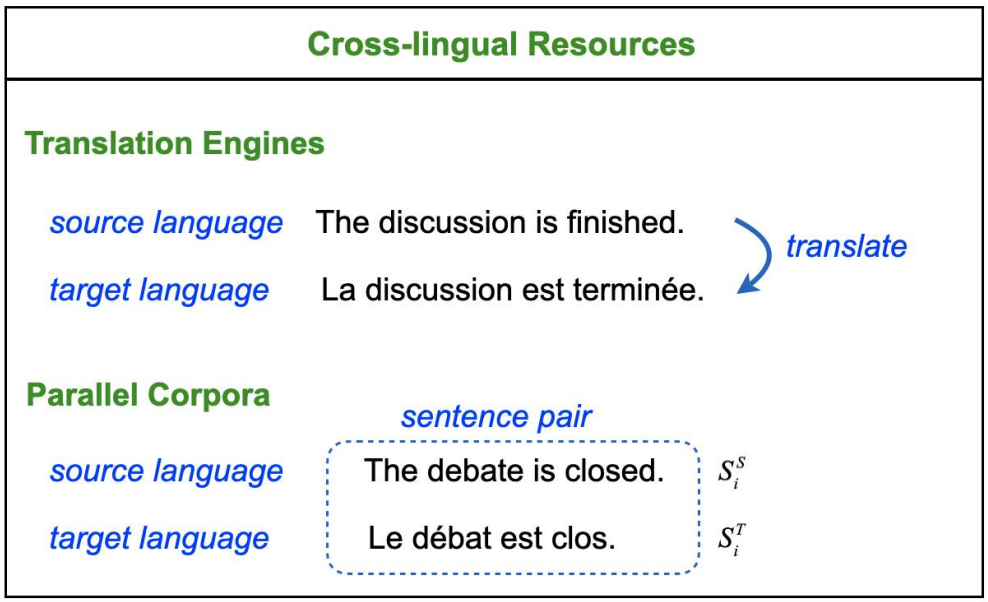

Figure 3.3: Cross-lingual resources in cross-lingual SRL.

itself, and therefore this process is also referred to as *role induction*. Role induction performs argument classification by grouping arguments into clusters, and each cluster represents a semantic role. After role induction, each cluster can be assigned a semantic role by a human being. Alternatively, clusters can be labeled automatically with roles such as A0, A1, and so on, similar to the core roles used in PropBank.

3.4 Cross-lingual Semantic Role Labeling

For both supervised and semi-supervised settings, labeled datasets are vital, directly to provide supervision for model training. For popular languages (e.g., English), we have access to a large quantity of labeled data. Nonetheless, semantic role annotations are not available for low-resource languages. Cross-lingual SRL (Aminian et al., 2019; Kozhevnikov and Titov, 2013) offers the possibility of learning models for a low-resource language (target language) using annotated data from other languages (source language).

Formally, cross-lingual SRL requires a labeled dataset C^{SL} for the source language consisting of labeled training instances $\{(s_i^{SL}, g_i^{SL})\}_{i=1}^{|C^{SL}|}$. Moreover, some cross-lingual

resources can be also utilized: with the development of neural machine translation, some translation tools (e.g., Google Translate[1]) with robust performance are available free of charge, in contrast to annotations. Parallel datasets (e.g., Europarl Parallel Corpus[2]) are also available for many languages. As shown in Figure 3.3, translation engines can be used to translate a source-language sentence s_i^{SL} into a target-language sentence. A parallel corpus C^P consists of source-target sentence pairs $\{(s_i^S, s_i^T)\}_{i=1}^{|C^P|}$ without SRL annotations, where s_i^T can be regarded as the result of translating sentence s_i^S into the target language, and vice versa. Although cross-lingual resources do not contain SRL annotations for target-language sentences, they are the basis for performing cross-lingual testing. For testing, we apply a trained system to an unseen target-language labeled dataset C^{TL}, which consists of labeled testing instances $\{(s_i^{TL}, g_i^{TL})\}_{i=1}^{|C^{TL}|}$.

It is straightforward to obtain an SRL model for the source language for which labeled data are available. In this way we obtain a mono-lingual semantic role labeler for the source language, and its parameters are optimized to learn conditional probability distributions $P(g^{SL}|s^{SL})$ of that source language. In practice, due to the differences between source and target languages, s^{SL} and s^{TL} are distributed in different areas of the input space.

Hence, feeding s^{TL} directly to the source-language labeler as input usually fails to yield satisfying results. Cross-lingual resources provide opportunities to transfer the supervision signal from the source language to the target, while not relying on human annotations on the target side. The evaluation of a cross-lingual labeler is straightforward: we test the labeler on unseen target-language samples, the semantic roles of which are given. The reader should note that there is no source-language sample in the test set, since we are concerned only with performance regarding the target language. Figure 3.4 illustrates the training and evaluation of a cross-lingual SRL system.

[1] https://translate.google.com/
[2] https://www.statmt.org/europarl/

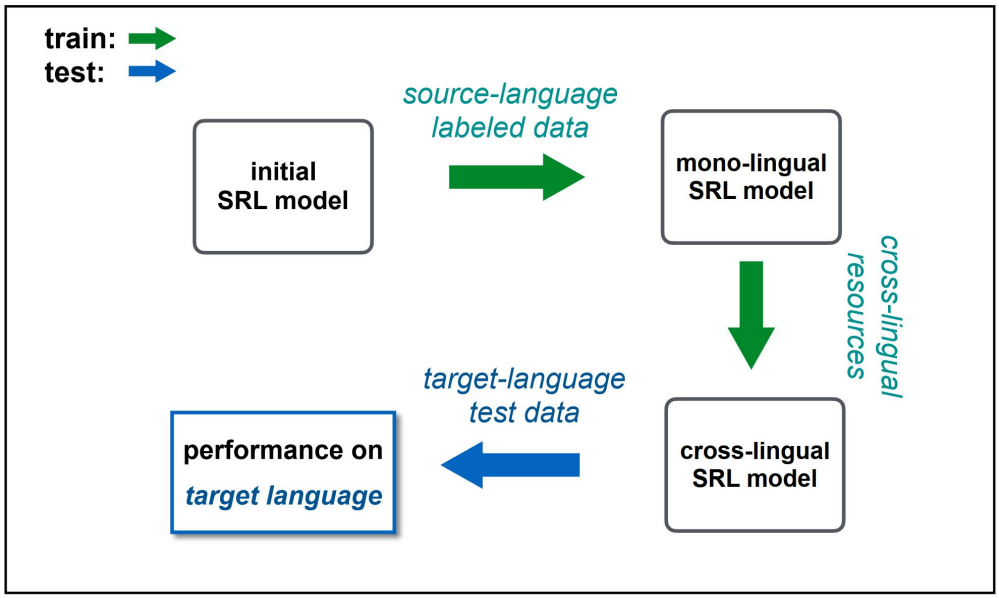

Figure 3.4: Steps of cross-lingual SRL.

3.5 Summary

In this chapter, we have introduced different learning settings for semantic role labeling. Supervised semantic role labeling relies purely on large quantities of hand-labeled sentences, which are costly to obtain. Trained SRL models produce a probability distribution over semantic roles and are tested on unseen samples. In order to improve the performance of SRL systems without the need for more hand-labeled instances, semi-supervised SRL draws upon unlabeled data, which can be obtained more easily. When labeled data are not available, unsupervised semantic role labeling can still classify arguments by grouping them into different clusters, each of which represents a semantic role.

In contrast to those monolingual SRL systems, cross-lingual semantic role labeling tries to transfer a system trained on a source language to a target language, with no access to target-language labeled data. When performing cross-lingual training, certain cross-lingual resources, such as a parallel corpora and translation tools, are employed. The test set only contains target-language samples, as we are only concerned with performance in terms of the target language.

Chapter 4

Supervised Semantic Role Labeling

In this chapter, we present *a supervised approach* for the detection of semantic roles in sentences. Our goal is to design neural models that learn to label semantic roles with the help of dependency information. Instead of relying on external dependency parsing tools, we enable our model to extract dependency features by training it with gold dependency annotations in a supervised fashion.

To that end, besides an LSTM-based semantic role labeler, we propose a dependency information extractor jointly trained with two *auxiliary* tasks: dependency label prediction and link type prediction. The former focuses on predicting the dependency labels of predicates as opposed to all words (labels of red arcs in Figure 4.1) in a dependency tree. The latter aims to capture how semantic predicates are linked to adjacent words in a sentence. Specifically, we are interested in predicting whether they are linked, and, if they are, what type of link they have. The auxiliary tasks provide syntactic information that is specific to semantic role labeling and is learned from training data (dependency annotations) without relying on existing dependency parsers, which can be noisy (e.g., on out-of-domain data or infrequent constructions).

We evaluate our model on different domains and multiple languages. Experimental results on the CoNLL-2009 benchmark show that our model outperforms baseline models in English (on in-domain and out-domain test data), and additionally improves SRL performance in other languages, including Chinese, Czech, and Spanish.

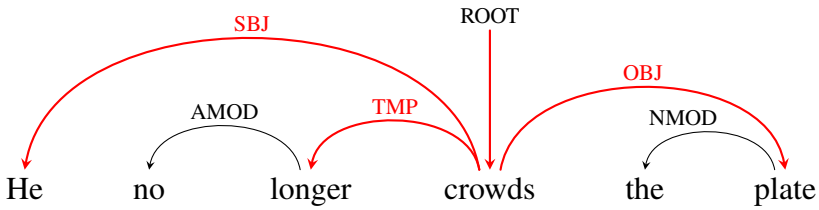

Figure 4.1: Dependency structure of sentence *He no longer crowds the plate*. Dependency arcs and labels pertaining to the predicate *crowds* are in red.

4.1 Introduction

The successful application of neural networks to various NLP tasks (Bahdanau et al., 2015; Vinyals et al., 2015) has provided a strong impetus to develop end-to-end models for semantic role labeling, which forego the need for extensive feature engineering. Some recently proposed methods (He et al., 2018; Li et al., 2019b; Marcheggiani et al., 2017) rely on bi-directional recurrent neural networks (Hochreiter and Schmidhuber, 1997) and predict semantic roles from textual input.

Previous work (He et al., 2017; Marcheggiani et al., 2017) has achieved competitive results while being syntax agnostic, thereby challenging the conventional wisdom that parse trees provide a better form of representation for assigning semantic role labels (Johansson and Nugues, 2008). There are, however, good reasons why syntax ought to help semantic role labeling. We show an example with role labels in the style of PropBank (Palmer et al., 2005) and gold dependency structure in Figure 4.2. As we can see, many arcs in the syntactic dependency graph are mirrored in the semantic dependency graph (e.g. A0 and SBJ), suggesting that syntactic dependencies could provide useful information to the SRL task. Furthermore, predicates are typically associated with a *standard* linking: a deterministic mapping from syntactic roles to semantic ones (Lang and Lapata, 2010). For example, object (OBJ) is commonly mapped onto role A1, whereas A0 is often realized as a subject (SBJ) . Although there are no such deterministic mappings in some cases, semantic roles are still related to certain dependency labels, such as the syntactic label TMP and the semantic role AM-TMP.

The relatedness between syntactic labels and semantic roles has provided motivation (Marcheggiani and Titov, 2017; Roth and Lapata, 2016) to incorporate syntactic information into neural models in semantic role labeling. One major obstacle is the

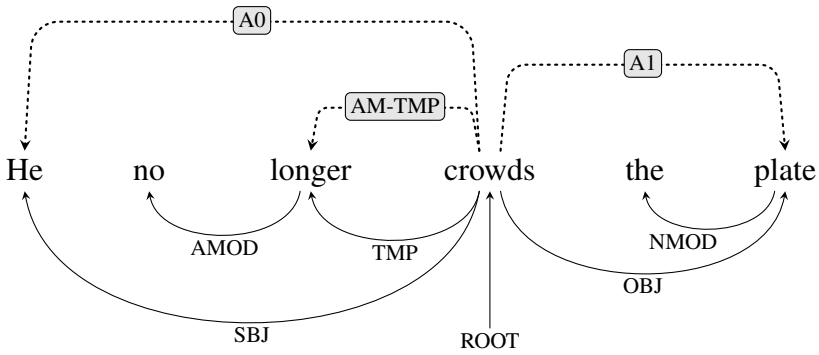

Figure 4.2: Example sentence from the CoNLL-2009 English dataset annotated with syntactic dependencies (bottom) and semantic roles (top).

tree structure of dependency labels, which prevents dependency labels from being directly fed as input of sequence-modeling encoders like LSTMs and transformers. The question of how to incorporate structural dependency information has met different answers in the literature. Instead of extracting features from the entire dependency tree, *dependency paths* have been widely used instead of tree structures not only for the SRL task (Roth and Lapata, 2016) but also for other downstream tasks such as relation classification (Cai et al., 2016; Liu et al., 2015). *Dependency paths* can be taken as input conveniently; however, many useful words and syntactic information outside these paths are discarded. Marcheggiani and Titov (2017) make use of graph convolutional networks (GCNs; Duvenaud et al. 2015; Kearnes et al. 2016; Kipf and Welling 2017), which is a recent class of multilayer neural networks operating on graphs. Each word in the input sentence is regarded as a node, and GCNs are used to encode relevant information about its neighborhood in syntactic dependency trees as a real-valued feature vector. As a one-layer GCN encodes only information about immediate neighbors, multiple layers are stacked together to encode K-order neighborhoods (i.e., information about nodes at most K hops aways).

Another problem is that gold dependency trees are not available at test time. Despite recent advances in dependency parsing (Dozat and Manning, 2016; Kiperwasser and Goldberg, 2016), utilizing the output of external dependency parsers inevitably introduces noise for SRL systems, where errors propagate to later processing stages and thereby affect model performance. To mitigate the impact of noisy parses, Marcheggiani and Titov (2017) calculate a scalar gate for each edge in the dependency tree. However, their system's performance decreases when more than one GCN layers are stacked, and this is perhaps caused by noisy information amplified by the stacking

operation.

In this chapter, we argue that syntactic information is important for semantic role labeling while syntactic parsers are not. Our key idea is to focus on dependency features which provide important information for SRL without full syntactic analysis of input sentences. Instead of extracting information from every node and edge in dependency trees, we concentrate on the dependency structures pertaining to the current predicate. According to statistics from the CoNLL-2009 English development set, a majority of arguments (68%) are directly linked with predicates in dependency trees or are predicates themselves.

The backbone of our model is an LSTM based semantic role labeler jointly trained with a *dependency information extractor* performing two auxiliary tasks: predicting the dependency label and link type between a word and the predicate. The extractor provides dependency information that is specific to the SRL task and is learned from training data (gold dependency annotations) without ever utilizing an external parser. Inspired by ELMo (Peters et al., 2018), we also utilize the combination of the intermediate-layer representations in the dependency information extractor. The embeddings of dependency information and intermediate-layer representations compose *syntax-aware representations*, which are fed to the semantic role labeler as a part of input. The main body of our semantic role labeler contains (1) a stacked bidirectional Long Short-term Memory neural network (BiLSTM) serving as the sentence encoder, and (2) a biaffine attentional scorer (Dozat and Manning, 2016), which predicts the semantic role for each word in the input sentence. Experimental results on the CoNLL-2009 benchmark dataset show that our model can outperform baselines in English, and improve SRL performance in other languages, including Chinese, Czech, and Spanish.

4.2 Dependency Information Extractor

Unlike other syntax-aware SRL systems (He et al., 2018; Marcheggiani and Titov, 2017) relying on external parsing tools, we introduce a *dependency information extractor* to obtain dependency information about semantic predicates in input sentences.

The extractor is trained for two auxiliary tasks: *dependency label prediction* and *link type prediction*. Both tasks focus on predicting the dependency information of predicates as opposed to all words in a dependency tree. As predicate information is required by the dependency information extractor, it operates over sentences after predicate identification (and disambiguation) has taken place.

4.2.1 Dependency Information Pertaining to Predicates

Dependency parsing tries to analyze the grammatical structure of a sentence and establish the relationships between "head" words and their dependents. Basically, a dependency relationship consists of a head (H), a dependent (D) and a label identifying the relation between H and D. The dependency structure of a sentence is a tree with the words of the sentence as its nodes, among which the only one is linked with a virtual root node (ROOT). Figure 4.1 gives the dependency structure of sentence *He no longer crowds the plate*, where *crowds* is a predicate and related to ROOT. To ensure the dependency structure is a tree, it usually has the following properties:

1. *connected*: every node is related to at least one other node;
2. *single headed*: every node (except ROOT) has exactly one incoming edge (from its head);
3. *acyclic*: the graph cannot contain cycles of directed edges.

There are mainly two types of dependency parsers: graph-based dependency parsing based on maximum spanning trees (Dozat and Manning, 2016) and transition-based dependency parsing which are an extension of shift-reduce parsing (Kiperwasser and Goldberg, 2016). These parsers try to find every head-dependency pair present in sentences, and the time complexity of graph-based approaches can reach $O(n^2)$. We observe that the majority of arguments (approximately 68%) in the CoNLL-2009 English development set are directly linked to the predicate or are predicates themselves. In this chapter, rather than capturing information relating to every arc in the dependency tree, we concentrate on dependency structures pertaining to the predicate in a given sentence (e.g., red-colored arcs in Figure 4.1). In this way, dependency information unrelated to predicates is filtered out. Moreover, compared with tree structures, remaining dependency information can be more easily fed to sequence modeling neu-

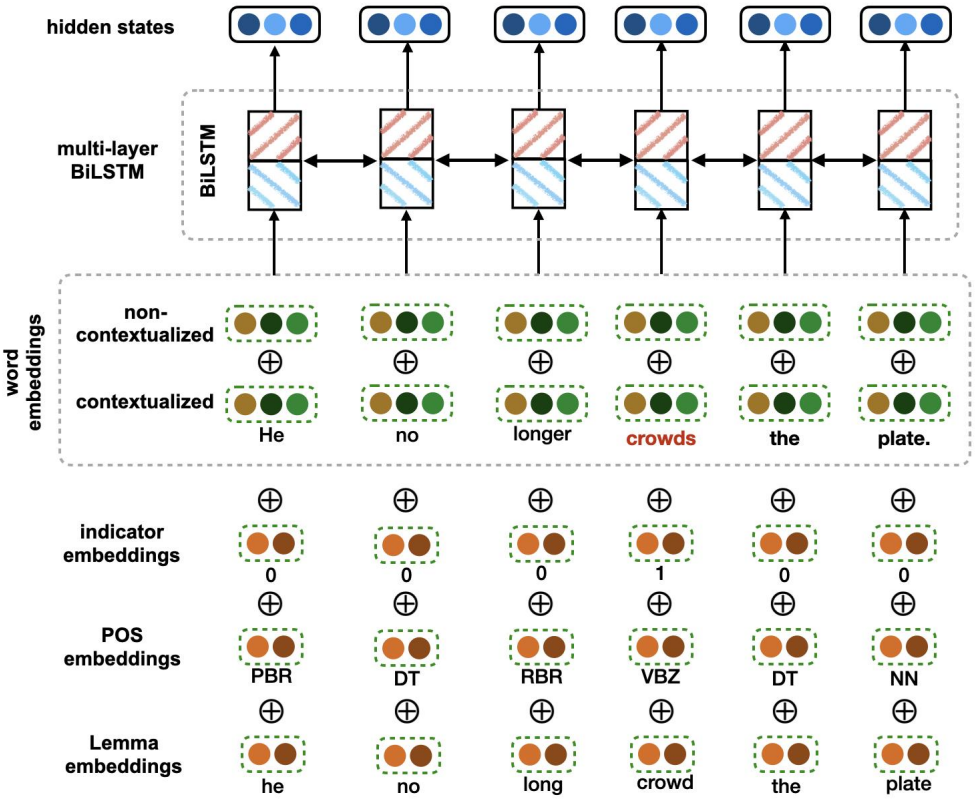

Figure 4.3: Sentence encoder in dependency information extractor.

ral networks.

4.2.2 Sentence Encoder

Let $\hat{x}_w$ denote the representation of word w in a sentence of length n, and $\hat{x}_w$ is the concatenation of a few vectors, i.e., a randomly initialized word embedding $\hat{x}_w^{re}$, a pre-trained word embedding $\hat{x}_w^{pe}$, a randomly initialized part-of-speech tag embedding $\hat{x}_w^{pos}$ and a randomly initialized embedding $\hat{x}_w^{flag}$, which is a binary flag indicating whether the current word is a predicate. Additionally, we further enhance the word representation by concatenating an external embedding $\hat{x}_w^{lm}$ from a pretrained successful language models, ELMo (Embeddings from Language Models) (Peters et al., 2018). Compared with $\hat{x}_w^{re}$ and $\hat{x}_w^{pe}$, $\hat{x}_w^{lm}$ is contextualized and fixed during training. The word representation is thus given by $\hat{x}_w = \hat{x}_w^{re} \circ \hat{x}_w^{pe} \circ \hat{x}_w^{pos} \circ \hat{x}_w^{flag} \circ \hat{x}_w^{lm}$, where $\circ$ represents the concatenation operator.

Word representations are contextualized with a bi-directional re-current neural net-

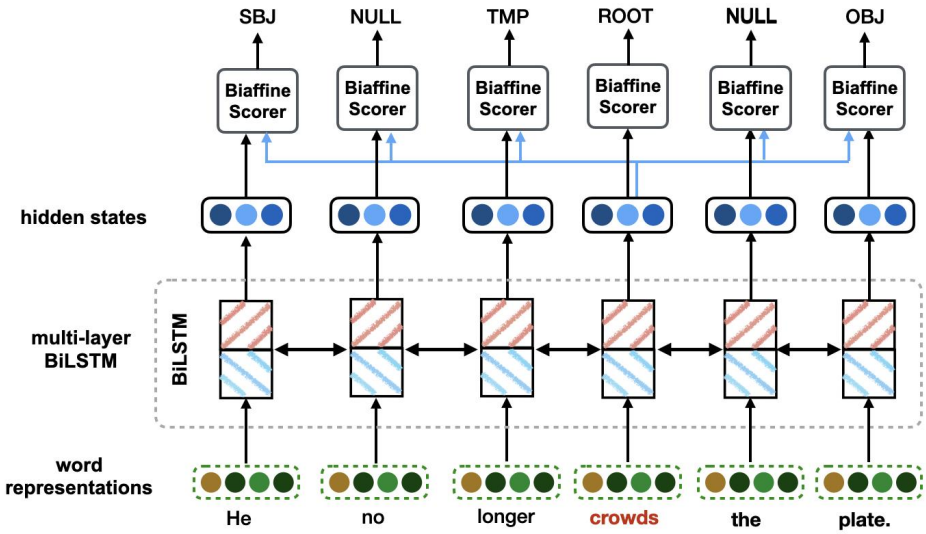

Figure 4.4: The dependency label predictor outputs labels of each word in the sentence *He no longer crowds the plate* and *crowds* is namely the current predicate.

work with long-short term memory units (LSTM;Hochreiter and Schmidhuber (1997)). A bidirectional LSTM receives at time step t a representation $\hat{x}_t$ for each word and recursively computes two hidden states, one for the forward pass ($\overrightarrow{h_t}$), and another one for the backward pass ($\overleftarrow{h_t}$). Each word is the concatenation of its forward and backward LSTM $h_t = c\overrightarrow{h_t} \circ \overleftarrow{h_t}$. Figure 4.3 illustrates the architecture of our sentence encoder.

4.2.3 Dependency Label Prediction

As discussed in Section 4.2.1, our model focuses on predicting the dependency labels of predicates as opposed to all words in a dependency tree. Taking Figure 4.1 as an example, our model solely tries to output the labels of arcs related to the predicate (red-colored). For each arc (w, p) linking predicate p and modifier w, our model assigns the dependency label l with the highest score according to a biaffine scorer ($\text{Biaffine}_{\mathcal{LBL}}$):

$$label(w,p) = \arg\max_{l \in labels} \text{Biaffine}_{\mathcal{LBL}}[l] \tag{4.1}$$

$$= \arg\max_{l \in labels} \text{Biaffine}(h_p, h_w, l) \tag{4.2}$$

where l are pre-defined dependency labels (e.g., SBJ, OBJ), h_p and h_w are hidden states for predicate p and modifier w respectively, produced by bidirectional sentence encoder

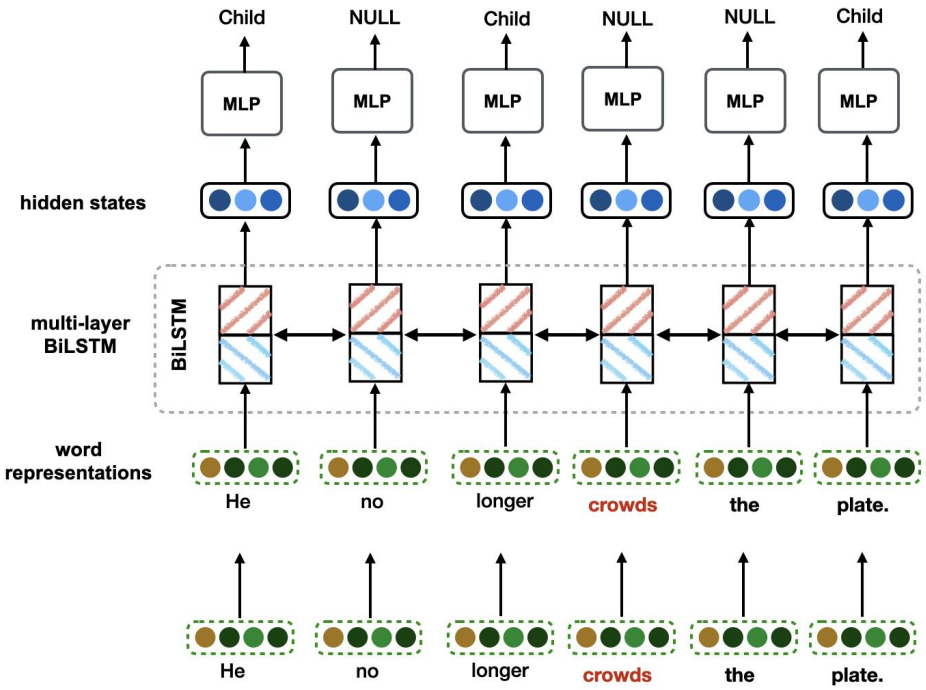

Figure 4.5: Link type predictor outputs a label for each word in sentence *He no longer crowds the plate* and *crowds* is the predicate of interest.

in Figure 4.3. The inner structure and output labels of dependency label predictor is shown in Figure 4.4. The score of dependency label l between predicate p and word w is calculated by a Biaffine$_{\mathcal{LBL}}$ scorer as follows:

$$\text{Biaffine}(h_p, h_w, l) = h_p W_l h_w + U_l(h_p \circ h_w) + b_l \tag{4.3}$$

where W_l, U_l, and b_l are parameters updated during training.

4.2.4 Link Type Prediction

In Section 4.2.3, our model tries to predict dependency labels of arcs pertaining to the predicate of interest while ignoring the direction of these arcs. Here we aim to capture how semantic predicates are linked to adjacent words in a sentence. Specifically, we are interested in predicting whether they are linked, and, if they are, what type of link they have. Analogously to dependency label prediction, we only focus on dependency arcs pertaining to the predicate, rather than *all* arcs in the dependency tree, and assign each word a label representing its link type in relation to the predicate. Label "NULL"

indicates there is no arc between a word and the predicate, whereas labels "Child" and "Parent" represent child and parent nodes of the predicate, respectively. We extract predicate linking information from dependency trees, and use a MLP predictor to identify link type information for word w:

$$\text{MLP}_{\mathcal{LNK}} = W_{\mathcal{LNK}} \tanh(W_L h_w) \tag{4.4}$$

where $W_{\mathcal{LNK}}$ and W_L are parameter matrices. Different from the biaffine scorer in the dependency label predictor, $\text{MLP}_{\mathcal{LNK}}$ does not explicitly use h_p, namely, predicate-specific information. By doing this, we force the model to learn the linking between long-distant word pair. For a sentence with multiple predicates, the dependency information extractor will produce different results for each predicate.

4.3 Syntax-aware Semantic Role Labeler

Figure 4.6 presents an overview of our end-to-end SRL system. The sentence encoder of the semantic role labeler (the upper block in Figure 4.6) takes syntax-aware word representations as input and produces hidden states for each word. The semantic role labeler estimates the probability of role r given the hidden states of candidate argument word w and predicate word p:

$$p(r|h_w, h_p, l) \propto exp(W_{l,r}(h_w \circ h_p)), \tag{4.5}$$

where h_w and h_p are representations for word w and predicate p, respectively, and l is the lemma of predicate p; symbol $\circ$ denotes concatenation and $\propto$ signifies proportionality. Following Marcheggiani and Titov (2017), matrix $W_{l,r}$ is the joint embedding of role r and the predicate lemma l using a non-linear transformation:

$$W_{l,r} = \text{ReLU}(U(e_l \circ e_r)), \tag{4.6}$$

where U is a parameter matrix, and $e_l \in R^{d_l'}$ and $e_r \in R^{d_r}$ are randomly initialized embeddings of predicate lemmas and roles. This way, each role prediction is predicate-specific, and a good representation for roles associated with infrequent predicates can be learned.

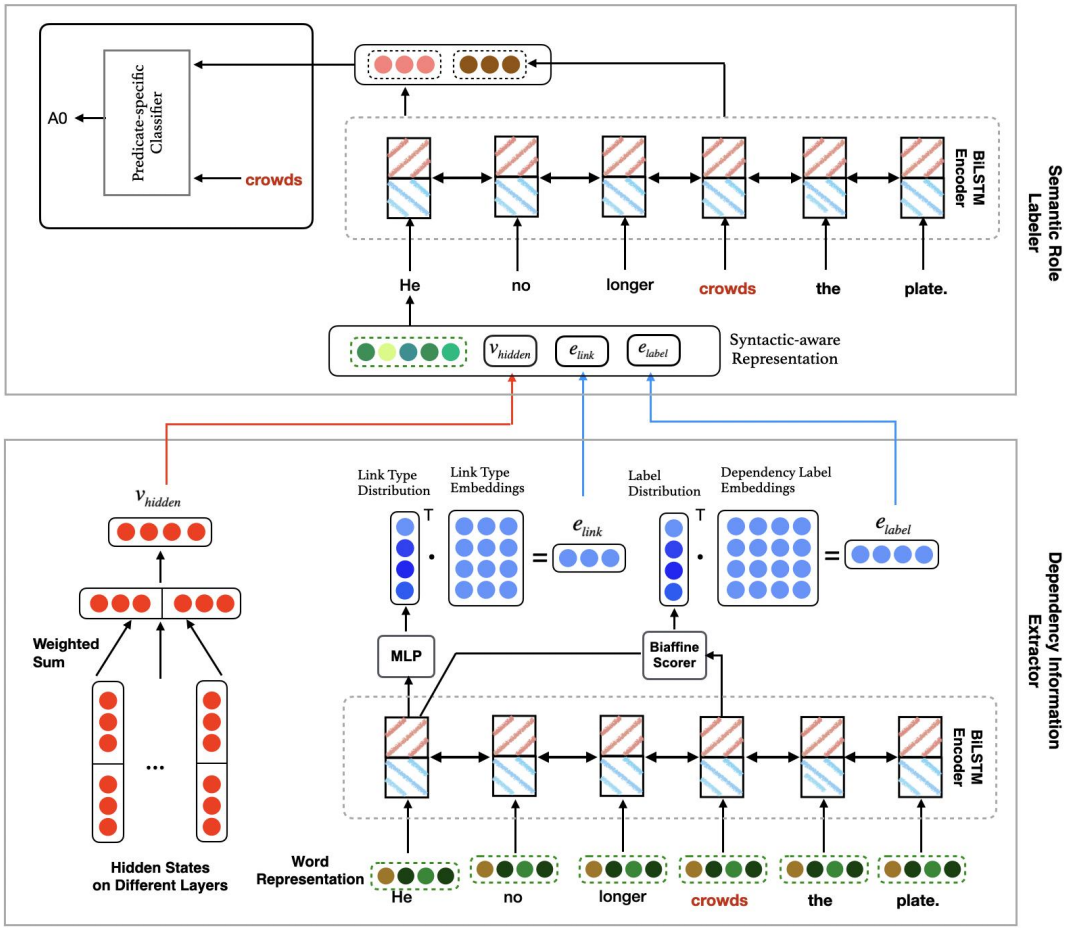

Figure 4.6: Model overview: Dependency information extractor (bottom) and a semantic role labeler (top). Colored lines are syntax-aware representations for the word *He* and are shared between the two components. In the semantic role labeler, the representation of each input token consists of its word embeddings and syntax-aware representations.

The model's training objective $\mathcal{L}$ is the weighted sum of objectives for the SRL task and the two auxiliary tasks. Formally,

$$\mathcal{L} = \mathcal{L}_{\mathcal{SRL}} + \alpha(\mathcal{L}_{\mathcal{LBL}} + \mathcal{L}_{\mathcal{LNK}}), \tag{4.7}$$

where $\mathcal{L}_{\mathcal{SRL}}$, $\mathcal{L}_{\mathcal{LBL}}$ and $\mathcal{L}_{\mathcal{LNK}}$ are the categorical cross-entropy of SRL, dependency label prediction and link type prediction tasks, respectively. α is a scalar weight for the auxiliary tasks, whose value is tuned experimentally on the development dataset.

4.3.1 Input Layer Representations

Given a sentence of length n, we use x to denote the representation of each word. Similar to the input of dependency information extractor, x is a concatenation of multiple vectors, including a randomly-initialized word embedding x^{re}, a pre-trained word embedding x^{pe}, a randomly initialized POS tag embedding x^{pos}, a predicate indicator embedding x^{flag}, and pre-trained language model layer features x^{lm}. The word representation is thus given by $x = x^{re} \circ x^{pe} \circ x^{pos} \circ x^{flag} \circ x^{lm}$, where $\circ$ represents the concatenation operator.

Additionally, in order to obtain more syntax-aware representations, we take advantage of hidden-layer representations v_{hidden}, dependency label embedding e_{label} and link type embedding e_{link}. These three syntactically informed representations are concatenated together with x, and serve as input R to our semantic role labeler:

$$R = x \circ v_{hidden} \circ e_{label} \circ e_{link} \tag{4.8}$$

4.3.2 Hidden-layer Representations

There are multiple hidden layers in the sentence encoder of the dependency information extractor, which raises the question of how to utilize these hidden layers effectively. We draw inspiration from ELMo (Peters et al., 2018), a popular pre-trained language model based on bidirectional LSTMs. Unlike more traditional word embeddings (Mikolov et al., 2013), word representations in ELMo are deep and obtained by linearly combining the representations learned at all layers of the LSTM instead of just the final layer.

Similar with ELMo, we also utilize the combination of the hidden-layer representations in the dependency information extractor. For each word in the input sentence, the sentence encoder (based on a BiLSTM) with L layers produces a set of $2L$ repre-

sentations:

$$\begin{aligned} S &= \{\overrightarrow{h}_j, \overleftarrow{h}_j | j = 1, ..., L\} \\ &= \{h_j | j = 1, ..., L\}, \end{aligned} \tag{4.9}$$

where $h_j = [\overrightarrow{h}_j; \overleftarrow{h}_j]$ for each hidden layer in the BiLSTM encoder. We then collapse these representations into a single vector. Although we can simply concatenate them, we compute a single vector v_{hidden} as a weighting of all BiLSTM layers, followed by a non-linear projection:

$$v_{hidden} = \text{ReLU}(W_{hidden}(\gamma \sum_{j=1}^{j=L} \beta_j h_j)) \tag{4.10}$$

where β are softmax-normalized weights for h_j; the scalar parameter γ is of practical importance for optimization, and has shown to be effective in Peters et al. (2018). Both β and γ are updated during training.

4.3.3 Dependency Embeddings

A simple way to utilize the result of dependency label prediction would be to use the embedding e_l of the label l with the highest confidence score. However, this would place too much emphasis on the highest confidence labels, which can be erroneous. Instead, we use the weighted composition of all dependency label embeddings e_{label}, which is calculated as:

$$e_{label} = \sum_{l \in labels} \text{softmax}(\text{Biaffine}_{\mathcal{LBL}})[l] * e_l \tag{4.11}$$

where the weight of each label embedding is the normalized probability given by the label classifier. Analogously, we represent dependency link information e_{link} as:

$$e_{link} = \sum_{l \in types} \text{softmax}(\text{MLP}_{\mathcal{LNK}})[l] * e_l \tag{4.12}$$

4.4 Experiments

In this section we present our experimental evaluation and analysis on the CoNLL-2009 benchmark datasets across four languages (English, Chinese, Czech, and Spanish). We first present details on implementation and training, Our results are discussed in Section 4.4.3. Ablation experiments and analysis are presented in Section 4.4.5, which tries to evaluate the contribution of various model components.

4.4.1 Datasets

For years *the Conference on Computational Natural Language Learning* (CoNLL) has been accompanied by a shared task whose purpose is to promote natural language processing applications (Hajič et al., 2009a). In 2009, the shared task was dedicated to the joint parsing of syntactic and semantic dependencies in multiple languages. We evaluated our model on the English, Chinese, Czech, and Spanish CoNLL-2009 benchmarks following the standard training, testing, and development set splits. The datasets contain gold-standard dependency annotations, and also gold lemmas, part-of-speech tags, and morphological features. Data for the different languages was generated by merging various language specific treebanks such as the Penn Treebank (Marcus et al., 1993b) and Brown corpus (Francis and Kucera, 1979) for English, the Prague Dependency Treebank for Czech (Hajičová et al., 1999), the Chinese Treebank (Xue et al., 2005) and the Proposition Bank (Xue and Palmer, 2009) for Chinese, and so on (we refer the interested reader to Hajič et al. 2009a for details on the individual languages and their annotations).

4.4.2 Model Settings

For English, we used the 100-dimensinal Glove vectors of (Pennington et al., 2014). For Chinese, Spanish, and Czech, 300-dimensional word embeddings were used pre-trained on Wikipedia using `fastText` (Bojanowski et al., 2017). For experiments on English, we added contextualized ELMo embeddings (Peters et al., 2018) into the input

Hyperparameter	Value	
	Labeler	*Extractor*
English word embeddings size	100	100
other languages embeddings size	300	300
character embeddings size	300	300
lemma embeddings size	100	100
POS embeddings size	16	16
LSTM hidden states size	300	300
input vectors size of biaffine scorer	300	200
BiLSTM depth	4	2
hidden-layer representations size (v_{hidden})	128	—
dependency embeddings size (e_{link} & e_{label})	32	—
batch size	30	30
input layer dropout rate	0.3	0.3
hidden layer dropout rate	0.3	0.3
learning rate	0.001	0.001
auxiliary tasks loss weight α	—	0.5

Table 4.1: Values of hyperparameters in semantic role labeler (*Labeler*) and dependency information extractor (*Extractor*)

of our model. ELMo is purely character-based and pretrained on the 1 Billion Word Benchmark[1], which is publicly released as part of the AllenNLP toolkit.[2]

The dropout mechanism was applied to the input layer and the top hidden layer of the BiLSTM encoders. We used the Adam optimizer (Kingma and Ba, 2014) to train our models. We performed hyperparameter tuning and model selection on the English development set; optimal hyperparameter values (for all languages) are shown in Table 5.1. Predicted POS tags were provided by the CoNLL-2009 shared-task organizers. We used the same predicate disambiguator as Roth and Lapata (2016) for all languages which employ a pipeline of mate-tools.

4.4.3 Results

Experimental results on the CoNLL-2009 English in-domain test set are presented in Table 4.2, and are divided into three blocks. In the first block of the table we compared our system with models that use an external dependency parser. while in the second

[1] http://www.statmt.org/lm-benchmark/
[2] https://allennlp.org/elmo

Single Models (with external parser)	P	R	F_1
Björkelund et al. (2010)	87.1	84.5	85.8
Lei et al. (2015)	—	—	86.6
FitzGerald et al. (2015)	—	—	86.7
Roth and Lapata (2016)	88.1	85.3	86.7
Marcheggiani and Titov (2017)	89.1	86.8	88.0
He et al. (2018)	89.7	89.3	89.5
Single Models (w/o external parser)	P	R	F_1
Marcheggiani and Titov (2017)	88.7	86.8	87.7
He et al. (2018)	89.5	87.9	88.7
Ours (w/o ELMo)	90.7	88.5	**89.7**
Ours (with ELMo)	91.0	89.1	**90.1**
Ensemble Models	P	R	F_1
FitzGerald et al. (2015)	—	—	87.7
Roth and Lapata (2016)	90.3	85.7	87.9
Marcheggiani and Titov (2017)	90.5	87.7	89.1

Table 4.2: English results on the CoNLL-2009 in-domain (WSJ) test set.

block our system are compared with those that do not. Results of various ensemble SRL models are reported in the third block for comparison. For a fair comparison with some recent models (He et al., 2018; Marcheggiani and Titov, 2017), we present performances with and without contextualized word embeddings (ELMo). Most baseline models are based on BiLSTMs (He et al., 2018; Marcheggiani et al., 2017; Marcheggiani and Titov, 2017) or learning SLR-specific embeddings (FitzGerald et al., 2015; Roth and Lapata, 2016). Results for two strong symbolic models are also reported; these are based on tensor factorization (Lei et al., 2015) and a pipeline of modules that carry out tokenization, part-of-speech tagging, dependency parsing , and semantic role labeling (Björkelund et al., 2010).

As shown in Table 4.2, our model outperforms previous single and ensemble models, regardless of whether they utilize a dependency parser or not (the differences over the baseline models are statistically significant at $p < 0.05$ using stratified shuffling (Noreen, 1989)). With ELMo embeddings, our model outperforms the best baseline model (He et al., 2018) by 0.6% in F_1 score, which is a syntax-aware model pruning candidate arguments from the output of an external parser. It is worth noting that the performance of He et al. (2018) drops from 89.5% to 88.7% when an external parser is not available. Recall that our model extracts dependency information on its

Single Models (with external parser)	P	R	F_1
Björkelund et al. (2010)	75.7	72.2	73.9
Lei et al. (2015)	—	—	75.6
FitzGerald et al. (2015)	—	—	75.2
Roth and Lapata (2016)	76.9	73.8	75.3
Marcheggiani and Titov (2017)	78.5	75.9	77.2
He et al. (2018)	81.9	76.9	79.3
Single Models (w/o external parser)	P	R	F_1
Marcheggiani et al. (2017)	79.4	76.2	77.7
He et al. (2018)	81.7	76.1	78.8
Ours (w/o ELMo)	80.3	78.4	**79.4**
Ours (with ELMo)	81.0	78.8	**79.9**
Ensemble Models	P	R	F_1
FitzGerald et al. (2015)	—	—	75.5
Roth and Lapata (2016)	79.7	73.6	76.5
Marcheggiani and Titov (2017)	80.8	77.1	78.9

Table 4.3: English results on the CoNLL-2009 out-of domain (Brown) test set.

own, and thus is not affected by the availability and quality dependency parsers.

Table 4.3 presents the results on the out-of-domain English test set. Again we compare our model with the same models as in the in-domain case. As shown in Table 4.3, our approach achieves better performance than comparison systems even without utilizing ELMo embeddings (the differences over the baseline models are statistically significant at $p < 0.05$ using stratified shuffling(Noreen, 1989)). With ELMo, F_1 further increases from 79.4% to 79.9% . We suspect that our model can outperform comparison systems by a wide margin because it does not have to rely on syntactic parsers (He et al., 2018; Marcheggiani and Titov, 2017; Roth and Lapata, 2016) while our model does not. The performance of external parsing tools degrades significantly on out-of-domain data, and they produce noisy dependency trees, thereby hurting the performance of semantic role labelers relying on them. Since our model only uses hidden features extracted automatically and the weighted sum of output embeddings, rather than the results of any parser, it is less brittle in the out-of-domain setting.

Table 4.4 shows our results on Chinese, Czech, and Spanish. Although we have not performed any model selection for these languages (i.e., we directly used the same hyperparameters as in English), our approach still achieves better performance than

Chinese	P	R	F_1
Björkelund et al. (2009)	82.4	75.1	78.6
Roth and Lapata (2016)	83.2	75.9	79.4
Marcheggiani and Titov (2017)	84.6	80.4	82.5
He et al. (2018)	84.2	81.5	82.8
Ours	85.4	81.8	**83.6**
Czech	P	R	F_1
Björkelund et al. (2009)	88.1	82.9	85.4
Marcheggiani et al. (2017)	86.6	85.4	86.0
Ours	87.5	86.6	**87.1**
Spanish	P	R	F_1
Björkelund et al. (2009)	78.9	74.3	76.5
Roth and Lapata (2016)	83.2	77.4	80.2
Marcheggiani et al. (2017)	81.4	79.3	80.3
Ours	83.3	80.4	**81.9**

Table 4.4: Results on the CoNLL-2009 test sets for Chinese, Czech, and Spanish.

System	P	R	F_1
Ours	86.8	84.9	85.8
w/o sharing word embeddings	86.7	84.7	85.6
w/o hidden-layer representation	86.3	84.1	85.1
w/o output embeddings	86.5	84.5	85.5
w/o multi-task learning	85.8	84.2	84.9
with full parser	86.5	85.0	85.7
w/o joint training	86.2	84.7	85.4

Table 4.5: Ablation results on the CoNLL-2009 English development set.

baseline models.

4.4.4 Analysis

In order to investigate the contribution of different components of our model, we performed a series of ablation studies on the English development set. These ablation studies are performed without ELMo embeddings, as these embeddings could introduce external syntactic and semantic information, which might obscure any conclusions about the effectiveness of our model. Evaluations in these experiments excludes predicate disambiguation (i.e., the ablation study is conducted on gold predicates), since we want to fully focus on the SRL model per se.

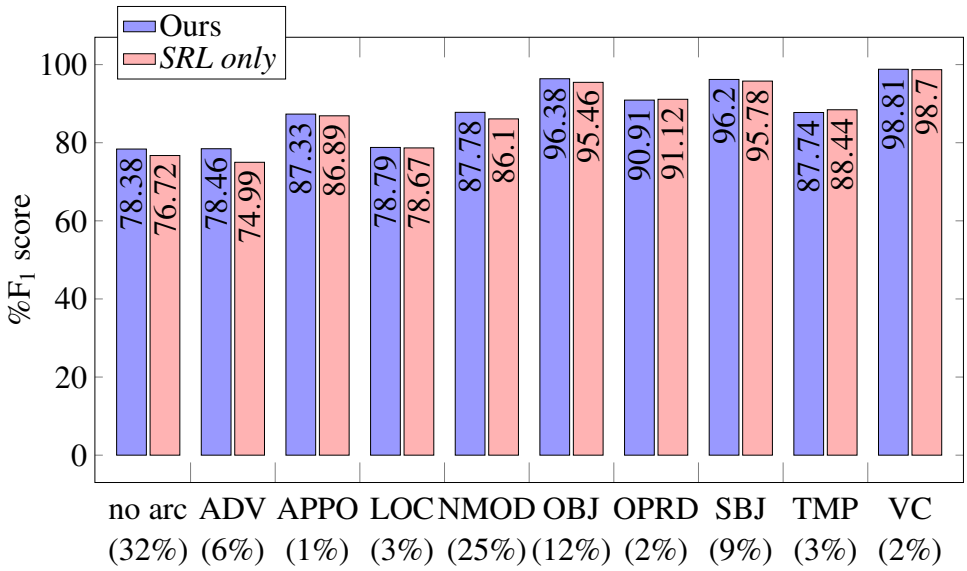

Figure 4.7: Semantic role labeling performance on the English CoNLL-2009 development set; roles are grouped into corresponding dependency relations whose proportional frequencies shown in parentheses (x-axis).

Table 4.5 presents the results of various ablation studies against our full model (first row). In the second block, we investigate the effect of different kinds of input representations. First, we verified the necessity of sharing word embeddings between the semantic role labeler and the dependency extractor. Experimental result shows that updating word embeddings separately hurts the model slightly. Secondly, we observe a 0.7% drop in F_1 is observed when hidden-layer representations are not fed to the semantic role labeler, indicating that hidden features captured by hidden layers for the dependency information extraction are helpful in the semantic role labeling task. Thirdly, we observe that not using the results of dependency label and link prediction slightly hurts the performance of semantic role labeler. This result indicates that our semantic role labeler does not rely on the performance of dependency extractor, instead, it just cautiously makes use of the results of the dependency extractor. The reason might be that we train the labeler and extractor simultaneously and at the beginning of training process the performance of extractor is quite low, as a result our semantic role labeler learns how to utilize the results of extractor gradually, instead of relying on it from the start. This makes our model more robust when the performance of the extractor drops (e.g., on out-of-domain test set).

In the third block of Table 4.5, we first examine whether multi-task learning is helpful to our SRL task. When removing the terms $(\mathcal{L}_{\mathcal{LBL}} + \mathcal{L}_{\mathcal{LNK}})$ from the training

objective (Equation 4.7) we observe a 0.9% drop in F_1. To further investigate the improvement brought by multi-task learning, we compare the full model against a model trained only for semantic role labeling (SRL *only*). Figure 4.7 shows results for both models according to different dependency labels. As can be seen, the full model outperforms SRL *only* on most dependency labels except OPRD and TMP, which only account for about 2% and 3% of semantic roles, respectively. Significant improvements are observed for semantic roles on NMOD, ADV, OBJ and SBJ, which appear more frequently in the developments set. In Table 4.6, we break down the results of two models into verbal and nominal predicates. Compared with verbal predicates, both models are relatively weak at predicting semantic roles pertaining to nominal predicates. Overall, our model outperforms SRL *only*, especially for nominal predicates, bring an overall improvement of 0.9% in F_1. In order to check whether improvements are due to a larger model, we added an additional (i.e., the fifth) BiLSTM layer to the encoder of SRL *only* and observed that the performance slightly decreases by 0.2% in F_1. This result indicates that only increasing the complexity of the model could not improve the performance of SRL.

Next, we exchanged the dependency extractor with a full parser, specifically, the graph-based neural parser proposed by Dozat and Manning (2016). Similar with the dependency extractor, the full parser is also built upon a multi-layer BiLSTM encoder, and its hidden layers and outputs are fed to the semantic role labeler as input. With the full parser our model achieves a performance of 85.7% in F_1, which is quite close to the model relying on the dependency extractor. This implies that our model is able to extract most SRL-related information contained in the dependency parser while avoiding the high overhead brought by full-blown parsing.

Finally, we substituted the dependency information extractor with an external full parser, and took one-best outputs instead of dependency embeddings (e_{label} and e_{link}) as input to the semantic role labeler. The external parser was pretrained and its parameters fixed during the training process of SRL model. Experimental results (see row "w/o joint training" in Table 4.5) show that compared that removing joint training further hurts SRL performance (by 0.3 F_1 points).

Verbal	Ours	*SRL only*	Frequency(%)
A0	93.0	92.2	15%
A1	93.5	92.8	21%
A2	84.5	82.8	5%
AM-*	80.9	80.1	16%
All	89.2	88.3	61%
Nominal	Ours	*SRL only*	Frequency(%)
A0	84.5	83.1	10%
A1	88.0	86.3	16%
A2	82.7	81.5	7%
AM-*	77.2	75.7	5%
All	84.7	83.2	39%

Table 4.6: F_1 results on the English test set broken down into verbal and nominal predicates.

4.4.5 Dependency Annotations

As described earlier, our model tries to extract dependency information without relying on external parsing tools. As a result, it requires gold dependency annotations to train the dependency extractor. The datasets we used provide gold standard semantic and syntactic annotations for every sample, which greatly facilitate model training. However, manual annotations are expensive and not always available, so we further conducted experiments to see whether it is possible to obtain competitive performance with less labels.

Figure 4.8 shows how F_1 changes when different amounts of dependency annotations are employed for training. As in our previous ablation studies, these experiments did not use character embeddings and the accuracy of predicate disambiguation was not taken into consideration. A subset of training samples was randomly selected (e.g., 10%, 20% and so on) with dependency annotations, and if the input sample was in the subset, we updated model parameters during training according to the combined loss of the SRL (see Equation 4.7) and auxiliary tasks, otherwise parameters were updated for the SRL task only (e.g., the training objective is $\mathcal{L}_{\mathcal{SRL}}$).

We can see from Figure 4.8 that the performance of our model increases gradually with more dependency annotations provided. Interestingly, a large jump in performance (F_1 improves from 84.5% to 85.8%) can be observed with only 10% of the total dependency annotations. The model's performance becomes competitive when

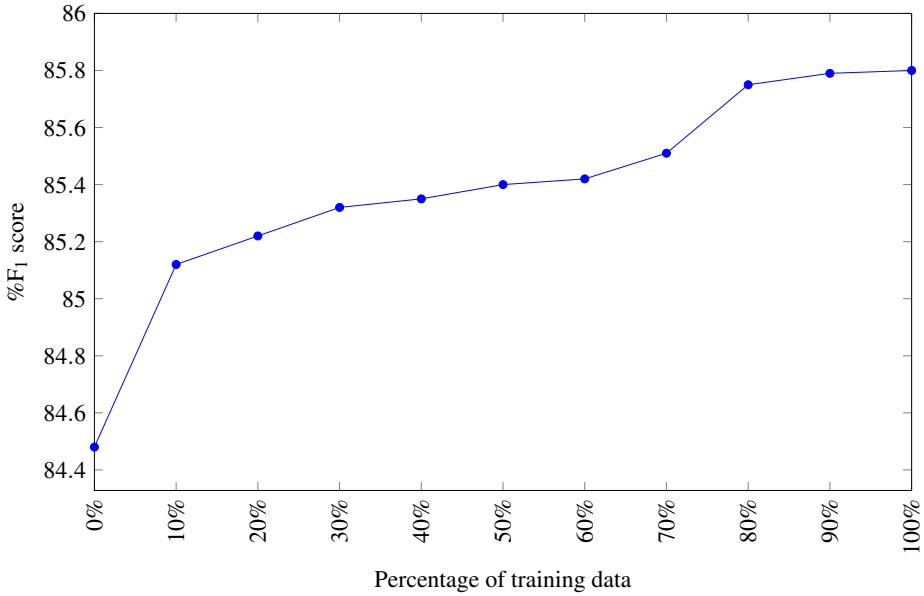

Figure 4.8: Semantic role labeling performance on the English CoNLL-2009 development set when different proportions of dependency annotations are used for training.

80% of the annotations are used, and remains stable when more annotations are provided. These results indicate that our model can also work effectively with only a small amount of gold dependency labels, and its performance becomes stable when a certain number of dependency labels are given.

4.4.6 Summary

In this chapter, we described a supervised semantic role labeler which is syntax-aware while not relying on any external dependency parsers. Experimental results across languages have shown that our model outperforms competitive baselines. Through ablation analysis we have also confirmed that sharing word embeddings, hidden-layer representations and joint training all contribute in improving the performance of our SRL model. Although the dependency extractor only concentrates on syntactic information pertaining to the predicate and closely-related neighbors, more global syntactic information is nevertheless implicitly learned.

Compared with baselines, our model might be criticized for requiring gold dependency annotations for training. However, it is able to improve upon a syntax agnostic variant with sparse dependency annotations. Besides, external parsers usually require

a large number of annotated data for training. In contrast, our model requires less dependency annotations than baselines and obtain more useful syntax-aware features by extracting dependency information on its own. In the next chapter, we expand the dependency information extractor to a sentence learner, which is able to perform multiple tasks subsidiary to SRL, thereby facilitating the leverage of unlabeled data.

Chapter 5

Semi-supervised Semantic Role Labeling with Cross-view Training

In Chapter 4, we developed a supervised SRL model which uses only labeled data for training. Now, we look at semi-supervised learning for semantic role labeling. Our goal is to reduce the annotation effort involved via semi-supervised learning (SSL), as high-quality annotations are costly to obtain and not available in low-resource scenarios (e.g., rare domains). To this end, we apply *cross-view training* (CVT; Clark et al., 2018), a recently proposed SSL approach, to the semantic role labeling task.

Cross-view training is a recently proposed semi-supervised learning algorithm that improves the hidden representation learning of a Bi-LSTM sentence encoder using a mix of labeled and unlabeled data. The main idea is to teach auxiliary modules that can only see restricted views of the input to make consistent predictions with the full model seeing the complete input. Since intermediate representations are shared by auxiliary models *and* the full model, this in turn should improve the performance of the full model. Clark et al. (2018) have shown the effectiveness of CVT on sequence tagging tasks, machine translation, and dependency parsing. In order to apply CVT on the semantic role labeling task which relies on various syntactic features, we develop a sentence learner which is able to perform all tasks subsidiary to semantic role labeling (e.g., predicate identification, POS tagging, and dependency parsing). The sentence learner is trained on labeled data, and receives cross-view training on unlabeled data

to further improve itself.

We evaluate our model on different domains and multiple languages. Experimental results on the CoNLL-2009 benchmark dataset show that it is able to outperform baseline models in English (on in-domain and out-domain test data), and to improve SRL performance in other languages, including Chinese, Czech, and Spanish.

5.1 Introduction

Recent successful semantic role labelers (He et al., 2017; Marcheggiani and Titov, 2017) rely on high-quality annotations of semantic predicates and their arguments for training. However, these annotations are expensive and mostly unavailable in low-resource scenarios (e.g., rare languages or domains), which stimulates the need for effective semi-supervised approaches that leverage unlabeled data. Semi-supervised learning algorithms like self-training have been historically effective for NLP tasks (Yarowsky, 1995; McClosky et al., 2006). In self-training, a model plays the role of both a teacher who produces pseudo labels on unlabeled examples and a student trained to learn those labels. This process is somewhat tautological: the model is trained on the labels produced by itself. In order to address this problem, Clark et al. (2018) take inspiration from multi-view learning (Blum and Mitchell, 1998) and propose Cross-View Training (CVT), which encourages the model to produce consistent prediction across different *views* of the input. Compared with self-training, the students in CVT are auxiliary prediction modules instead of the full model. The input of each auxiliary prediction module is a subset of the full model's intermediate representations, which corresponds to a restricted view of the input example. The auxiliary module can learn from the full model's prediction which can see unrestricted input, and thereby improve the quality of the representations they learn. Since intermediate representations are shared between auxiliary modules and the full model, this process also improves the full model.

Clark et al. (2018) demonstrated the effectiveness of CVT on sequence tagging tasks, machine translation, and dependency parsing. Unfortunately, the application of CVT to semantic role labeling is still fraught with difficulty. One factor is the nature

of the SRL task, which has to rely on various syntactic features, even though it can be conceptualized as a sequence labeling task (He et al., 2018; Marcheggiani et al., 2017). How to extract these syntactic features in unlabeled data is problematic for semi-supervised learning. Another factor is the detection of predicates in unlabeled data (predicates are given in labeled datasets such as CoNLL2009), which has to be performed prior to argument identification and classification.

In this chapter, we simplify semi-supervised learning for semantic role labeling and propose a model that does not relying on external pre-processing tools. We develop a sentence learner which can perform all tasks subsidiary to semantic role labeling, including predicate identification, POS tagging and dependency parsing. The sentence learner is jointly trained with the semantic role labeler during supervised and semi-supervised training, and its outputs are taken as input features of the semantic role labeler all the time. Apart from multi-task features, intermediate layers in the sentence learner are also naturally provided and can be utilized as hidden-layer representations.

In addition to building a self-sufficient semantic role labeler, we show that applying CVT to SRL needs special attention over and above the sequence tagging and dependency parsing tasks discussed in Clark et al. (2018). Specifically, we investigate the impact of different predicate selection strategies on the effectiveness of CVT for SRL. Interestingly, we observe that performance dramatically varies with different selection strategies. Experimental results on the CoNLL-2009 benchmark dataset show that our model can outperform baseline models in English, and improve SRL performance in other languages, including Chinese, Czech, and Spanish.

5.2 Sentence Learner

The sentence learner (see Figure 5.1) operates over sentences to perform all tasks subsidiary to semantic role labeling (i.e., POS tagging, dependency parsing, predicate detection), which are subsequently used to inform the decisions of the semantic role labeler.

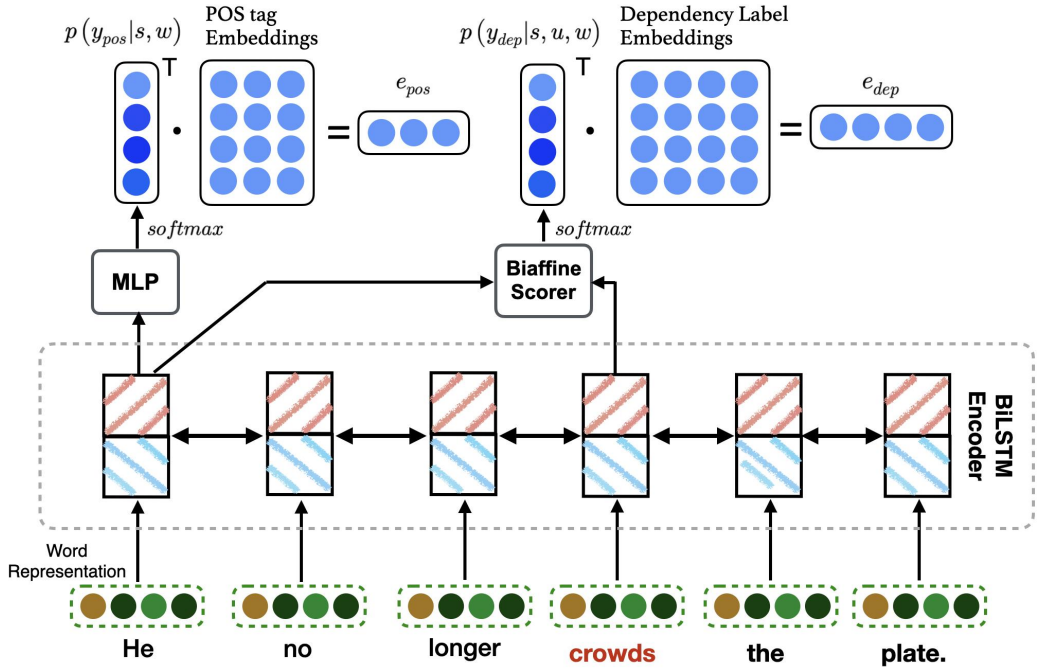

Figure 5.1: Overview of the sentence learner, which performs tasks (predicate identification, POS tagging, and dependency parsing) subsidiary to semantic role labeling.

5.2.1 Word Representations and Sentence Encoder

Since we expect to apply the sentence encoder directly on plain text without relying on external pre-processing tools, input words are represented as the concatenation of character-level and word-level features. Character-level features x^{cr} are learned by feeding character embeddings into a convolutional neural network (Chiu and Nichols, 2016). We represent words with three vectors: randomly initialized word embeddings x^{re}, pre-trained word embeddings x^{pe} estimated on an external text collection, and pre-trained ELMo embeddings x^{elmo} (Peters et al., 2018). The final word representation is given by $x = x^{re} \circ^{pe} \circ x^{cr} \circ x^{elmo}$, where $\circ$ represents concatenation operation. Following Chapter 4, we use the multi-layer BiLSTM to encode the input sentence. For the word at time step t, the BiLSTM computes a hidden representation h_t which is the concatenation of the forward and backward LSTM state vectors ($\overrightarrow{h_t}$ and $\overleftarrow{h_t}$).

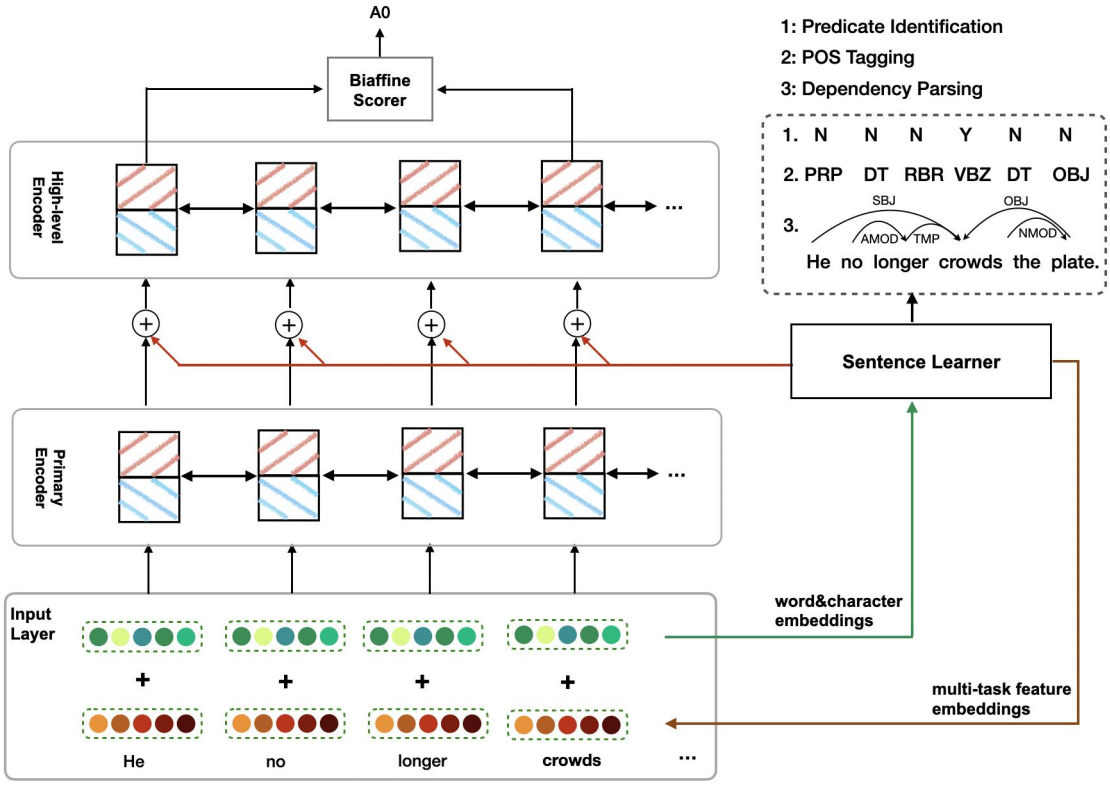

Figure 5.2: End-to-end SRL model with multi-task learning. Tasks subsidiary to semantic role labeling is performed by the sentence learner, and its outputs are fed to the high-level encoder.

5.2.2 Multi-task Learning

As shown in Figure 5.2, after obtaining word representations with the sentence encoder, the sentence learner performs POS tagging, dependency parsing, and predicate identification simultaneously. Given sentence s, the probability distribution of POS tags for word w is obtained by a multi-layer perceptron which is applied to the corresponding encoder output h_w^2 (the top-layer output of Bi-LSTM encoder):

$$p(y^{pos}|s,w) = \text{softmax}(U \cdot \text{ReLU}(Wh_w^2) + b) \tag{5.1}$$

For dependency parsing, words in a sentence are treated as nodes in a graph. In particular, each word w in sentence s receives exactly one in-going edge (u,w,r) from head word u to its dependent w with relation r. As shown in Figure 5.1, we use a graph-based dependency parser similar to the one presented in Clark et al. (2018), which treats dependency parsing as a classification task, and its goal is to predict which

in-going edge (u, w, r) connects to each word w. Mathematically, the probability of an edge is:

$$p((u,w,r)|s) = \propto \exp(\text{score}(h_u^2, h_w^2, r)) \tag{5.2}$$

where "score" is the scoring function:

$$\begin{aligned} \text{score}(z_1, z_2, r) = &\text{ReLU}(W^{head} z_1 + b^{head}) \\ &(W^r + W)\,\text{ReLU}(W^{dep} z_2 + b^{dep}) \end{aligned} \tag{5.3}$$

The bilinear classifier uses a weight matrix W^r specific to the candidate relation and a weight matrix W shared across all dependency relations.

With regard to predicate identification, we introduce a virtual root following Cai et al. (2018) who model the entire SRL task as word pair classification. Similar to the dependency parsing module described above, representations produced by the encoder for the virtual root and words are passed through separate hidden layers, and a biaffine classifier is applied to produce a score for each word.

5.3 Semantic role Labeler

As shown in Figure 5.2, the primary encoder is a single layer BiLSTM and produces hidden forward and backward LSTM state vectors $\overrightarrow{h}_t^{pri}$ and $\overleftarrow{h}_t^{pri}$ for word w_t. Since $\overrightarrow{h}_t^{pri}$ and $\overleftarrow{h}_t^{pri}$ could only "see" the left and right context of w_t respectively, these vectors are fed to auxiliary modules for cross-view training. The sentence learner performs all subsidiary tasks and produces multi-task feature embeddings. Following Chapter 4, the hidden-layer representations in the sentence learner are also collapsed into a single vector are then fed to the high-level encoder. The biaffine scorer takes hidden representations produced by the the high-level encoder as input and then calculates scores for each semantic role.

5.3.1 Word Representations and Sentence Encoder

In addition to character-level and word-level features fed to the sentence learner, our SRL model also takes multi-task features as input, which are produced by the sentence learner. Besides, following Marcheggiani and Titov (2017), we leverage a predicate-specific indicator embedding x^{ie} rather than directly using a binary flag. The final word representations for the semantic role labeler is the concatenation of four types of features: character-level, word-level, multi-task, and predicate specific features.

Concretely, multi-task features contain a composed POS tag embedding e^{pos} and a composed dependency relation embedding e^{dep}. Both types of embeddings are probability-weighted and calculated as follows:

$$\begin{aligned} e^{pos} &= \sum_{l \in tags} p(y^{pos} = l|s,w)[l] * e_l \\ e^{dep} &= \sum_{r \in rels} p(y^{dep} = r|s,u,w)[r] * e_r \end{aligned} \tag{5.4}$$

where e_l and e_r are the embeddings of POS tags and dependency relations, respectively, and $p(y^{dep} = r|s,u)$ is the probability of relation r given its predicted dependency head u. In order to incorporate more syntactic features, we adopt the probability of linking a word to the current predicate as an additional feature x^{pr}. x^{pr} consists of two scalar values: the probability of a word being the syntactic head (p^{head}) or dependent (p^{dep}) of the current predicate.

5.3.2 Multi-task Hidden Features

Inspired by ELMo (Peters et al., 2018), a recently proposed model for generating word representations based on bidirectional LSTMs and trained with a coupled language modeling objective, we extract various hidden features via multi-task learning. ELMo representations are deep, essentially a linear combination of representations learnt at all layers of an LSTM instead of just the final layer. Compared with unsupervised ELMo representations, our sentence learner takes advantage of labeled data — it attempts to learn representations towards multiple SRL-related tasks rather than generally effective ones.

Similar to the hidden-layer representation in section 4.3.2, we also utilize the combination of the hidden-layer representations in the sentence learner. In order to utilize all hidden layers in the sentence learner, we collapse them into a single vector. Although we could simply concatenate these or select the top layer, we compute multi-task hidden features h_{MT} as a weighting of the BiLSTM layers, followed by a non-linear projection:

$$h_{MT} = \text{ReLU}(W_{hidden}(\gamma \sum_{j=1}^{j=L} \beta_j h_j)) \tag{5.5}$$

where L is the depth of the sentence learner, β is softmax-normalized weights for h_j, and the scalar parameter γ is of practical importance to aid optimization (Peters et al., 2018).

5.3.3 Biaffine Role Scorer

The advantage of biaffine scorer has been discussed in Cai et al. (2018): the distribution of the role labels is uneven and the problem becomes worse after introducing *NULL* for non-argument words, and the biaffine scorer addresses such uneven problems better than more traditional scorers. After the high-level BiLSTM encoder produces representations h for each word, we perform two distinct non-linear transformations for the currently considered predicate and its candidate arguments, respectively:

$$\begin{aligned} h^{pred} &= \text{ReLU}(W^{pred} h + b^{pred}) \\ h^{arg} &= \text{ReLU}(W^{arg} h + b^{arg}) \end{aligned} \tag{5.6}$$

where h^{pred} and h^{arg} are hidden representations for the predicate and the candidate arguments. The score s^{role} of a semantic role between a predicate and its arguments is calculated as:

$$\begin{aligned} s^{role} &= h^{arg\top} W^{role} h^{pred} \\ &+ U^{role}(h^{arg} \circ h^{pred}) \\ &+ b^{role} \end{aligned} \tag{5.7}$$

where W^{role}, U^{role}, and b^{role} are parameters updated during training.

5.4 Cross-view Training for SRL

CVT (Clark et al., 2018) is a recently proposed semi-supervised learning algorithm, which trains the model to produce consistent predictions across different *views* of input. Although CVT can be applied to a variety of NLP tasks, Clark et al. (2018) focus on sequence modeling tasks (NER, POS tagging, and dependency parsing) where inputs of models are solely character and word embeddings. Unlike sequence modeling tasks, the semantic role of each argument varies with the current predicate (there might be multiple predicates in an input sentence). In Section 5.4.2, we illustrate how to adapt CVT for the SRL task.

5.4.1 Cross-view Training

CVT works by improving representation learning for a model. Let $D_{ul} = \{x_1, x_2, .., x_N\}$ represent an unlabeled dataset, and $p_\theta(y|x_i)$ denote the output distribution over classes produced by the model with parameters θ. CVT defines multiple different auxiliary prediction modules for a model, used when learning on unlabeled examples. In Clark et al. (2018), each auxiliary prediction module takes as input an intermediate representation $h^j(x_i)$ produced by a primary BiLSTM encoder and outputs a distribution over all possible classes $p^j_\theta(y|x_i)$. Each h^j is chosen so that each auxiliary prediction module can only see parts of the input.

During training, the model alternates between learning on a minibatch of labeled examples and learning on a minibatch of unlabeled examples. Given an labeled example, the model is trained in a supervised fashion (standard cross-entropy is used in Clark et al. (2018)). Given an unlabeled example, the model first produces soft targets $p_\theta(y|x_i)$ by performing inference. CVT then trains auxiliary prediction modules to match the teacher prediction module on the unlabeled data by minimizing:

$$\mathcal{L}_{\text{CVT}}(\theta) = \frac{1}{D_{ul}} \sum_{x_i \in D_{ul}} \sum_{j=1}^{k} D(p_\theta(y|x_i), p^j_\theta(y|x_i)) \tag{5.8}$$

where D is a distance function between probability distributions (we use KL divergence). During training, predictions $p_\theta(y|x_i)$ from the teacher module keep fixed so

that the auxiliary modules learn to imitate the teacher, but not vice versa. As auxiliary modules train, the representations they use as input improve, so they are useful for making predictions even when some of the models' inputs are not available. This in turn improves the primary prediction module, which is built on top of the same shared representations.

5.4.2 CVT for SRL

Like Clark et al. (2018), we used a multi-layer Bi-LSTM sentence encoder. Its first layer is used as the primary Bi-LSTM encoder, and other layers serve as the higher-level Bi-LSTM encoder. We applied CVT on the primary Bi-LSTM encoder while utilizing the output of the sentence learner on unlabeled data. Given unlabeled sentence $s = w_1, ..., w_T$, the primary Bi-LSTM encoder produces hidden representations h^{pri} for each word. The sentence learner may recognize more than one words as predicates in sentence s, and we just randomly choose one as the target predicate w_p.

The auxiliary prediction modules take $\overrightarrow{h}_t^{pri}$ and $\overleftarrow{h}_t^{pri}$ as input. Specifically, we add the following four auxiliary prediction modules to the model:

$$\begin{aligned} p_\theta^{\text{fwd}}(r_t|w_t, w_p, s) &= \text{NN}^{\text{fwd}}(\overrightarrow{h}_t^{pri}(s)) \\ p_\theta^{\text{bwd}}(r_t|w_t, w_p, s) &= \text{NN}^{\text{bwd}}(\overleftarrow{h}_t^{pri}(s)) \\ p_\theta^{\text{future}}(r_t|w_t, w_p, s) &= \text{NN}^{\text{future}}(\overrightarrow{h}_{t-1}^{pri}(s)) \\ p_\theta^{\text{past}}(r_t|w_t, w_p, s) &= \text{NN}^{\text{past}}(\overleftarrow{h}_{t+1}^{pri}(s)) \end{aligned} \tag{5.9}$$

The "forward" module makes predictions without seeing the right context of the current word. The "future" module makes predictions without seeing the right context *and* the current word itself. The "backward" and "past" modules are defined analogously for the left contexts. Figure 5.3 illustrates the auxiliary modules and the types of context they see. Unlike the biaffine role scorer, the auxiliary modules do not explicitly use the hidden state of the target predicate, so as to encourage the primary Bi-LSTM encoder to capture long-distance relations between the predicate and its context.

We empirically observed (see Section 5.5.5) that applying CVT on representations

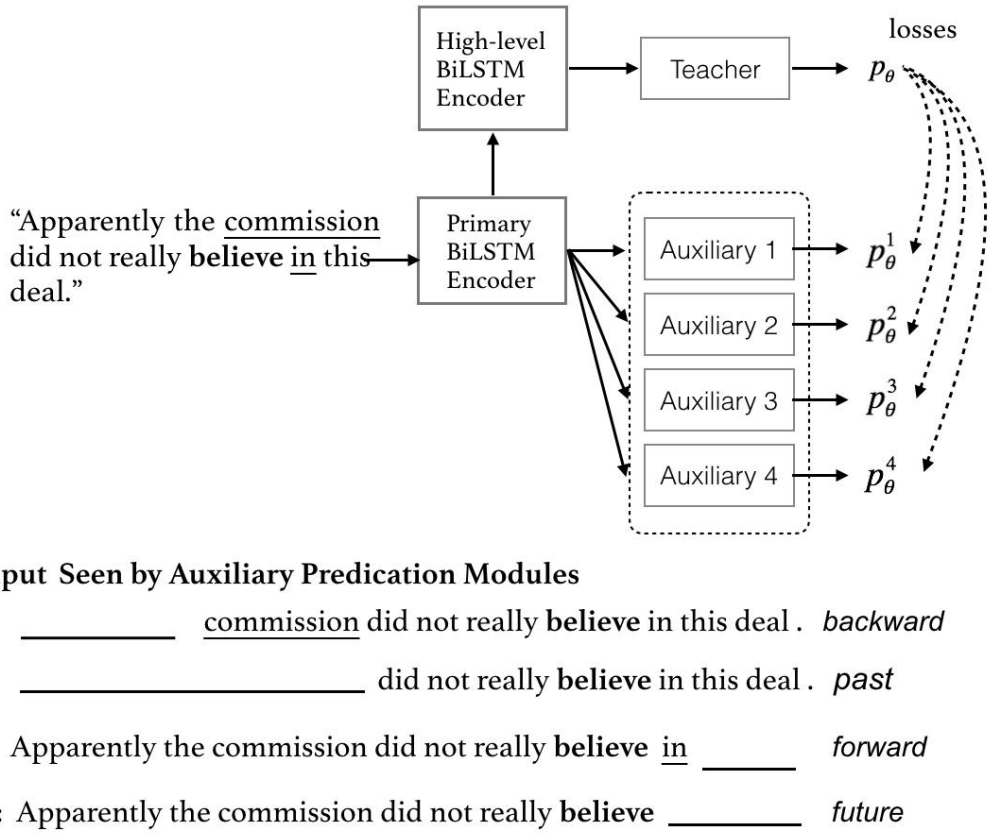

Figure 5.3: CVT on a sentence from the CoNLL 2009 training set. The current predicate is *believe*, while *commission* and *in* are two candidate arguments.

that do not "see" the target predicate is not ideal for SRL. Therefore, we devised a strategy that applies different auxiliary modules to each word depending on its relative position to the target predicate. We only apply "backward" and "past" modules to words preceding the predicate, while "forward" and "future" modules apply to words following the predicate. In this way, we ensure that each word is aware of the current predicate when performing CVT. In the example in Figure 5.3, "backward" and "past" views would be applied to "*commission*", and "forward" and "future" views to "*in*".

To further improve the performance of auxiliary tasks, we also apply CVT on the first hidden layer of the sentence learner, utilizing the views (the "forward", "backward", "future" and "past" view) introduced in Clark et al. (2018) for sequence tagging and dependency parsing.

5.4.3 Training Objective

For both supervised training and cross-view training, our model make predictions on labeled and unlabeled examples across all tasks (SRL and auxiliary tasks). For supervised training, our model is only trained on labeled data and the training objective is the sum of cross-entropy losses for all tasks. For cross-view training, our model takes unlabeled data as input and calculates the CVT loss given in Equation 5.8. The semi-supervised objective is the sum of the CVT loss for all auxiliary modules across all tasks.

5.5 Experiments

In this section we perform evaluation and analysis on CoNLL-2009 benchmark datasets across four languages (English, Chinese, Czech, Spanish). Details on implementation and training are presented in Section 5.5.2, experimental results are discussed in Section 5.5.2. Ablation experiments and analysis are presented in Section 5.5.5, which aims to evaluate the contribution of the sentence learner and the cross-view training.

5.5.1 Datasets

For supervised training and evaluation, we use the English CoNLL-2009 benchmark following the standard training, testing, and development set splits. To evaluate whether our model generalizes to other languages, we also report experiments on Chinese, Czech, and Spanish, again using standard CoNLL-2009 splits. This subset of languages has been commonly used in previous work (Björkelund et al., 2009; Marcheggiani et al., 2017) and allows us to compare our model against a wide range of alternative approaches. The benchmarks contain gold-standard dependency annotations and also gold lemmas, part-of-speech tags, and morphological features.

For cross-view training, we take advantage of a series of unlabeled datasets. Concretely, we used the 1 Billion Word Language Model Benchmark (Chelba et al., 2013)

for English, the Sougou News Data[1] for Chinese, the Czech Text Document Corpus[2] for Czech, and the Spanish Language News Corpus[3] for Spanish.

Hyperparameter	**Value**	
	Labeler	*Learner*
English word embeddings size	100	100
other languages embeddings size	300	300
character-level representations	100	100
lemma embeddings size	100	100
POS embeddings size	16	16
LSTM hidden states size	400	300
input vectors size of biaffine scorer	300	200
primary BiLSTM depth	1	1
high-level BiLSTM depth	2	1
hidden-layer representations size (v_{hidden})	200	-
dependency embeddings size (e_{link} & e_{label})	32	-
batch size	30	30
input layer dropout rate	0.3	0.3
hidden layer dropout rate	0.3	0.3
learning rate	0.001	0.001

Table 5.1: Values of hyperparameters in the semantic role labeler (*Labeler*) and the sentence learner (*Learner*)

5.5.2 Model Settings

For experiments on English, we used the embeddings of Dyer et al. (2015), which were learned using the structured skip n-gram approach of Ling et al. (2015). We also used a convolutional neural network (Chiu and Nichols, 2016; Ma and Hovy, 2016) to learn character-level representations. As for contextualized embeddings, we added ELMo embeddings (Peters et al., 2018) as input to our model. The model of ELMo is purely character-based and pretrained on the 1 Billion Word Benchmark[4], which is publicly released as part of the AllenNLP toolkit.[5] For Chinese, Spanish, and Czech word embeddings were pre-trained on Wikipedia using `fastText` (Bojanowski et al., 2017).

[1] www.sogou.com/labs/resource/ca.php
[2] http://ctdc.kiv.zcu.cz
[3] catalog.ldc.upenn.edu/LDC99T41
[4] http://www.statmt.org/lm-benchmark/
[5] https://allennlp.org/elmo

Single Models	P	R	F_1
Björkelund et al. (2010)	87.1	84.5	85.8
Lei et al. (2015)	—	—	86.6
FitzGerald et al. (2015)	—	—	86.7
Roth and Lapata (2016)	88.1	85.3	86.7
Marcheggiani and Titov (2017)	89.1	86.8	88.0
Marcheggiani et al. (2017)	88.7	86.8	87.7
He et al. (2018)	89.7	89.3	89.5
Cai et al. (2018)	89.9	89.2	89.6
Li et al. (2018)	90.3	89.3	89.8
Ours (supervised training)	91.1	90.4	**90.7**
Ours (with CVT)	91.7	90.6	**91.2**
Ensemble Models	P	R	F
FitzGerald et al. (2015)	—	—	87.7
Roth and Lapata (2016)	90.3	85.7	87.9
Marcheggiani and Titov (2017)	90.5	87.7	89.1

Table 5.2: English results on CoNLL-2009 in-domain (WSJ) test set.

The Bi-LSTM encoders in our model used recurrent dropout (Gal and Ghahramani, 2016) with an 80% keep probability between time-steps and layers during supervised training; keep probability was set to 90% when applying the model to unlabeled data. We used the Adam optimizer (Kingma and Ba, 2014) and performed hyperparameter tuning and model selection on the English development set; optimal hyperparameter values (for all languages) are shown in Table 5.1.

5.5.3 Results

Our results on the English (in-domain) test set are summarized in Table 5.2. In the first block, we compared our system against baseline models, which employ external tools to obtain required features. We also report the results of various ensemble SRL models in the second block. Most comparisons involve neural systems which are based on BiLSTMs (Cai et al., 2018; He et al., 2018; Li et al., 2018; Marcheggiani et al., 2017; Marcheggiani and Titov, 2017) or use neural networks for learning SLR-specific embeddings (FitzGerald et al., 2015; Roth and Lapata, 2016). We also report the results of two strong symbolic models based on tensor factorization (Lei et al., 2015) and a pipeline of modules that carry out tokenization, lemmatization, part-of-speech tagging, dependency parsing, and semantic role labeling (Björkelund et al., 2009). As can be

Single Models	P	R	F_1
Björkelund et al. (2009)	75.7	72.2	73.9
Lei et al. (2015)	—	—	75.6
FitzGerald et al. (2015)	—	—	75.2
Roth and Lapata (2016)	76.9	73.8	75.3
Marcheggiani and Titov (2017)	78.5	75.9	77.2
Marcheggiani et al. (2017)	79.4	76.2	77.7
He et al. (2018)	81.9	76.9	79.3
Cai et al. (2018)	79.8	78.3	79.0
Li et al. (2018)	80.6	79.0	79.8
Ours (supervised training)	82.1	81.3	**81.6**
Ours (with CVT)	83.2	81.9	**82.5**
Ensemble Models	P	R	F
FitzGerald et al. (2015)	—	—	75.5
Roth and Lapata (2016)	79.7	73.6	76.5
Marcheggiani and Titov (2017)	80.8	77.1	78.9

Table 5.3: CoNLL-2009 out-of-domain results (English; Brown test set).

seen in Table 5.2, our supervised model outperforms comparison models. With cross-view training, our model achieves 91.2% F_1 (the difference over the supervised model is statistically significant at $p < 0.05$ using stratified shuffling; Noreen, 1989), which is an absolute improvement of 0.8% over the best baseline model (Li et al., 2018).

Results on the out-of-domain English test set are presented in Table 5.3. We include comparisons with the same models as in the in-domain case. Again, our end-to-end model significantly outperforms previously published single and ensemble models, even without taking unlabeled data into account (i.e., without CVT). We achieve a relatively higher improvement with CVT on out-of-domain data (F_1 increases from 81.6% to 82.5%, and the difference is significant at $p < 0.05$ using stratified shuffling; Noreen, 1989). This suggests that semi-supervised training indeed increases the robustness of our model, leading to more accurate predictions for both SRL and auxiliary tasks.

Table 5.4 presents the results of our experiments on Chinese, Czech, and Spanish. Although we have not performed detailed parameter selection in these languages (i.e., we used the same parameters as in English), our model achieves state-of-the-art performance across all three languages.

Chinese	P	R	F_1
Björkelund et al. (2009)	82.4	75.1	78.6
Roth and Lapata (2016)	83.2	75.9	79.4
Marcheggiani and Titov (2017)	84.6	80.4	82.5
He et al. (2018)	84.2	81.5	82.8
Cai et al. (2018)	84.7	84.0	84.3
Li et al. (2018)	84.8	81.2	83.0
Ours (supervised training)	84.9	84.1	84.5
Ours (with CVT)	85.4	84.3	**85.0**
Czech	P	R	F_1
Björkelund et al. (2009)	88.1	82.9	85.4
Marcheggiani et al. (2017)	86.6	85.4	86.0
Ours (supervised training)	88.1	87.2	87.6
Ours (with CVT)	88.5	87.6	**88.0**
Spanish	P	R	F
Björkelund et al. (2009)	78.9	74.3	76.5
Roth and Lapata (2016)	83.2	77.4	80.2
Marcheggiani et al. (2017)	81.4	79.3	80.3
Ours (supervised training)	83.0	81.3	82.1
Ours (with CVT)	83.6	82.2	**82.9**

Table 5.4: CoNLL-2009 results on Chinese, Czech, and Spanish (test sets).

5.5.4 Ablation Studies

In this section, we perform a series of ablation studies to examine the contribution of the sentence learner and cross-view training. These experiments are conducted on the CoNLL-2009 English development set. Evaluations in these experiments exclude predicate disambiguation (i.e., the ablation study is conducted on gold predicates), since we want to focus on the SRL model per se.

Table 5.5 presents the results of various ablation studies against our full model (the first row). In the second block, we focus on the effect of different kinds of representations used in our model. Interestingly, we observed that the impact of ELMo (about 0.6% in F_1) is quite close to multi-task hidden features (about 0.7% in F_1). This suggests that multi-task hidden features provide as useful information for SRL as pre-trained representations. We next eliminate the sentence learner model and have the semantic role labeler use the predicted POS tags and dependency labels provided in CoNLL-2009 dataset. As can be seen, this leads to a substantial drop in performance over the full model (1.4% in F_1).

Model	P	R	F_1
Ours	88.6	86.9	87.7
w/o ELMo	87.9	86.3	87.1
w/o multi-task hidden features	88.1	86.0	87.0
w/o sentence learner	87.2	85.5	86.3
w/o CVT	88.0	85.8	86.9
w/o splitting sentence	87.4	85.7	86.6

Table 5.5: Ablation results on the CoNLL-2009 English development set.

In the third block, we only perform supervised learning on our model, and observe a 0.8% drop in F_1 over the full model. Finally, we apply the auxiliary modules directly on the full sentence instead of treating the words preceding and following the target predicate differently, and observe a 0.3% drop in F_1 over the supervised-only model. This is not surprising as the semantic roles of arguments are largely determined by the current predicate. Enforcing the primary encoder to make predictions while neglecting the current predicate could eventually hurt the performance of the full model.

5.5.5 CVT Analysis

In Table 5.6, we briefly investigate the effect of different auxiliary prediction modules. We apply two types of auxiliary modules: the "forward/backward" module does not see the right/left context of the current word; the "future/past" module does not see the right/left context and the current token itself. We found that both kinds of modules improve performance (over a supervised model without CVT, see the second row in Table 5.5); future and past modules are slightly better corroborating the results of Clark et al. (2018) on sequence tagging. Overall, the results in Table 5.6 suggest that more restricted views of the input are beneficial.

Next, we explore how the strategy of selecting the target predicate (in sentences containing multiple candidates) influences performance. For each unlabeled sentence, we adopt the strategy of randomly selecting a predicate amongst those words identified as predicate candidates by the sentence learner. An alternative is selecting the predicate with the highest predicted score or a random word from the sentence. The experiments in Table 5.7 confirm that the adopted strategy works best, delivering an improvement of 0.8 points in F_1 over a supervised model without CVT (the second row in Table 5.5).

Preceding	Following	F_1	ΔF_1
Backward	*Forward*	87.3	0.4
Past	*Future*	87.4	0.5

Table 5.6: CVT with different auxiliary modules (CoNLL-2009 English development set). Δ denotes difference from a model trained without CVT.

Strategy	F_1	ΔF_1
Randomly chosen word	87.0	0.1
Randomly chosen predicate	87.7	0.8
Most confident predicate	86.4	-0.5

Table 5.7: CVT with different predicate selection strategies for SRL (CoNLL-2009 English development set). Δ denotes difference from a model trained without CVT.

Interestingly, we observed that selecting the most confident predicate is the worst possible strategy, decreasing the performance by 0.6 F_1 points over a supervised model without CVT (see Table 5.5). The reason might be that the model only concentrates on a few predicates with very high scores (these tend to be common verbs such as *say*, *is*, and *have*), while ignoring nominal predicates and less frequent verbs. The strategy of randomly selecting a word from the sentence performs slightly better than the SRL only model, precisely because it pays attention to a wide range of predicates.

5.5.6 SOTA Dependency-based SRL Models

Instead of resorting to more powerful word embeddings (e.g., BERT (Devlin et al., 2018)) or encoders (e.g., transformer (Vaswani et al., 2017)), current SRL systems (Chen et al., 2019; Li et al., 2020; Lyu et al., 2019) commonly pay attention to high-order interactions (e.g., between arguments or arguments and predicates). Differently from previous work (Cai et al., 2018; Marcheggiani and Titov, 2017) which processes arguments independently, recent research (Chen et al., 2019; Li et al., 2020; Lyu et al., 2019) is based on the intuition that the labels of individual arguments are strongly interdependent.

Lyu et al. (2019) argue that enabling an encoder to automatically capture high-order information would require substantial amounts of data and it is hard to directly inject this information into an encoder. To this end, Lyu et al. (2019) model interactions

between argument labeling decisions through *iterative refinement*. During iterative refinement, a refinement network repeatedly takes previous output as input and produces its refined version. Naturally, such refinement strategy also requires an initial prediction, which is produced by a factorized model (base model). Similar to Lyu et al. (2019), Chen et al. (2019) propose an iterative structure-refinement algorithm. It starts with independent predictions and refines them in every subsequent iteration. Specifically, they use *Capsule Networks*: each proposition is encoded as a tuple of capsules, one capsule per argument type (i.e., role). These tuples are essentially embeddings of entire propositions, and the capsules interact with each other and with representations of words in the sentence. Each iteration results in updated proposition embeddings and updated predictions about the SRL structure. Both Lyu et al. (2019) and Chen et al. (2019) achieve better performance against non-refinement baselines on 7 languages of the CoNLL-2009 benchmark.

Instead of performing iterative refinement, Li et al. (2020) explicitly model three high-order relations: *siblings* (arguments of the same predicate), *co-parents* (predicates sharing the same argument) and *grandparents* (predicates that are the argument of another predicates). Specifically, Li et al. (2020) extend the original biaffine attention to a triaffine attention for scoring these high-order relations. Li et al. (2020) use a variational inference algorithm that allows the model to condition on high-order structures while being fully differentiable. Experimental results on 7 languages of the CoNLL-2009 benchmark show that Li et al. (2020) achieve better performance than refinement methods (Chen et al., 2019; Lyu et al., 2019) (e.g., for English, Li et al. (2020) achieve 91.77% in F_1 score, while Lyu et al. (2019) and Chen et al. (2019) achieve 90.99% and 91.06% respectively). These approaches (Chen et al., 2019; Li et al., 2020; Lyu et al., 2019) surpass our syntax-aware model in Chapter 4. With the improvement brought by CVT, our semi-supervised SRL model achieves comparable performance to refinement networks (Lyu et al., 2019) and capsule networks (Chen et al., 2019).

5.5.7 Summary

In this chapter, we described an end-to-end semantic role labeler and demonstrated it could effectively leverage unlabeled data under the cross-view training modeling paradigm. The backbone of our model is an LSTM-based semantic role labeler equipped

with a sentence learner, which performs all tasks subsidiary to SRL. The semantic role labeler and the sentence learner are trained jointly, and cross-view training is applied on the first layer of both modules.

Experiments on the CONLL-2009 benchmark datasets show that our approach outperforms baseline models in English, Chinese, Spanish and Czech. Through ablation analysis we have confirmed that ELMo embeddings, multi-task hidden features and cross-view training all contribute in improving the performance of our SRL model. Compared with the sequence tagging tasks discussed in Clark et al. (2018), we found that applying CVT to SRL requires special attention: it is important to ensure all views can "see" the target predicate and choose a wider range of predicates.

In the next chapter, we move on to cross-lingual semantic role labeling, where our model has no access to labeled data for languages of interest (target languages). We design a semantic role compressor for learning cross-lingual representations of semantic roles, which lives in a multilingual embedding space and provides direct supervision for predicting semantic roles in the target language.

Chapter 6

Alignment-free Cross-Lingual Semantic Role Labeling

Since human-labeled annotations are expensive and not available for low-resource languages, cross-lingual semantic role labeling provides an opportunity to build an SRL system for these languages while requiring annotations in a source language only. For cross-lingual semantic role labeling, word alignments play a crucial role in transferring source-language annotations into a target language. In this chapter, we present a cross-lingual SRL model which does not rely on any word-alignment tools. Concretely, our cross-lingual SRL model only requires annotations in a source language and access to raw text in the form of a parallel corpus.

The backbone of our model is an LSTM-based semantic role labeler jointly trained with a semantic role compressor and multilingual word embeddings. The compressor collects useful information from the output of the semantic role labeler, filtering noisy and conflicting evidence. It lives in a multilingual embedding space and provides direct supervision for predicting semantic roles in the target language. Results on the Universal Proposition Bank and manually annotated datasets show that our method is highly effective, even against systems utilizing supervised features.

6.1 Introduction

Although there have been considerable efforts on developing annotated resources for semantic role labeling (Palmer et al., 2005; Zaghouani et al., 2010), semantic role annotations are available for only a handful of the world's languages. To tackle this problem, much previous work has focused on *cross-lingual* SRL, which aims at leveraging existing resources in a source language to minimize the effort required to construct a model or annotations for a new target language.

A simple approach is *model transfer* which directly applies a source-language model to the target language, employing cross-lingual word representations and universal POS tags. During training, the model not only learns to embed a sentence but it also extracts language-specific features from source-languages sentences (Ahmad et al., 2019b), such as word order typology. Therefore, this method usually suffers from the inherent discrepancies (e.g., different word orders) in different languages. A popular alternative is *annotation projection* which is based on large-scale parallel corpora between source and target languages. Firstly, source-side sentences are automatically annotated with SRL tags by a semantic role labeler trained with labeled source-language data. Then the source annotations are projected to the target-side sentences based on word alignment utilizing word alignment tools (Aminian et al., 2019; Padó and Lapata, 2005). Finally, these target-side sentences with projected annotations are used to train the target-language semantic role labeler. Intuitively, it is sensitive to the accuracy of alignment tools, the quality of the parallel data, and the performance of the source-language SRL model, all of which could introduce noise. In order to alleviate the noise brought by the source-side labeler, the *translation-based approach* (Fei et al., 2020) directly translates the gold-standard data into the target language. As a result, this method is sensitive to the quality of translation tools and the accuracy of alignment tools (it still relying on alignment tools to transfer word-level labels from the source language to the target language).

As observed in previous work (Aminian et al., 2019; Fei et al., 2020), word alignment noise poses problems for annotation-projection and translation-based methods. For example, there could be one-to-many or many-to-many alignments, leading to semantic role conflicts in the target language (Fei et al., 2020). To alleviate alignment

noise, some form of filtering is inevitably introduced: parallel sentence pairs are discarded according to projection density (Aminian et al., 2019) or alignment confidence (Fei et al., 2020). Therefore, in this respect, model transfer is an appealing alternative. However, it relies on perfectly aligned word embeddings (which are usually not available) and accurate features based on lemmas, POS tags, and syntactic parse trees (Fei et al., 2020; Kozhevnikov and Titov, 2013), which are obtained with access to additional annotation. However, it is not realistic to assume that treebank-style resources will be available for low-resource languages.

In this chapter, we propose a novel method for cross-lingual SRL which does not rely on word alignments, machine translation or pre-processing tools such as parsers or POS taggers. Aside from semantic role annotations in the source language, we only assume access to raw text in the form of a (source-target) parallel corpus. The main idea of our approach is to encode a proposition into a fixed-size and cross-lingual representation. Having such proposition representations let us easily perform the reconstruction of the SRL prediction within and across languages. Given a parallel sentence pair, the representation of the source-language proposition could provide supervision for the target-language semantic role labeling. To this end, we define a semantic role compressor, which is jointly trained with an LSTM-based semantic role labeler. The compressor distills useful information pertaining to arguments and their roles from the output of the semantic role labeler (e.g., by automatically filtering unrelated or conflicting information). Importantly, the compressor lives in a multilingual space and can provide direct supervision for predicting semantic roles in the target language, sidestepping intermediaries like word-level alignments and machine translation.

For evaluation, we make use of several multilingual SRL benchmarks. These benchmarks include the Universal Proposition Bank (UPB; Akbik et al. 2016), a recently released resource which contains semi-automatically created annotations under a unified labeling scheme across several languages, and a French corpus (Van der Plas et al., 2010) which follows PropBank-style annotations (Palmer et al., 2005). We also construct two additional manually labeled resources in Chinese and German, following the annotation guidelines of UPB (see Section 6.5.1 for more details). Experimental results show that our method is highly effective across languages and annotation schemes, even compared against systems making use of supervised features.

6.2 Semantic Role Labeler

Figure 6.1 provides an overview of our model, which consists of the semantic role labeler (lower part in Figure 6.1) and the semantic role compressor (upper part in Figure 6.1). The semantic role labeler used in this chapter is similar to labelers in Chapter 4 and 5, since they all encode sentences with Bi-LSTM and compute scores of roles with a biaffine-scorer. Different from labelers in previous chapters, the output of semantic role labeler used in this chapter is fed to the compressor as input, and its parameters are updated not only during supervised training, but also cross-lingual training.

6.2.1 Input Layer and Encoder

For each input sentence, the representation of i-th word w_i is the concatenation of multilingual contextualized word embeddings $e^w_{w_i}$ and predicate indicator embedding $e^p_{w_i}$. The former are pretrained on a large-scale unlabeled corpus containing data in multiple languages, and their parameters stay *frozen* during the training of our model. Predicate embeddings are randomly initialized and updated constantly during model training. Different from previous works (Cai et al., 2018; Li et al., 2019b), our model avoids using any syntactic information (e.g., POS embeddings and dependency relations), since we cannot assume it is available for low-resource languages. Following Chapter 4, we use the multi-layer BiLSTM to encode the input sentence. For the word at time step t, the BiLSTM computes a hidden representation h_t which is the concatenation of the forward and backward LSTM state vectors ($\overrightarrow{h_t}$ and $\overleftarrow{h_t}$).

6.2.2 Biaffine Role Scorer

Following supervised SRL models (Cai et al., 2018; He et al., 2019), once the BiLSTM encoder produces representations h for each word, two distinct non-linear transformations are applied to predicate w_p (being considered at the time) and word w_i,

respectively:

$$\begin{aligned} h_{w_p}^{pred} &= f(W_p h_{w_p} + b_p) \\ h_{w_i}^{arg} &= f(W_w h_{w_i} + b_w) \end{aligned} \tag{6.1}$$

where f is a non-linear activation function (we use Leaky ReLu). The score $s(r_j, h_{w_i}^{arg}, h_{w_p}^{pred})$ of semantic role r_j between current predicate w_p and word w_i is calculated as:

$$\begin{aligned} s(r_j, h_{w_i}^{arg}, h_{w_p}^{pred}) = {} & h_{w_i}^{arg\top} W_{r_j} h_{w_p}^{pred} \\ & + U_{r_j}(h_{w_i}^{arg} \circ h_{w_p}^{pred}) + b_{r_j} \end{aligned} \tag{6.2}$$

where W_{r_j}, U_{r_j}, and b_{r_j} are parameters specific to role r_j, and are updated during training.

6.2.3 Predicate Identification and Disambiguation

For an end-to-end SRL system, predicate identification is an indispensable step before predicate disambiguation and argument identification. Since predicates in most SRL datasets (e.g., CoNLL-2009) are explicitly annotated, the SRL labeler presented thus far assumes that predicates are known. However, such annotations are absent from unlabeled parallel data, and our model would need to automatically identify predicates if it were to be useful in practice. To this end, we run two modules on top of the sentence encoder in order to identify the predicate and disambiguate its senses. Each module is a multi-layer perceptron (MLP) with a softmax layer and is trained jointly with the semantic role labeler.

When performing predicate identification on unlabeled parallel data, we need to find aligned predicate pairs. Given an unlabeled parallel source-target sentence pair (S^S and S^T), we first perform predicate identification on both sentences and then randomly choose a predicate w_p^S in S^S as the current predicate of interest. We then find, amongst all words identified as predicates in S^T, predicate w_p^T which has the highest word embedding similarity with w_p^S. Then, we check whether the opposite (amongst all words identified as predicates in S^S, w_p^S has the highest word embedding similarity with w_T^S) is also true. If the opposite is true, we use w_p^S and w_p^T as the predicates in source and target sentences, respectively. Otherwise, we randomly choose a predicate other than w_p^S and check it again, until we find an aligned predicate pair which passes

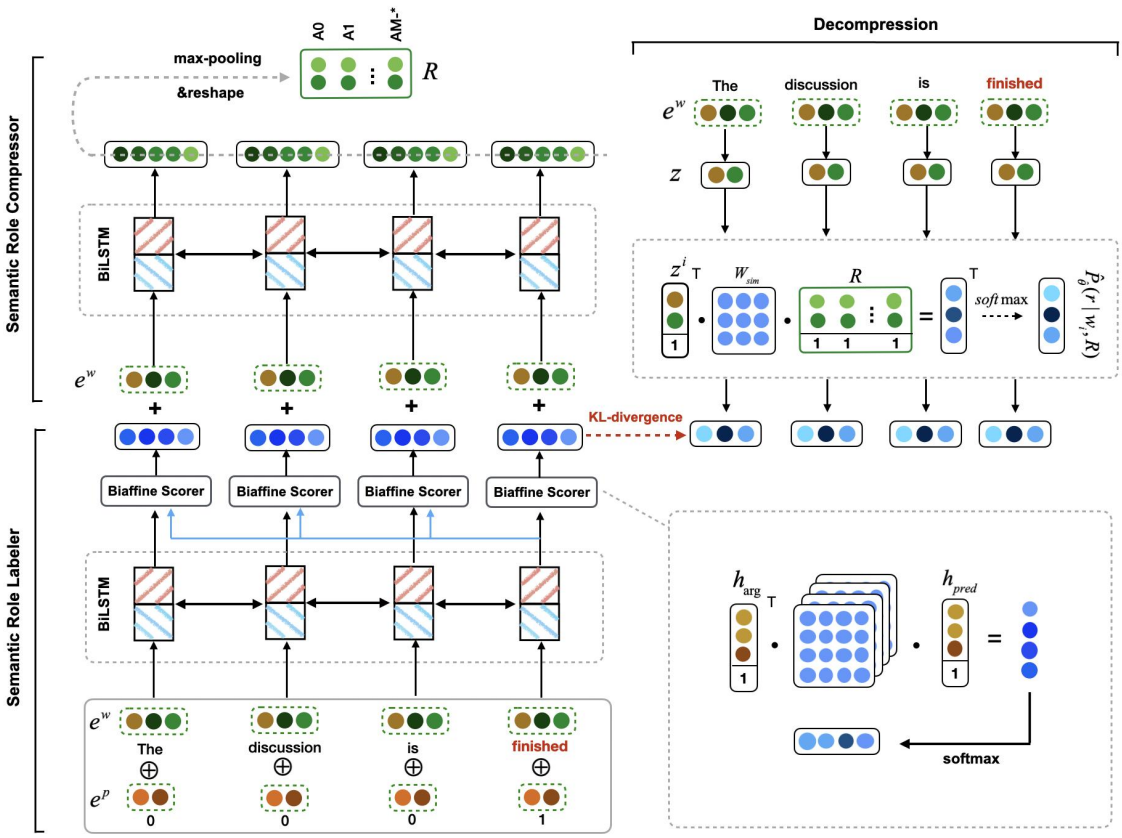

Figure 6.1: Model overview: semantic role labeler (left-bottom) and semantic role compressor (left-top). The right-top part presents the process of decompression after obtaining R.

the test.

6.3 Semantic Role Compressor

We elaborate now on how our model works by splitting the cross-lingual transferring into two sub-steps: *compression* and *decompression*. The key component of our model is the semantic role compressor (the upper part in Figure 6.1) which is able to produce compressed role information (*compression*) and reconstruct SRL predictions (*decompression*). For compression, the semantic role compressor operates over the output of the semantic role labeler. After obtaining the compressed role information produced by the compressor, we can perform decompression by combining the compressed role information with multilingual embeddings. During cross-lingual training, given a parallel sentence pair, both sentences are fed to the semantic role labeler and then the semantic role compressor to obtain source-side and target-side compressed role infor-

mation, respectively. After compression, both source-side and target-side compressed role information take part in decompression for cross-lingual transfer. With the help of semantic role compressor, our model could provide direct supervision for predicting semantic roles in the target language without relying on word alignment tools.

6.3.1 Semantic Role Compression

Although the semantic role labeler assigns a label to every word in the input sentence, most words will bear the label "NULL", indicating that they are not arguments of the predicate of interest. In theory, we expect each semantic role could appear no more than once in a sentence. This constraint of semantic roles largely facilitates compression: we can compress the output of the semantic role labeler into a fixed-size matrix which only records information about arguments. Concretely, we use a fixed-size matrix $R \in \mathbb{R}^{n_r \times d_r}$ to represent compressed information, where n_r is the size of the semantic role set, and d_r denotes the length of the hidden representation for each semantic role.

In order to compress the output of the semantic role labeler, the semantic role compressor operates over word embeddings (see the upper left part in Figure 6.1); for sentence S, word w_i is represented by $P_\theta(r|w_i, w_p, S) \circ e^w_{w_i}$, where $e^w_{w_i}$ is the multilingual embedding of w_i, and $P_\theta(r|w_i, w_p, S)$ is the probability distribution over roles produced by the semantic role labeler:

$$\begin{aligned} P_\theta(r|w_i, w_p, S) = \text{softmax}\{&s(r_1, h'_{w_i}, h'_{w_p}), \\ &\ldots, s(r_{n_r}, h'_{w_i}, h'_{w_p})\} \end{aligned} \tag{6.3}$$

where θ are the parameters of the semantic role labeler. Analogously to the semantic role labeler, a multi-layer BiLSTM yields sentence representations (see the upper block in Figure 6.1). At time step t, forward and backward hidden states $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ are concatenated and then fed to a non-linear layer. A max-pooling layer thereafter gathers global information from hidden features at each time step and compresses them into a fixed-size vector:

$$R = \max_{t=1}^{n} f(W_1[\overrightarrow{h}_t \circ \overleftarrow{h}_t] + b_1) \tag{6.4}$$

where W_1 is a weight matrix, b_1 is a bias term for the hidden state vector, and n is the

length of the sentence. For the sake of decompression (see next section), R is reshaped from a vector into a matrix with n_r rows and d_r columns, where each row corresponds to a unique semantic role (see very top in Figure 6.1, left side).

6.3.2 Decompression

During compression, we try to recover distribution $P_\theta(r|w_i, w_p, S)$ for w_i, based on compressed information R produced by the compressor. Concretely, for word w_i and role j, we use a biaffine scorer[1] to calculate the similarity between $e^w_{w_i}$ and R_j (the j-th row of R). We first perform a non-linear transformation f for word embedding $e^w_{w_i}$:

$$z_i = f(W_2 e^w_{w_i} + b_2) \tag{6.5}$$

where z_i contains hidden features for word w_i. And then, use a biaffine scorer to calculate the similarity score between z_i and R_j:

$$\begin{aligned} \hat{s}(z_i, R_j) = z_i \top W_{sim} R_j \\ + U_{sim}(z_i \circ R_j) + b_{sim} \end{aligned} \tag{6.6}$$

where W_{sim}, U_{sim}, and b_{sim} are parameters updated during training. For word w_i, the final probability distribution over semantic roles is obtained by applying a softmax operation on the scores of all semantic roles:

$$\begin{aligned} \hat{P}_{\hat{\theta}}(r|w_i, R) = \text{softmax}\{\hat{s}(z_i, R_1), \\ \ldots, \hat{s}(z_i, R_{n_r})\} \end{aligned} \tag{6.7}$$

where $\hat{\theta}$ are the parameters of the compressor. Figure 6.1 (right upper part) illustrates decompression.

[1] The score for the label "NULL" is fixed to 0, as R does not record information for non-argument words.

6.3.3 Gaussian Noise

In order to improve the robustness of the semantic role compressor, we inject Gaussian noise to multilingual word embeddings. This is an effective regularization method (Liu et al., 2019) which improves the model's ability to generalize to unseen inputs from different languages. The final embeddings are: $e^w = [e^w_{w_1} + N_1, ..., e^w_{w_n} + N_n]$, where $\mathbf{N} \sim \mathcal{N}(0, 0.1\mathbf{I})$ ($\mathbf{I}$ is a vector with the same dimension of e^w and its elements are all 1) and n is the length of the sentence.

6.4 Training

In our learning setting, semantic role annotations are only available for the source language. Therefore, we rely on (unlabeled parallel) sentence pairs to transfer cross-lingual supervision for the target language. Generally, the model alternates training on batches of annotated source-language data and unlabeled parallel data. When training with annotated source-language data, we can easily perform supervised training to improve the performance of the source-language labeler. When it comes to unlabeled parallel data, we propose a new approach to transfer cross-lingual supervision for the target language, and we refer to this process as cross-lingual training.

6.4.1 Mono-lingual Training

With labeled data in the source language, we train the semantic role labeler in a supervised fashion (only parameters of the the semantic role labeler are updated), using a cross-entropy loss objective:

$$\mathcal{L}_{\text{ce}} = \frac{1}{n} \sum_{i=1}^{n} t_i \log P_\theta(r | w_i, w_p, S) \tag{6.8}$$

where n is the length of the sentence and $t_i \in \mathbb{R}^{n_r}$ are one-hot ground truth representations. When training the compressor network (only parameters of the semantic role compressor are updated), the objective is defined as the KL-divergence between the

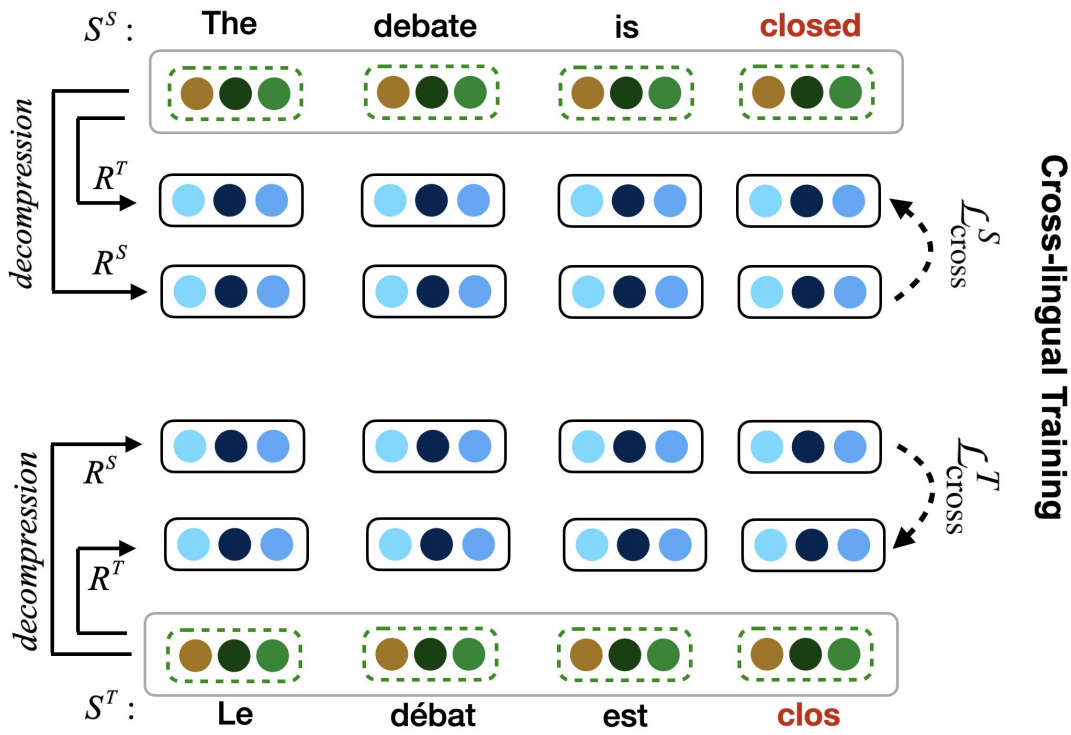

Figure 6.2: Cross-lingual training given an English-French sentence pair (S^S and S^T) from the Europarl Prallel Corpus, where R^S and R^T are the output of the compressor taking S^S and S^T as input, respectively.

input distribution (produced by semantic role labeler) and the output distribution of the decompressor:

$$\mathcal{L}_{\text{com}} = \frac{1}{n}\sum_{i=1}^{n} D(P_\theta(r|w_i, w_p, S), \hat{P}_{\hat{\theta}}(r|w_i, R)) \tag{6.9}$$

where D is a distance function between probability distributions (we use the Kullback-Leibler divergence). The final objective $\mathcal{L}_{\text{mono}}$ for mono-lingual learning is the sum of $\mathcal{L}_{\text{ce}}$ and $\mathcal{L}_{\text{com}}$.

6.4.2 Cross-lingual Training

Given an unlabeled source-target sentence pair S^S and S^T, the first step is to identify predicates and then find aligned predicate pairs (discussed in Section 6.2.3). For a predicate pair w_p^S and w_p^T, we obtain compressed role representations R^S and R^T for source and target sentences S^S and S^T, by feeding word embeddings and predicate information into our model. In order to obtain role-specific information for S^S and S^T, we must apply decompression. Since decompression operates over multilingual

representations, it is relatively straightforward to obtain semantic roles for source and target sentences. In fact, we apply R^S and R^T on both S^S and S^T and compare the outcome (see Figure 6.2). The training objectives are defined as:

$$\mathcal{L}^S_{\text{cross}} = \frac{1}{n_S}\sum_{i=1}^{n} D(\hat{P}_{\hat{\theta}}(r|w_i^S, R^S), \hat{P}_{\hat{\theta}}(r|w_i^S, R^T)) \tag{6.10}$$

$$\mathcal{L}^T_{\text{cross}} = \frac{1}{n_T}\sum_{i=1}^{n} D(\hat{P}_{\hat{\theta}}(r|w_i^T, R^S), \hat{P}_{\hat{\theta}}(r|w_i^T, R^T)) \tag{6.11}$$

where n_S and n_T are the length of S^S and S^T, respectively. $\mathcal{L}^S_{\text{cross}}$ and $\mathcal{L}^T_{\text{cross}}$ are used for updating parameters of the semantic role labeler and compressor.

In order to improve the performance of the semantic role compressor on the source and target language, we train it using parallel sentence pairs by minimizing:

$$\mathcal{L}^S_{\text{com}} = \frac{1}{n_S}\sum_{i=1}^{n} D(P_{\theta}(r|w_i^S, w_p^S, S^S), \hat{P}_{\hat{\theta}}(r|w_i^S, R^S)) \tag{6.12}$$

$$\mathcal{L}^T_{\text{com}} = \frac{1}{n_T}\sum_{i=1}^{n} D(P_{\theta}(r|w_i^T, w_p^T, S^T), \hat{P}_{\hat{\theta}}(r|w_i^T, R^T)) \tag{6.13}$$

$\mathcal{L}^S_{\text{com}}$ and $\mathcal{L}^T_{\text{com}}$ are only used for updating parameters of the semantic role compressor. The final training loss during cross-lingual training $\mathcal{L}_{\text{cross}}$ is the sum of above losses:

$$\mathcal{L}_{\text{cross}} = \mathcal{L}^S_{\text{cross}} + \mathcal{L}^T_{\text{cross}} + \mathcal{L}^S_{\text{com}} + \mathcal{L}^T_{\text{com}} \tag{6.14}$$

6.5 Experiments

In this section, we present our experimental evaluation and analysis on a series of multilingual SRL benchmarks. We first introduce these benchmarks and details on implementation and training. Our results are discussed in Section 6.5.3. Ablation experiments and analysis are presented in Section 6.5.5, which tries to evaluate the contribution of various model components.

PropBank v3	UPB					
EN	DE	FR	IT	ES	PT	FI
272,380	997	298	489	1,995	936	716

CoNLL-09	Van der Plas et al. (2010)
EN	FR
39,279	1,000

ProBank v3	UPB (manually re-labeled)	
EN	ZH	DE
272,380	304	258

Table 6.1: Annotated data used in our experiments. We show the English source annotations (left column) used for *training* and corresponding target annotations used for *testing* in various languages.

6.5.1 Datasets

We trained our model using English as the source language and obtained semantic role labelers in German (DE), Spanish (ES), Finish (FI), French (FR), Italian (IT), Portuguese (PT), and Chinese (ZH). For the source language (English), we used labeled datasets including the Propositional Bank (v3; Palmer et al., 2005) and the CoNLL-09 shared task (Hajič et al., 2009b). During cross-lingual training, we used the Europarl parallel corpus (Koehn, 2005) for the European languages (Table 6.2 gives the size of the various parallel corpora used in our experiments); For Chinese, we used a large-scale EN-ZH parallel corpus (Xu, 2019), which contains about 5 million sentence pairs.

For evaluation, we used the Universal Proposition Bank (UPB, v1.0; Akbik et al., 2016), which is built upon the Universal Dependency Treebank (UDT, v1.4) and the Proposition Bank (PB, v3.0). Since all languages in the UPB follow a unified dependency-based SRL annotation scheme, we converted argument spans in the English Proposition Bank to dependencies by labeling the syntactic head of each span.

UPB is the first "universal proposition banks" for multilingual SRL with homogeneous label style, and it is semi-automatically created via annotation projection. As a result, it unavoidably contains a certain amount of errors. Therefore, we also tested our model on manually annotated datasets. Presumably due to the labeling effort in-

Language	size
German	1,920,209
Spanish	1,965,734
Finnish	1,924,942
Italian	1,909,115
Portuguese	1,960,407
French	2,007,723

Table 6.2: Number of sentence pairs in Europarl for six languages.

volved, manually annotated datasets are still few and far between, but they tend to be more accurate than automatically created datasets. For French, we used a manually annotated dataset (Van der Plas et al., 2010) which provides SRL labels following an annotation scheme similar to CoNLL-09 for English (Hajič et al., 2009b). It is worth noting that although the CoNLL-09 shared task provides dependency-based semantic role annotations for seven languages, the role sets differ across languages, and unifying them is far from trivial. To this end, we created two manual datasets (available at https://github.com/RuiCaiNLP/ZH_DE_Datasets), by randomly sampling 258 German and 304 Chinese sentences from UPB. The manual annotation was performed by native speakers following the annotation guidelines of UPB, which in turn follows the English Proposition Bank. Table 6.1 provides a breakdown of labeled data used in our experiments.

6.5.2 Model Configuration

We implemented our model in Pytorch and optimized it using the Adam optimizer (Kingma and Ba, 2014). For the word embeddings, we used the officially released multilingual BERT (base; cased version; Devlin et al. 2018). The parameters of BERT were kept fixed during training in order to preserve the cross-lingual nature of the embeddings. The embedding of the predicate indicator was randomly initialized and updated during training. Optimal hyperparameter values (for all languages) are shown in Table 6.3.

Hyperparameter	Value	
	Labeler	*Compressor*
multilingual BERT embeddings size	768	768
predicate indicator embeddings size	16	16
BiLSTM hidden states size	400	256
BiLSTM depth	3	2
batch size	30	30
input layer dropout rate	0.3	0.3
hidden layer dropout rate	0.3	0.3
learning rate	0.001	0.001
compressed role representation size	-	30

Table 6.3: Values of hyperparameters in the semantic role labeler (*Labeler*) and the sentence learner (*Learner*)

6.5.3 Results on Universal Proposition Bank

We compared our model against several baselines on the UPB test set. These baselines include a cross-lingual model *Bootstrap* (Aminian et al., 2017) and a character-based approach *CModel* (Aminian et al., 2019), which both perform annotation projection through parallel data and then filter word alignments empirically. We also report the results of several models trained with a pseudo translated target corpus and a source language corpus. Similar to *Bootstrap* and *CModel*, filtering is also performed to improve the quality of the pseudo target corpus. We compared against two strong mixture-of-experts models which focus on combining language specific features automatically (*MOE*; Guo et al. 2018), and also on learning language-invariant features with a multinomial adversarial network as a shared feature extractor (*MAN-MOE*; Chen et al. 2019). We also include a recently proposed translation-based model (*PGN*; Fei et al. 2020), using the same training data with *MOE* and *MAN-MOE*; this system adopts a parameter generation network (PGN) to enhance the BiLSTM module and capture language differences.

Our results on the Universal Proposition Bank test set are summarized in Table 6.4. We report experiments on six languages, and these languages are ordered according to their typological distance to English based on the word order (Ahmad et al., 2019b) with Portuguese being closest and Finnish farthest. As shown in Table 6.4, our model outperforms previous systems on Portuguese (PT), French (FR) and German (DE), and is on average better (see column "avg" in Table 6.4). It is worth noting that, apart from

Models	PT	FR	ES	IT	DE	FI	avg
Dist. to EN	0.09	0.09	0.12	0.12	0.14	0.20	0.13
Bootstrap	53.9	63.4	52.2	52.3	55.0	53.1	55.0
CModel	56.5	58.5	56.0	55.5	57.0	**58.9**	57.1
MAN-MOE	55.2	65.3	**62.8**	57.1	64.3	52.3	59.4
MoE	55.5	63.3	60.3	56.7	63.2	50.6	58.2
PGN	56.0	64.8	62.5	**58.7**	65.0	54.5	60.3
Ours	**57.8**	**66.2**	61.5	57.6	**65.7**	57.6	**61.1**

Table 6.4: Results (F_1) on UPB test sets for six languages. Results for comparison systems are taken from previous papers (Aminian et al., 2019; Fei et al., 2020).

pretrained word-alignment tools, both *Bootstrap* and *PGN* utilize supervised part-of-speech (POS) tags for the source and target language. Without using any additional features, our model still achieves the better average F-score (61.1%) than baselines.

6.5.4 Results on Human-labeled Data

As UPB datasets are semi-automatically created and therefore possibly contain a certain amount of projection errors, we further performed testing using manual annotations on French, German, and Chinese. Since previous work has not provided results on these new datasets, we re-implemented three strong comparison systems, i.e., *CModel*, *MAN-MOE*, and *PGN*. When implementing translation-based models, we used Google Translate[2] as our translation engine. In order to obtain word alignment, we used a popular tool giza++[3]. When prepossessing the Chinese part in the EN-ZH parallel corpus (containing about 5 million sentence pairs), we used Jieba[4] for tokenization. The Chinese testset in UPB is in traditional Chinese, and we used Zhtools[5] to convert it to simplified Chinese to be compatible with our EN-ZH parallel corpus which is also in simplified Chinese.

Our results are summarized in Table 6.5, where again we order languages in terms of their word order distances to English (Ahmad et al., 2019b). We note that our approach significantly outperforms previously published models on these three lan-

[2] ttps://translate.google.com/
[3] https://github.com/moses-smt/giza-pp
[4] https://github.com/fxsjy/jieba
[5] https://github.com/skydark/nstools/tree/master/zhtools

Models	FR	DE	ZH	avg
CModel	68.5	66.9	62.3	65.9
MAN-MOE	72.8	69.2	64.7	68.9
PGN	73.2	70.1	65.4	69.5
Ours	**75.3**	**71.4**	**68.5**	**71.7**

Table 6.5: Results (F_1) on manually annotated test sets for German, French, and Chinese. Pairwise differences between our model and previous systems are all statistically significant ($p < 0.05$) using stratified shuffling (Noreen, 1989).

Models	DE	FR	ZH
Ours	63.4	68.8	60.4
w/o BERT	47.7	52.6	44.5
w/o BERT (+position)	55.3	60.5	53.0
w/o Gaussian noise	61.7	66.2	57.7
w/o cross-lingual training	52.5	59.8	49.5
w/o compressor (+attention)	51.7	59.5	47.1

Table 6.6: Ablations on manually annotated datasets.

guages. Pairwise differences in F_1 between our model *MAN-MOE*, *CModel*, and *PGN*) are all statistically significant ($p < 0.05$) using stratified shuffling (Noreen, 1989). It is perhaps not surprising to see that all systems perform best on French and worst on Chinese, since French is closest to English and Chinese is least related to English. This suggests that it is easier to transfer SRL annotations between languages with similar word orders.

6.5.5 Ablation Studies and Analysis

In this section, we perform a series of ablation studies to examine the contribution of the semantic role compressor and cross-lingual training. These ablation studies are conducted on the manually annotated DE, FR, and ZH datasets. Our ablation studies are conducted without predicate disambiguation since we want to focus on the SRL model per se.

Table 6.6 presents the results of various ablations against our full model (first row). In the second block, we examine the effect of different kinds of representations used in our model. First, we substitute contextualized word embeddings (BERT) with non-contextualized ones (MUSE, Lample et al. 2018), which were obtained by aligning

(monolingual) fastText embeddings for various languages onto a universal space. We can see that the performance of our model using MUSE embeddings drops significantly. This is largely due to the phenomenon that some words appear multiple times in the same sentence. Since MUSE embeddings are not contextualized, they are unable to distinguish words appearing at different positions in the same sentence, which in turn leads to severe conflicts during decompression. To solve this problem, we concatenate MUSE with word position embeddings during compression and decompression. Different from standard transformers where positional features are bound to word indices, the positional features we used for word w_i only record the number of words that are the same as w_i and appeared before w_i.

We illustrate the positional features with an example (shown in Figure 6.3). There are two reasons why we adopt these new positional features. Firstly, during cross-lingual training, the length of parallel sentences S^S and S^T is usually different. Secondly, for i-th word in S^S, its corresponding words $w^{'}_j$ in S^T is not the i-th word in S^T in most cases. However, in our model it is important that w_i and $w^{'}_j$ have the same position embeddings, so that they can obtain a similar result after decompression. Figure 6.3 shows a pair of English-French parallel sentences with our positional embeddings. Although *he* (*il* in French) appears twice in the English sentence, its French counterpart still shares the same positional features. Experimental results also show that positional features can effectively improve cross-lingual training. However, there are still some cases where our positional features do not work. For example, when the word order changes dramatically after translation, counting the number of identical words cannot help us find aligned source-target word pairs. The only perfect solution seems to be using contextualized multilingual embeddings like multilingual BERT or multilingual ELMO, as every word in a sentence will receive unique word embeddings and aligned word pairs tend to share similar embeddings.

Adopting positional features improves SRL performance from 47.7% (DE), 52.6% (FR), and 44.5% (ZH) to 55.3%, 60.5% and 53.0%, but it is still inferior to the original model. Next, we remove Gaussian noise from the model, and as can be seen there is a drop in performance, indicating that it further boosts SRL accuracy. In the third block, we remove cross-lingual training (only perform supervised training) and observe a significant drop in F-score over the full model.

Figure 6.3: Positional features for English-French parallel sentences.

French	*SRL only*	Ours	Frequency(%)
A0	71.9	83.6	26%
A1	65.7	78.8	37%
A2	37.8	43.6	7%
AM-*	46.7	48.5	30%

Chinese	*SRL only*	Ours	Frequency(%)
A0	59.2	63.7	18%
A1	59.9	74.4	38%
A2	38.6	65.6	15%
AM-*	36.0	37.3	29%

Table 6.7: Results (F_1) on French and Chinese test sets grouped by gold role labels.

In order to verify the necessity of performing semantic role compression, we substitute the compressor with an attention-based module (Bahdanau et al., 2015) and perform supervised training and cross-lingual training as described in Section 6.4. Specifically, we obtain soft alignments and use these to weight all annotations $P_\theta(r|w_i, w_p, S)$, thereby obtaining a normalized expectation over role assignments. During cross-lingual training, the alignment module and the basic semantic role labeler are trained jointly. We observe that compared with the full model, the performance drops substantially for all three languages when using the attention-based module. The reason might be that the output of the semantic role labeler is noisy and attention can not resolve labeling conflicts effectively (e.g., when two words show high confidence for the same semantic role).

We further explore our model's performance for French and Chinese for different (gold) role labels. We compare the full model against an *SRL only* model which performs supervised training for the source language (English). As shown in Table 6.7, cross-lingual training improves SRL performance in French and Chinese on all semantic roles. For French, the most significant improvement comes from A1; for Chinese,

cross-lingual training benefits labeling A1 and A2 significantly. For both French and Chinese, the improvements on AM-* (modifiers for current predicate) are modest compared with A0, A1, and A2. One possible reason is that the head words of A0, A1, and A2 are usually nouns or adjectives, which tend to have fixed positions in parallel sentence pairs. However, modifiers can be optional and have more varied positions within and across languages, which increases the difficulty for cross-lingual learning.

6.6 Summary

In this chapter, we have presented a cross-lingual semantic role labeler and demonstrated it can effectively leverage unlabeled parallel data without access to word alignments or any other external tools. Ablation studies have shown that multilingual contextualized word embeddings are indispensable to our model, as they can distinguish words with multiple occurrences in a sentence. Although our model achieves better performance than baselines, it is also influenced by language distance. This indicates that transfer between languages with similar word order is an easier task. Since semi-automatically created datasets contain a certain amount of errors, we have also contributed two quality controlled datasets (following the annotation scheme of PropBank), which we hope will be useful for the future research on cross-lingual SRL models.

Although our model focuses on dependency-based SRL, it can also be adapted to span-based SRL (Carreras and Màrquez, 2005; Pradhan et al., 2013). To this end, the semantic role compressor would need to be modified to record the information of the whole span rather than just the head word. In this case, the function of decompression remains unchanged, it would still output a probability distribution over all semantic roles for each word in the input sentence.

Chapter 7

Conclusions and Future Directions

The present chapter offers a summary of the principal contributions and findings of this study. It also outlines potential avenues for future research.

7.1 Conclusions

This thesis has been concerned with the task of semantic role labeling. Specifically, we have investigated semantic role labeling in various learning settings, from monolingual semantic role labeling where we assume semantically-annotated data are always available for the languages of interest, to cross-lingual SRL where SRL annotations are only available for the source language. Three main research questions motivated our work:

1. *Assuming a resource-rich setting, can we employ existing additional resources (such as treebanks) to enhance SRL performance, without resort to full-blown parsing?*

Neural-network models have driven state-of-the-art development regarding SRL. These are typically LSTMs without access to any linguistic information beyond predicate-role annotations. In this thesis, we argue that syntactic information is useful in terms

of SRL and can improve its performance. Furthermore, practitioners have access to resources such as treebanks (for instance, with dependency annotations), and these may be deployed to improve SRL performance.

In the present thesis, we proposed a syntax-aware SRL model, which can learn syntax-aware representations autonomously. It was found that hidden-layer representations, rather than the dependency-information extractor output, was the principal source of improvement. Since syntactic analysis would become noisy on low-resource domains, our model is, in this regard, more robust than other syntax-aware SRL systems. We also investigated how the performance of our model varied when different proportions of dependency annotation were used for training. In general, our model can also work effectively when a small number of gold dependency labels are given as a supervision signal to the dependency-information extractor.

2. *Can we enhance SRL performance with unannotated data?*

A semi-supervised SRL model was developed, and it was demonstrated that, within the paradigm of cross-view training modeling, this model can successfully leverage unlabeled data. A key aspect of our model is the sentence learner, which can perform all tasks (predicate identification, POS tagging, dependency parsing) subsidiary to semantic role labeling, making our model self-sufficient and directly applicable to plain text. The benefits of our approach have been empirically demonstrated by the experiments undertaken within this study. Our model significantly outperforms baselines, even without cross-view training; *with* cross-view training, meanwhile, it achieves an impressive performance across different languages. Through ablation studies, we have shown that multi-task hidden features, the sentence learner, and the cross-view training all contribute to improving the performance of our semi-supervised SRL model.

Experiments with cross-view training, in the context of semantic role labeling, have shown that applying views that cannot "see" the predicate of interest significantly reduces model performance. Consequently, the sentence was segmented, and words before and after the predicate were subjected to different treatment. This ensured that information from the predicate percolated to every word in the sentence. Secondly, we explored how the strategy for selecting the target predicate (in sentences containing multiple candidates) influenced performance. Experiments indicated that the worst

strategy was invariably to select the predicate with the highest confidence. Therefore, we proposed a strategy to choose a predicate randomly from candidate predicates, which led to a better SRL performance compared with other, more sophisticated strategies.

3. *Can we obtain semantic role labelers for target (low-resource) languages without target-language annotations?*

We developed a cross-lingual SRL and demonstrated that it can effectively leverage unlabeled parallel data. Experimental results across languages have shown improvements over competitive baselines. Through several ablation studies, we have also confirmed that multilingual contextualized word embeddings (multilingual BERT), Gaussian noise injection and the semantic role compressor all contribute to improving the performance of our cross-lingual SRL model. Conversely, when non-contextualized embeddings (MUSE) were deployed in place of multilingual BERT, it was found that duplicate words severely impaired the performance of our model. Although the addition of positional embeddings can alleviate this problem, the use of multilingual BERT is a better solution, which makes contextualized multilingual embeddings an indispensable part of our model.

Following PropBank-style guidelines, we also provided two manually annotated datasets with a view to evaluating our model more effectively. The two datasets contained 258 German and 304 Chinese sentences, the size of which was smaller than UPB. Nonetheless, our datasets evinced a greater degree of accuracy than datasets created semi-automatically, since the manual annotation was carried out by native speakers.

7.2 Future Research Directions

Span-based SRL The present thesis focused exclusively on dependency-based SRL within three settings. These were: supervised, semi-supervised and cross-lingual SRL. An obvious future research direction is the extension of our methods to span-based SRL. For supervised and semi-supervised SRL, the dependency-information extrac-

tor and the sentence learner are not influenced by the SRL annotation style. As a result, they can be directly transferred to span-based semantic role labeling. A major challenge relates to the application of CVT to span-based SRL: since the output of span-based SRL consists of BIO transitions, it is quite difficult to capture dependencies between output tags correctly, without seeing the complete sentence.

A potential solution would be to simplify the tagging scheme during semi-supervised training. In other words, auxiliary modules should concentrate on the role of each span, while ignoring the question of whether the current word is the beginning or end of the span. Hence, the global decoding stage of span-based SRL would be eliminated, which would facilitate the application of semi-supervised models such as CVT.

When adapting our cross-lingual SRL model to span-based SRL, we need to pay special attention to semantic role compression, since here, the semantic roles are assigned to the entire span, which usually contains multiple words. Once again, since word order generally changes during the source-to-target translation, the BIO tagging scheme must be abandoned during cross-lingual training. As a result, the first step would be to check whether fix-sized vectors can record information for every word in a span. If the answer is yes, the semantic role compressor can easily be adapted to compress information for span-based SRL. Nonetheless, if fix-sized vectors cannot record information for multiple words, one option would be to expand the compressed role information to an unlimited size. In this case, the decompression stage would also need to be modified to match the new compression strategy adopted for span-based SRL.

Cross-lingual SRL A further avenue of potential research consists of the combination of cross-lingual training with other methods of deep learning, such as adversarial training. Our cross-lingual model relies heavily on multilingual contextualized word embeddings. Replacing contextualized word embeddings with non-contextualized ones would result in inferior results (see Section 6.5.5). One cannot assume that ELMo, BERT and other multilingual contextualized word embeddings will always be available for certain extremely low-resource languages, simply because the former have become popular in NLP applications. The question of how to transfer cross-lingual signals, while not relying on the similarity between multilingual contextualized word embeddings or the results of word-alignment tools, is an important direction for future

research.

Meanwhile, another constraint of current cross-lingual SRL models is their dependence on a parallel corpus or high-quality translation tools. As with multilingual contextualized word embeddings, these resources might not be available for some languages. The ideal cross-lingual SRL model would only require SRL annotations in the source language and large-scale unlabeled data in the target language. As translation models develop, one finds that some translation systems do not require parallel data during training, so to some degree, they capture correlations between different languages without aligned sentence pairs.

Adversarial training has been applied in cross-lingual dependency parsing (Ahmad et al., 2019a), and this can also be utilized to improve the quality of compressed role information in our cross-lingual SRL model. A language-identification task, for instance, may be performed if one stacks a classifier on top of the compressor. Following the terminology in adversarial-learning literature, the classifier would be a *discriminator* and the compressor a *generator*. In this case, the objective of adversarial training would then be to oblige the generator to produce representations independent of language. A further step would be to inject predicate information into the *discriminator*, which would enable the classifier to learn the argument distributions for different predicates, thereby improving the role information learnt by the target-side semantic role labeler.

Bibliography

Fillmore, C. (1968). “The case for case”. In: pp. 1–88.

Minsky, M. (1974). “A framework for representing knowledge”. In:

Noreen, E. W. (1989). *Computer-intensive methods for testing hypotheses*. Wiley New York.

Dowty, D. (1991). “Thematic proto-roles and argument selection”. In: *language* 67.3, pp. 547–619.

Yarowsky, D. (June 1995). “Unsupervised Word Sense Disambiguation Rivaling Supervised Methods”. In: *33rd Annual Meeting of the Association for Computational Linguistics*. Cambridge, Massachusetts, USA: Association for Computational Linguistics, pp. 189–196. DOI: 10.3115/981658.981684.

Koehn, P. (2005). “Europarl: A parallel corpus for statistical machine translation”. In: *MT summit*. Vol. 5. Citeseer, pp. 79–86.

Xu, B. (Sept. 2019). *NLP Chinese Corpus: Large Scale Chinese Corpus for NLP*. Version 1.0. DOI: 10.5281/zenodo.3402023.

Ahmad, W. U., Z. Zhang, X. Ma, K.-W. Chang, and N. Peng (Nov. 2019a). “Cross-Lingual Dependency Parsing with Unlabeled Auxiliary Languages”. In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China: Association for Computational Linguistics, pp. 372–382. DOI: 10.18653/v1/K19-1035.

Ahmad, W., Z. Zhang, X. Ma, E. Hovy, K.-W. Chang, and N. Peng (June 2019b). “On Difficulties of Cross-Lingual Transfer with Order Differences: A Case Study on Dependency Parsing”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 2440–2452. DOI: 10.18653/v1/N19-1253.

Akbik, A., V. Kumar, and Y. Li (Nov. 2016). “Towards Semi-Automatic Generation of Proposition Banks for Low-Resource Languages”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 993–998. DOI: 10.18653/v1/D16-1102.

Aminian, M., M. S. Rasooli, and M. Diab (2017). “Transferring semantic roles using translation and syntactic information”. In: *arXiv preprint arXiv:1710.01411*.

Aminian, M., M. S. Rasooli, and M. Diab (May 2019). “Cross-Lingual Transfer of Semantic Roles: From Raw Text to Semantic Roles”. In: *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*. Gothenburg, Sweden: Association for Computational Linguistics, pp. 200–210. DOI: 10.18653/v1/W19-0417.

Aziz, W., M. Rios, and L. Specia (2011). “Shallow Semantic Trees for SMT”. In: *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland, pp. 316–322.

Bahdanau, D., K. Cho, and Y. Bengio (2015). “Neural machine translation by jointly learning to align and translate”. In: *Proceedings of the 3rd International Conference on Learning Representations*. San Diego, California.

Baker, C. F., C. J. Fillmore, and J. B. Lowe (Aug. 1998). “The Berkeley FrameNet Project”. In: *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*. Montreal, Quebec, Canada: Association for Computational Linguistics, pp. 86–90. DOI: 10.3115/980845.980860.

Björkelund, A., L. Hafdell, and P. Nugues (June 2009). “Multilingual Semantic Role Labeling”. In: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*. Boulder, Colorado: Association for Computational Linguistics, pp. 43–48. URL: https://www.aclweb.org/anthology/W09-1206.

Björkelund, A., B. Bohnet, L. Hafdell, and P. Nugues (2010). “A high-performance syntactic and semantic dependency parser”. In: *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*. Association for Computational Linguistics, pp. 33–36.

Blum, A. and T. Mitchell (1998). “Combining labeled and unlabeled data with co-training”. In: *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 92–100.

Bojanowski, P., E. Grave, A. Joulin, and T. Mikolov (2017). “Enriching Word Vectors with Subword Information”. In: *Transactions of the Association for Computational Linguistics* 5, pp. 135–146.

Cai, J., S. He, Z. Li, and H. Zhao (Aug. 2018). “A Full End-to-End Semantic Role Labeler, Syntactic-agnostic Over Syntactic-aware?” In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 2753–2765. URL: https://www.aclweb.org/anthology/C18-1233.

Cai, R., X. Zhang, and H. Wang (2016). “Bidirectional Recurrent Convolutional Neural Network for Relation Classification”. In: *Proceedings of the 54th Annual Meeting of the Asso-*

ciation for Computational Linguistics (Volume 1: Long Papers). Berlin, Germany, pp. 756–765.

Cai, R. and M. Lapata (Nov. 2019a). "Semi-Supervised Semantic Role Labeling with Cross-View Training". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 1018–1027. DOI: 10.18653/v1/D19-1094.

Cai, R. and M. Lapata (Mar. 2019b). "Syntax-aware Semantic Role Labeling without Parsing". In: *Transactions of the Association for Computational Linguistics* 7, pp. 343–356. DOI: 10.1162/tacl_a_00272.

Cai, R. and M. Lapata (Nov. 2020). "Alignment-free Cross-lingual Semantic Role Labeling". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 3883–3894. DOI: 10.18653/v1/2020.emnlp-main.319.

Carreras, X. and L. Màrquez (2005). "Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling". In: *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*. Ann Arbor, Michigan, pp. 152–164.

Chelba, C., T. Mikolov, M. Schuster, Q. Ge, T. Brants, and P. Koehn (2013). "One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling". In: *CoRR* abs/1312.3005. arXiv: 1312.3005. URL: http://arxiv.org/abs/1312.3005.

Chen, X., C. Lyu, and I. Titov (Nov. 2019). "Capturing Argument Interaction in Semantic Role Labeling with Capsule Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 5415–5425. DOI: 10.18653/v1/D19-1544.

Chiu, J. P. and E. Nichols (2016). "Named entity recognition with bidirectional LSTM-CNNs". In: *Transactions of the Association for Computational Linguistics* 4, pp. 357–370.

Clark, K., M.-T. Luong, C. D. Manning, and Q. Le (Oct. 2018). "Semi-Supervised Sequence Modeling with Cross-View Training". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 1914–1925. DOI: 10.18653/v1/D18-1217.

Daza, A. and A. Frank (Nov. 2019). "Translate and Label! An Encoder-Decoder Approach for Cross-lingual Semantic Role Labeling". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 603–615. DOI: 10.18653/v1/D19-1056.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805*.

Dozat, T. and C. D. Manning (2016). "Deep Biaffine Attention for Neural Dependency Parsing". In: *CoRR* abs/1611.01734.

Duvenaud, D. K., D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, et al. (2015). "Convolutional Networks on Graphs for Learning Molecular Fingerprints". In: *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., pp. 2224–2232.

Dyer, C., M. Ballesteros, W. Ling, A. Matthews, and N. A. Smith (2015). "Transition-Based Dependency Parsing with Stack Long Short-Term Memory". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China, pp. 334–343.

Fei, H., M. Zhang, and D. Ji (July 2020). "Cross-Lingual Semantic Role Labeling with High-Quality Translated Training Corpus". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online.

FitzGerald, N., O. Täckström, K. Ganchev, and D. Das (2015). "Semantic Role Labeling with Neural Network Factors". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal, pp. 960–970.

Francis, N. and H. Kucera (1979). *Brown Corpus Manual*. Tech. rep. Providence, Rhode Island, USA: Department of Linguistics, Brown Unviersity.

Gal, Y. and Z. Ghahramani (2016). "A Theoretically Grounded Application of Dropout in Recurrent Neural Networks". In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett. Curran Associates, Inc., pp. 1019–1027.

Giuglea, A.-M. and A. Moschitti (2006). "Semantic role labeling via framenet, verbnet and propbank". In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pp. 929–936.

Guo, J., D. Shah, and R. Barzilay (Oct. 2018). "Multi-Source Domain Adaptation with Mixture of Experts". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 4694–4703. DOI: 10.18653/v1/D18-1498.

Hajič, J., M. Ciaramita, R. Johansson, D. Kawahara, M. A. Martı, L. Màrquez, et al. (June 2009a). "The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages". In: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*. Boulder, Colorado: Association for Com-

putational Linguistics, pp. 1–18. URL: https://www.aclweb.org/anthology/W09-1201.

Hajič, J., M. Ciaramita, R. Johansson, D. Kawahara, M. A. Martı, L. Màrquez, et al. (2009b). "The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages". In: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*. Boulder, Colorado, pp. 1–18.

Hajičová, E., Z. Kirschner, and P. Sgall (1999). *A Manual for Analytic Layer Annotation of the Prague Dependency Treebank (English Translation)*. Tech. rep. Prague, Czech Republic: ÚFAL MFF UK.

He, L., K. Lee, M. Lewis, and L. Zettlemoyer (2017). "Deep Semantic Role Labeling: What Works and What's Next". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada, pp. 473–483.

He, S., Z. Li, H. Zhao, and H. Bai (July 2018). "Syntax for Semantic Role Labeling, To Be, Or Not To Be". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 2061–2071. DOI: 10.18653/v1/P18-1192.

He, S., Z. Li, and H. Zhao (Nov. 2019). "Syntax-aware Multilingual Semantic Role Labeling". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 5350–5359. DOI: 10.18653/v1/D19-1538.

Hochreiter, S. and J. Schmidhuber (Nov. 1997). "Long Short-Term Memory". In: *Neural Computation* 9, pp. 1735–1780.

Johansson, R. and P. Nugues (Aug. 2008). "Dependency-based Syntactic–Semantic Analysis with PropBank and NomBank". In: *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*. Manchester, England: Coling 2008 Organizing Committee, pp. 183–187. URL: https://www.aclweb.org/anthology/W08-2123.

Kearnes, S., K. McCloskey, M. Berndl, V. Pande, and P. Riley (2016). "Molecular Graph Convolutions: Moving beyond Fingerprints". In: *Journal of Computer-Aided Molecular design* 30.8, pp. 595–608.

Khan, A., N. Salim, and Y. J. Kumar (2015). "A framework for multi-document abstractive summarization based on semantic role labelling". In: *Applied Soft Computing* 30, pp. 737–747.

Kingma, D. P. and J. Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Kiperwasser, E. and Y. Goldberg (2016). "Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations". In: *Transactions of the Association for Computational Linguistics* 4, pp. 313–327.

Kipf, T. N. and M. Welling (2017). "Semi-supervised classification with graph convolutional networks". In: *Proceedings of the 5th International Conference on Learning Representations*. Toulon, France.

Kozhevnikov, M. and I. Titov (Aug. 2013). "Cross-lingual Transfer of Semantic Role Labeling Models". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 1190–1200. URL: https://www.aclweb.org/anthology/P13-1117.

Lample, G., A. Conneau, M. Ranzato, L. Denoyer, and H. Jégou (2018). "Word translation without parallel data". In:

Lang, J. and M. Lapata (2010). "Unsupervised Induction of Semantic Roles". In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Los Angeles, California, pp. 939–947.

Lang, J. and M. Lapata (Sept. 2014). "Similarity-Driven Semantic Role Induction via Graph Partitioning". In: *Computational Linguistics* 40.3, pp. 633–669. DOI: 10.1162/COLI_a_00195.

Lei, T., Y. Zhang, L. Màrquez, A. Moschitti, and R. Barzilay (2015). "High-Order Low-Rank Tensors for Semantic Role Labeling". In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado, pp. 1150–1160.

Li, Z., S. He, J. Cai, Z. Zhang, H. Zhao, G. Liu, et al. (Oct. 2018). "A Unified Syntax-aware Framework for Semantic Role Labeling". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 2401–2411. DOI: 10.18653/v1/D18-1262.

Li, Z., S. He, J. Zhou, H. Zhao, K. Parnow, and R. Wang (2019a). "Dependency and Span, Cross-Style Semantic Role Labeling on PropBank and NomBank". In: *arXiv preprint arXiv:1911.02851*.

Li, Z., S. He, H. Zhao, Y. Zhang, Z. Zhang, X. Zhou, et al. (July 2019b). "Dependency or Span, End-to-End Uniform Semantic Role Labeling". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33, pp. 6730–6737. DOI: 10.1609/aaai.v33i01.33016730.

Li, Z., H. Zhao, R. Wang, and K. Parnow (Nov. 2020). "High-order Semantic Role Labeling". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, pp. 1134–1151. DOI: 10.18653/v1/2020.findings-emnlp.102.

Ling, W., C. Dyer, A. W. Black, and I. Trancoso (2015). "Two/Too Simple Adaptations of Word2Vec for Syntax Problems". In: *Proceedings of the 2015 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies. Denver, Colorado: Association for Computational Linguistics, pp. 1299–1304.

Liu, Y., F. Wei, S. Li, H. Ji, M. Zhou, and H. WANG (2015). "A Dependency-Based Neural Network for Relation Classification". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Beijing, China, pp. 285–290.

Liu, Z., J. Shin, Y. Xu, G. I. Winata, P. Xu, A. Madotto, et al. (Nov. 2019). "Zero-shot Cross-lingual Dialogue Systems with Transferable Latent Variables". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 1297–1303. DOI: 10.18653/v1/D19-1129.

Lyu, C., S. B. Cohen, and I. Titov (Nov. 2019). "Semantic Role Labeling with Iterative Structure Refinement". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 1071–1082. DOI: 10.18653/v1/D19-1099.

Ma, X. and E. Hovy (Aug. 2016). "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1064–1074. DOI: 10.18653/v1/P16-1101.

Marcheggiani, D., A. Frolov, and I. Titov (2017). "A Simple and Accurate Syntax-Agnostic Neural Model for Dependency-based Semantic Role Labeling". In: *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. Vancouver, Canada, pp. 411–420.

Marcheggiani, D. and I. Titov (2017). "Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark, pp. 1506–1515.

Marcheggiani, D., J. Bastings, and I. Titov (2018). "Exploiting Semantics in Neural Machine Translation with Graph Convolutional Networks". In: *Proceedings of the the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018)*. New Orleans, US.

Marcus, M. P., B. Santorini, and M. A. Marcinkiewicz (1993a). "Building a Large Annotated Corpus of English: The Penn Treebank". In: *Computational Linguistics* 19.2, pp. 313–330.

Marcus, M. P., B. Santorini, and M. A. Marcinkiewicz (1993b). "Building a Large Annotated Corpus of English: The Penn Treebank". In: *Computational Linguistics* 19.2, pp. 313–330. URL: https://www.aclweb.org/anthology/J93-2004.

McClosky, D., E. Charniak, and M. Johnson (June 2006). "Effective Self-Training for Parsing". In: *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*. New York City, USA: Association for Computational Linguistics, pp. 152–159. URL: https://www.aclweb.org/anthology/N06-1020.

Meyers, A., R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, et al. (May 2004). "The NomBank Project: An Interim Report". In: *Proceedings of the Workshop Frontiers in Corpus Annotation at HLT-NAACL 2004*. Boston, Massachusetts, USA: Association for Computational Linguistics, pp. 24–31. URL: https://www.aclweb.org/anthology/W04-2705.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013). "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., pp. 3111–3119.

Padó, S. and M. Lapata (Oct. 2005). "Cross-linguistic Projection of Role-Semantic Information". In: *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, pp. 859–866. URL: https://www.aclweb.org/anthology/H05-1108.

Palmer, M., D. Gildea, and P. Kingsbury (2005). "The Proposition Bank: An Annotated Corpus of Semantic Roles". In: *Computational Linguistics* 31.1, pp. 71–106.

Pennington, J., R. Socher, and C. Manning (Oct. 2014). "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. DOI: 10.3115/v1/D14-1162.

Peters, M., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, et al. (2018). "Deep Contextualized Word Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana, pp. 2227–2237.

Pradhan, S., A. Moschitti, N. Xue, H. T. Ng, A. Björkelund, O. Uryupina, et al. (2013). "Towards Robust Linguistic Analysis using OntoNotes". In: *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*. Sofia, Bulgaria, pp. 143–152.

Punyakanok, V., D. Roth, and W.-t. Yih (2008). "The importance of syntactic parsing and inference in semantic role labeling". In: *Computational Linguistics* 34.2, pp. 257–287.

Roth, M. and M. Lapata (2016). "Neural Semantic Role Labeling with Dependency Path Embeddings". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany, pp. 1192–1202.

Thompson, C. A., R. Levy, and C. D. Manning (2003). "A generative model for semantic role labeling". In: *European Conference on Machine Learning*. Springer, pp. 397–408.

Toutanova, K., A. Haghighi, and C. D. Manning (2008). "A Global Joint Model for Semantic Role Labeling". In: *Computational Linguistics* 34.2, pp. 161–191. DOI: 10.1162/coli.2008.34.2.161.

Van der Plas, L., T. Samardžić, and P. Merlo (2010). "Cross-lingual validity of PropBank in the manual annotation of French". In: *Proceedings of the Fourth Linguistic Annotation Workshop*. Association for Computational Linguistics, pp. 113–117.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al. (2017). "Attention is all you need". In: *Advances in neural information processing systems*, pp. 5998–6008.

Vinyals, O., L. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton (2015). "Grammar As a Foreign Language". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems*. Montreal, Canada, pp. 2773–2781.

Xue, N. and M. Palmer (2004). "Calibrating features for semantic role labeling". In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 88–94.

Xue, N., F. Xia, F. D. Chiou, and M. Palmer (2005). "The Penn Chinese TreeBank: Phrase Structure Annotation of a Large Corpus". In: *Natural Language Engineering* 11.2, pp. 207–238.

Xue, N. and M. Palmer (2009). "Adding seman-tic roles to the Chinese Treebank". In: *Natural Language Engineering* 15.1, pp. 143–172.

Zaghouani, W., M. Diab, A. Mansouri, S. Pradhan, and M. Palmer (July 2010). "The Revised Arabic PropBank". In: *Proceedings of the Fourth Linguistic Annotation Workshop*. Uppsala, Sweden: Association for Computational Linguistics, pp. 222–226. URL: https://www.aclweb.org/anthology/W10-1836.

Zheng, C. and P. Kordjamshidi (Nov. 2020). "SRLGRN: Semantic Role Labeling Graph Reasoning Network". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, pp. 8881–8891. DOI: 10.18653/v1/2020.emnlp-main.714.