Cross-lingual Semantic Role Labeling through Translation and Multilingual Learning

José Angel Daza Arévalo

Department of Computational Linguistics
Heidelberg University

This dissertation is submitted for the degree of
Doctor of Philosophy

Reviewer: Prof. Dr. Anette Frank
Second reviewer: Prof. Dr. Sebastian Padó

Submission date: 06.05.2022.

Acknowledgements

I would like to thank my supervisor Anette Frank, for believing in me since the beginning, and for her guidance, creativity and tenacity as well as constant support since the first day I came to Germany. At the start of this thesis some doubted the potential of the idea this research is based on, but she pushed me to always keep going.

This project developed as part of the Leibniz ScienceCampus Empirical Linguistics and Computational Language Modeling, supported by Leibniz Association and by the Ministry of Science, Research, and Art of Baden-Wurttemberg. Thanks to the organizers of this project who integrated a vibrant diverse international team and allowed me to be part of it. Thanks to my team members for the fruitful meetings, and stimulating coffee breaks.

Just like my cross-lingual model, I needed to be flexible during this research to keep a healthy work-life balance, I thank my colleagues as well as my flatmates and friends for all the hikes, concerts, drinks and trips that kept me sane and happy in the last four years, and specially to Esther for her companionship, support and endless conversations that made my life in Germany complete.

Most of all, I want to thank those who gave up the most for this thesis: my parents and sisters: Angel, Lety, Gaby and Fer, my niece and nephew Sofia and Joshua, and all the family back home, who have always encouraged me to follow my dreams and supported my move abroad, standing virtually by my side despite the distance and time difference.

Abstract

Understanding an event means being able to answer the question *Who did what to whom?* (and perhaps also *how, when, where...*). The *what* in this sentence is called an event, and it is directly linked to a predicate, which admits event-specific roles for participants that take part in the event. Semantic Role Labeling (SRL) is the task of assigning semantic argument structures to words or phrases in a sentence, which comprises the predicate, its sense, the participants, and the roles they play in the event or state of affairs.

Nowadays the prevailing method for SRL is supervised learning, hence the quality of SRL systems is dependent on annotated training resources. In this thesis we address the problem of improving SRL performance for languages other than English. Given that annotation of SRL resources is time consuming, latest improvements on SRL have focused mainly on English; especially since the use of deep learning in Natural Language Processing (NLP) became the state-of-the-art (SOTA), annotated resources in other languages are not sufficient to compete with the latest improvements we witness for English.

Earlier research has tried to address the lack of training resources in specific languages with bilingual annotation projection methods, or monolingual data augmentation approaches to generate more labeled data that can be later used to train a labeler. Instead, we explore in this work a novel and flexible Encoder-Decoder architecture for SRL that is robust enough to work with more than two languages at the same time, immediately benefiting from more available training data. We are the first to apply sequence transduction for monolingual and cross-lingual SRL, and show that the Encoder-Decoder architecture yields competitive performance with the sequence labeling approaches. Moreover, by capitalizing on existing Machine Translation (MT) research, our model is capable of learning to *translate* from English to other target languages and *label* predicates and semantic roles on the target side within a *single inference step*. We show that – similar to multi-source machine translation – the proposed architecture can profit from multiple input languages and knowledge learned during translation to improve labeling performance on the otherwise resource-poor target languages. We see potential for future development of this framework for diverse structured prediction tasks.

In addition, this work addresses the long-standing problem of SRL annotation incompatibility across languages found in existing corpora; these divergences hinder the development of unified multilingual solutions for this task. To address and alleviate this problem, we define an automatic process for creating a new multilingual SRL corpus which is parallel, contains unified predicate senses and semantic roles across languages, and includes a manually validated test set on source and target sides. We demonstrate that this corpus is better suited than existing ones when used for joint multilingual training with neural models on lower-resource languages. Our work on this corpus is restricted to German, French, and Spanish as target languages; however, we see great potential to extend it to further languages.

In short, we propose the first model that is capable of solving the SRL task in a single language, as well as performing cross-lingual SRL via joint translation and semantic argument structure labeling while resorting to high-quality MT. Additionally, our novel annotation projection method allows us to transfer existing annotations into new languages to create a densely labeled parallel cross-lingual SRL resource with human-validated test data.

Table of Contents

Chapter 1

Introduction

1.1 Motivation

SRL is the task of automatically finding predicate-argument structures and assigning the appropriate roles to words or phrases in a sentence. SRL information has proven to be useful for related NLP tasks such as MT (Liu and Gildea, 2010; Marcheggiani et al., 2018), Relation Extraction (Shi and Lin, 2019) and Abstract Meaning Representation parsing (Wang et al., 2015a); as well as downstream applications that require understanding such as question answering (Chen et al., 2013), metaphor detection (Stowe et al., 2019), multi-document summarization (Khan et al., 2015), commonsense reasoning (Paul and Frank, 2020) and reading comprehension (Berant et al., 2014; Wang et al., 2015b; Mihaylov and Frank, 2019).

Since SRL is beneficial for language understanding, it is desirable to develop systems that are equally effective for different languages. So far, the best results have been obtained by using supervised machine learning, therefore, a large amount of high-quality annotated data is required to obtain them. This is problematic when dealing with several languages given the fact that a substantial amount of work is required, such as: i) the design of predicate and role definitions, ii) the development of guidelines that define a consistent annotation process, iii) linguistically trained annotators that apply the given definitions to a corpus of examples and, iv) the examples in the dataset need to be balanced to overcome label sparsity and allow models to learn and generalize from the data.

The major projects for formalizing role semantics were first conducted for English. Currently, the biggest projects include FrameNet (Fillmore, 1976; Baker et al., 2003), PropBank (Palmer et al., 2005) and VerbNet (Kipper-Schuler, 2006). FrameNet and PropBank in particular have been widely used because they already comprise a large enough amount of high-quality annotated examples to allow Machine Learning (ML) systems to perform well on the task. For this reason, in this thesis we refer to any language other than English as a *lower-resource language*, given that English has received considerably more attention in SRL research.

Even though the theory of role semantics was carefully crafted and defined since the 70s using different linguistic formalisms, the latest improvements we can observe for English SRL performance as a ML task have occurred mainly because of enhancement of architectures and training techniques in Deep Learning (DL). It is true that using DL alleviates the need for human-engineered features; however, one of its main drawbacks is that a considerable amount of training data is needed to achieve competitive results. While nowadays there are annotated resources available in other languages such as German (Burchardt et al., 2009), Spanish (Subirats and Sato, 2003), Japanese (Saito et al., 2008), Czech (Hajič et al., 2009), or Arabic (Pradhan et al., 2012) (just to mention a few), they are not as extensive and complete as their English counterparts, or they were separately annotated with language-specific role-sets, hampering the generalization and, as a consequence, impeding other languages to reach the same performance as English.

In this thesis we will focus on the PropBank-style semantic role annotations, given that PropBank defines a compact set of roles, contains the biggest high-quality annotation corpus for English and more importantly, there is already existing work that has confirmed the validity of extending a common PropBank semantic role set to other languages (van der Plas et al., 2010; Akbik et al., 2015). These resources were created by applying an automatic semantic role labeler on English sentences, from an existing parallel corpus, then transferring the labels to the target languages via annotation projection techniques. Because of this, the resulting data often contains noisy source annotations, faulty bilingual alignments, and low coverage of labels due to strict projection filtering, which particularly impacts the availability of infrequent roles and rare predicates; this being particularly noticeable when applied to text from a different domain.

Here we aim to address these resource-bottleneck issues by relying on the latest advances in neural NLP models to find more robust methods for transferring SRL data to other languages. We use the available high-quality SRL data that already exists in English and propose to transfer it to other languages with the goal of obtaining good-quality training data for them. To do this, we explore architectures that have already proven to be successful in MT when dealing with lower-resource languages, namely the Multi-source (Zoph and Knight, 2016) and Multi-target Encoder-Decoder models (Firat et al., 2016a). The main concept of these models is to use a single architecture for several language pairs where the model can profit from the language with more and higher-quality data in order to improve results for the remaining target languages. Our contribution is to apply similar techniques to the SRL task. Specifically, we propose to extend the decoder from a generator of token sequences to one that generates word tokens in the target language interspersed with SRL annotation labels. With this extension we are capable of translating from English into target languages while, at the same time, applying semantic labels on the target side. This approach comes with considerable advantages compared to annotation projection such as i) being less dependent on parallel data, thus gaining the ability to generate labeled sentences for novel domains or textual styles ii) avoiding the need for external automatic syntactic and semantic role labelers during the process, iii) bypassing the need for trained word aligners and iv) abstaining from hard-coded filtering projection rules. We expect to exploit this approach to obtain high-quality and more densely annotated training data for lower-resource languages.

1.2 Research Questions

The main goal of this work is to find methods that allow us to augment the availability of labeled data for PropBank SRL in languages other than English. We aim to develop methods for obtaining high-quality annotations while avoiding a resource-consuming manual annotation scenario.

This thesis raises the following research questions:

- **Is it possible to jointly translate and project annotations from English gold labeled sentences to a chosen target language (e.g. German) by exploiting Encoder-Decoder architectures?** The main problem when transferring annotations across languages is that lexical and semantic shifts occur in different languages, making the mapping of meaning from any source to a target language far from trivial. Also, we are interested in preserving the correct predicate sense and arguments that are present in English on the target side to produce high quality labeled sentences. Both problems may still persist in our approach; however, with a neural model that jointly translates from a source language into a target language while labeling the target sentence, we do no longer need to have corpus-specific filters nor parallel data at prediction time.

- Once we establish a method that is able to generate more labeled SRL data in different target languages, we will further assess the impact of a joint multilingual system that learns from different languages at the same time. Hence, we will try to answer the question: **will joint multilingual learning result in further improvements of SRL performance, particularly for lower-resource languages?**

- Our cross-lingual SRL labeling approach using a seq2seq model raises further questions related to the quality of the produced labeled data, and how to assess it. For example, **how can we determine the quality of the labeled sentences that we generate, regarding both the naturalness of the translations and SRL labeling quality?** and **how can we control meaning preservation of the generated sentences?** Since we are using an Encoder-Decoder model that generates words and labels *from scratch* instead of labeling an existing sentence (as is typical for a sequence labeling architecture), there is no guarantee that the decoded target sentences will be token-identical to an existing reference for evaluation, hence there is no gold-standard to compare to.

- Finally, our work is confronted with a lack of homogeneously labeled SRL training data across different languages, which is another obstacle when it comes to evaluating the impact of our novel cross-lingual SRL labeling architecture. Specifically, we ask: **how can we automatically generate a cross-lingual dataset that is fully compatible across languages, shares the same semantic role-sets, and at the same time has parallel information?** Such a dataset is necessary to fully explore the capabilities of multilingual learning and cross-lingual projection, while in the current situation no such evaluations are straight-forward.

1.3 Contributions

The main contributions of this thesis are:

- We propose the first Encoder-Decoder model for PropBank SRL that can translate a source sentence to a target language while at the same time applying semantic labels to the target sentence. This model is applicable for monolingual, multilingual and cross-lingual settings.

- We benchmark the performance of our model in *monolingual* settings and find that it outperforms the state-of-the-art SRL sequence labeling models in the case of English and improves monolingual baselines by applying joint *multilingual* learning in the case of lower-resource languages. Finally, we demonstrate that the *cross-lingual* system generates sentences that are highly grammatical and natural with sensible PropBank labels. We confirm the quality of the outputs by performing human evaluation on a subset of the generated sentences and by re-training systems with the generated data that obtain performance improvements.

- We construct the first fully parallel SRL dataset with dense, homogeneous annotations and human-validated test sets covering three new languages paired with the original high-quality English annotations. We employ a method to generate training sets by projecting existing SRL annotations automatically from English to lower-resource languages improving coverage and quality compared to previous techniques. We also describe a fast method to create a human-supervised test set that allows us to explore the syntactic and semantic divergences in SRL across languages and assess performance differences.

1.4 Thesis Overview

In the remainder of this thesis, we first provide the background on several aspects that we need to consider in order to achieve our proposed contributions. In **Chapter 2** we describe what SRL is and how it is defined as a ML task (Section 2.1). We then describe the fundamentals of neural models and how they are applied for solving different structured sequence label prediction tasks such as SRL (Section 2.2). Next, we present the Encoder-Decoder (Enc-Dec) architecture (Section 2.3) which is an extension of sequence labeling that allows us to predict sequences of varying lengths, hence they are also frequently called Sequence-to-Sequence (seq2seq) models. This architecture has been very successful in Neural Machine Translation (NMT) models for translating sequences across languages and also for mapping of sentences to structured representations, such as semantic parsing tasks. After this, we explain in Section 2.4 more complex architectures that have been built for NMT to deal with several language pairs at the same time, including multi-source and multi-target seq2seq that include several languages in a joint model and serve as the basis for our proposed architecture. Such models have already shown performance and quality improvements for the translation task, and in later chapters we demonstrate that they also help for our task. Next, we talk about the Transformer architecture (Section 2.5), which provides a robust method for solving a multiple variety of NLP problems such as language modeling, sequence labeling and sequence-to-sequence tasks and have also proved to work very well in multilingual settings. We close this chapter with a brief explanation of Contextualized Language Models (Section 2.6), which demonstrated to be very useful tools for our purposes in this work.

In **Chapter 3** we focus on the related work that exists for Neural SRL. We first describe existing SOTA models for the SRL task: from end-to-end deep neural models to the addition of self-attention and syntactic information for improved labeling performance (Section 3.1). The majority of those systems work only for English however, we also include recent attempts to improve SRL for other languages. We continue with a discussion on data augmentation methods that have been applied for lower-resource SRL, including monolingual data augmentation (Section 3.2), cross-lingual annotation projection (Section 3.3) and the training of joint multilingual models (Section 3.4).

Chapter 4 gives a broad overview of the areas of opportunity for the current SRL approaches and the reasons that motivate our proposed model. We aim to provide solutions that overcome the different problems that arise when creating more SRL resources in other languages using currently existing techniques.

Chapter 5 is concerned with our adaption of the Enc-Dec architecture for the task of SRL, which we are the first to formulate as a seq2seq task instead of the usual sequence labeling task formulation. In section 5.1 we explain how we adapt the SRL task to fit the Enc-Dec approach, and in Section 5.2 we describe the basic architecture we used to test our proposed method. We then describe our experimental setup (Section 5.3) and finally the results of our approach compared to monolingual SOTA models (Section 5.4). One of the main drawbacks of using an Enc-Dec is that the Decoder can generate sequences of varying length which do not necessarily correspond to a gold reference. Therefore, in this Chapter we limit evaluation to monolingual models, to have more control over the expected outputs (with the addition of a copying mechanism) and properly assess the feasibility of using an Enc-Dec for the SRL task. We find that our monolingual models give competitive results in the case of English but, as expected, lag behind when using the (considerably smaller or noisier) training data available for French and German.

In **Chapter 6**, first we discuss in detail the multilingual and cross-lingual settings of our approach (sections 6.1 and 6.2); then, section 6.3 states a description of the datasets we use to train these models, as well as the different experiments (section 6.4). For evaluation (Section 6.5), we discuss suitable methods for our novel task setup, which addresses the issue of not having a gold-standard available when generating labeled sequences from scratch in a different language, where copying the source words does not help anymore. We demonstrate in this chapter that multilingual settings boost labeling performance on the lower-resource languages because of parameter sharing with the higher-quality English data, and also show that, after applying appropriate output filtering, the generated sequences in a different target language (the cross-lingual setting) are grammatical and contain sensible labels.

Motivated by the annotation inconsistencies found in the datasets used for training the models of Chapter 6, we describe in **Chapter 7** a different approach to create compatible training data that is more suitable for multilingual and cross-lingual experiments. This separate approach benefits from the latest advances in large-scale contextualized multilingual language models. Section 7.1 states our motivations for constructing a fully parallel cross-lingual dataset, besides the data that is already available. In Section 7.2 we explain our approach for obtaining parallel data, in Section 7.3 we describe the test set construction which involved human validation and in Section 7.4 we describe our automatic annotation transfer method to the lower-resource languages. We finally perform experiments that assess the quality of our generated corpus and its capacity to help models learn in cross-lingual scenarios (Section 7.5). Our method described here led us to successfully publish the first fully parallel cross-lingual dataset for SRL.

Lastly, **Chapter 8** summarizes the findings of this thesis and discusses the open questions and potential future directions of research.

1.5 Published Work

The majority of the research presented in this thesis is an extension of published works that were first-authored by the author of this thesis. The first Encoder-Decoder model created for the Semantic Role Labeling task together with the analysis and results for English SRL was presented in the third Workshop on Representation Learning for NLP at the ACL Conference (Daza and Frank, 2018) where the feasibility of such a model for SRL was demonstrated.

The extension of the Encoder-Decoder architecture for multilingual and cross-lingual scenarios was published at the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (Daza and Frank, 2019), where we assess the potential of multilingual joint learning and the limitations caused by heterogeneous SRL annotations across languages in existing multilingual corpora. Additionally, we show a method for generating new labeled data on lower-resource languages by using our cross-lingual setting.

Finally, after analysing the limitations found when training our general multilingual model, we describe a method for creating a dataset that accomplishes more suitable conditions for multilingual and cross-lingual scenarios. We described the process of creation of such cross-lingual parallel dataset in (Daza and Frank, 2020), presented at the Conference on Empirical Methods in Natural Language Processing (EMNLP). Specifically, in this paper we describe a novel automatic projection method for creating the training set in target languages and also describe an annotation setting for obtaining test datasets with human-validated annotations that are considerably more efficient that the SRL annotation from scratch.

The code for processing the data as well as training and evaluating models presented in this thesis are listed in Appendix C.

Chapter 2

Background

2.1 Semantic Role Labeling

SRL is the task of assigning semantic argument structure to words or phrases in a sentence, in order to recognize the events that are present in the sentence and answer the question *Who did what to whom?* (and perhaps also *how, when, where...*). The difficulty of recognizing events occurs at the semantic level, since the same event can be conveyed in different syntactic structures and surface forms. Thus, semantic predicates and roles are important for understanding events in a generalized way, regardless of how they are expressed inside individual sentences, allowing us to identify the event (predicate) and its participants (roles) at the semantic level. To illustrate this generalization consider the following example from Jurafsky and Martin (2019):

(1) a. $\underset{\text{AGENT}}{\underline{\text{John}}}$ **broke** $\underset{\text{THEME}}{\underline{\text{the window}}}$.

b. $\underset{\text{AGENT}}{\underline{\text{John}}}$ **broke** $\underset{\text{THEME}}{\underline{\text{the window}}}$ $\underset{\text{INSTRUMENT}}{\underline{\text{with a rock}}}$.

c. $\underset{\text{INSTRUMENT}}{\underline{\text{The rock}}}$ **broke** $\underset{\text{THEME}}{\underline{\text{the window}}}$.

d. $\underset{\text{THEME}}{\underline{\text{The window}}}$ **broke**.

e. $\underset{\text{THEME}}{\underline{\text{The window}}}$ was **broken** $\underset{\text{AGENT}}{\underline{\text{by John}}}$.

Even when all sentences describe the same event, additional information is expressed through different syntactic elements and surface forms. For example, the syntactic subject of **1a** takes the semantic role of *agent* (the one who is breaking) while in **1c** the subject of the sentence has the role of *instrument* (the thing that is used to break something). On the other hand, we can conclude in every example that the *agent* (if present) is *John*, regardless of its position and syntactic function inside the sentences.

Once we understand the importance of semantic roles, we need to define the possible kinds of events (predicates) and subsequently what are the possible semantic roles that are involved with each defined event. Much research has been carried to create specific catalogues with a broad coverage of predicates and, more importantly, to define the granularity of predicate-specific roles. A few inventories even question whether roles should be predicate-independent or individually defined per-predicate. Some of the most prominent proposed semantic role formalisms are:

- **FrameNet (Fillmore, 1976)** which seeks to have a lexicon of fine-grained events and roles. It groups predicates with similar characteristics into *frames*, where each frame has its own set of core-roles (called frame elements) that are unique to describe the specific nature of the event. The frame also includes optional and adjunct roles that can be shared across frames as well as a hierarchy definition that captures more linguistic generalizations across frames.

- **ProtoRoles (Dowty, 1991):** a schema that goes on the opposite direction and defines a generalized set of *proto-agent* and *proto-patient* categories to which any component of a sentence can belong, provided it complies with some or all of a set of heuristically defined features. This is done purposefully to avoid the problem of proposing a fixed granular catalogue of roles and combat incomplete coverage of events.

- **PropBank (Palmer et al., 2005):** a schema that defines specific verb-senses and proposes a reduced set of core role names (*A0-A5*), which also includes syntactically grounded adjunct roles *ArgM's*. Unlike the previous two, this formalism was conceived directly as a corpus-based annotation project, which makes it more suitable for data-driven research.

- **VerbNet (Kipper-Schuler, 2006):** the largest verb lexicon currently available for English, with mappings to other annotated resources such as WordNet, PropBank, and FrameNet. It is organized into verb classes defined by Levin (1993). Each verb class can be defined by: its thematic roles (actor, agent, asset...), selectional restrictions (characteristics that words must fulfil to be considered a thematic role), and frames (syntactic patterns in which the events occur).

A good set of roles ideally should be small enough to allow for strong generalizations, yet large enough so that every argument of every predicate can be assigned a role from the set (Levin, 2019). While FrameNet is the richest lexicon and contains the most fine-grained information for semantic predicates, including verbs, nouns and adjectives; this same nature results on sparsity of annotations when building annotated resources with FrameNet, since it is very expensive to cover all possible cases and combinations of the defined predicates and roles. This reduces the predicting power of systems trained with datasets that follow such formalism. On the other hand, while the *ProtoRoles* definitions avoid the sparsity problems of any discrete role definition, this comes at the expense of lacking the granular information that could be informative enough and useful for downstream tasks.

We chose to focus on PropBank for the following reasons: i) it is grounded on a big corpus with high-quality linguistic annotations, namely the English Penn Treebank (Marcus et al., 1993), ii) it has enough granularity on its role definitions to achieve generalizations when training systems, iii) the automatic semantic role labelers that work with this annotations have shown steady improvements in the last years for the English SRL task (He et al., 2017; Strubell et al., 2018; Ouchi et al., 2018; Shi and Lin, 2019) and iv) there has been significant successful work on extending and standardizing the English PropBank set of roles into languages other than English, by using different techniques such as: direct role annotation on a foreign language (van der Plas et al., 2010), annotation projection (van der Plas et al., 2011; Akbik et al., 2015), and joint multilingual SRL learning (Kozhevnikov and Titov, 2013; Mulcaire et al., 2018).

2.1.1 PropBank

The Proposition Bank, or PropBank, is a resource of semantic role annotations added as a semantic layer on top of the Wall Street Journal (WSJ) section of the Penn TreeBank for English (Marcus et al., 1993). The semantic roles in PropBank are defined individually with respect to a particular verb sense. This definition of a predicate with its respective set of roles is also called *predicate frame*. It is worth to note that later on, nominal predicate frames and annotations were integrated with the creation of the NomBank project for English (Meyers et al., 2004).

Even though arguments are defined per predicate, the argument catalogue of PropBank is compact: The core semantic arguments are numbered (as opposed to having a descriptive name such as e.g. *Agent*), this results in a small number of core roles in the range of *A0-A5*. This is possible because, by definition, arguments belonging to different frames tend to have generalized commonalities (e.g. *A0* is the proto-agent, and *A1* is often the proto-patient). Additionally there are general *modifier* arguments, or adjuncts, that any predicate can optionally take, which are fully shared across predicate frames. For a complete catalogue of the roles originally defined for English PropBank see Table 2.1.

Tag	Role	Example
A0	Proto-Agent	He, the woman...
A1	Proto-Patient	the big window...
A2-A5	Predicate-specific	-
AM-ADV	Adverbial	Fortunately
AM-CAU	Cause	Because, as a result...
AM-DIR	Direction	to her house
AM-DIS	Discourse Marker	Also, however ...
AM-EXT	Extent Marker	more, raised by 15%
AM-LOC	Location	in Europe
AM-MNR	Manner Marker	closely, mechanically...
AM-MOD	Modals	May, could, must...
AM-NEG	Negation	Not, never...
AM-PNC	Purpose Clause	to pay, for future meetings...
AM-PRD	Secondary Predication	(adjunct carrying predication)
AM-REC	Reciprocal	himself, each other...
AM-TMP	Temporal	The next morning, On Friday...
C-Ax	Continuation of ARGx	-
R-Ax	Reference to ARGx	(*arg* in other part of the sentence)

Table 2.1 PropBank Semantic Roles defined for English (Palmer et al., 2005). Core roles (A0-A5) are specifically defined for each predicate sense. Modifier roles (AM-x) – also called adjuncts – are shared across predicate frames.

According to the PropBank annotation guidelines (Palmer et al., 2005), for every syntactic tree (sentence) in the Penn Treebank, each verb that appears in the sentence represents a *proposition* whose root (the main verb, adjective or noun) should be assigned with a specific predicate-sense. An example of a predicate frame[1] and its role definitions (in this case we keep using the verb *break* from Example 1) followed by a PropBank annotated example is:

break.01: break, cause to not be whole.

A0: breaker

A1: thing broken

A2: instrument

A3: pieces

A4: broken away from what?

EXAMPLE: $\underset{\text{AM-TMP}}{\underline{\text{Last night,}}}$ $\underset{\text{A0}}{\underline{\text{John}}}$ **broke** $\underset{\text{A1}}{\underline{\text{the window}}}$ $\underset{\text{A2}}{\underline{\text{with a rock}}}$.

[1] The full definition of the frame referenced here is available at `https://verbs.colorado.edu/propbank/framesets-english-aliases/break.html`

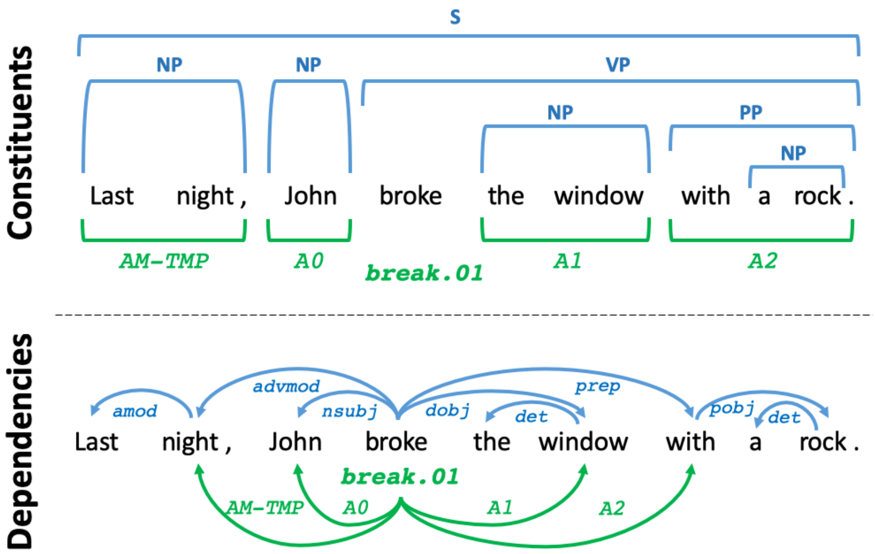


Fig. 2.1 Span-based SRL (annotations are done on top of syntactic constituents) and Dependency-based SRL (annotations are done on top of syntactic dependencies).

The annotated example above shows that the predicate frame *break.01* should be assigned to the verb **broke**. From the available roles of the assigned frame, *A0*, *A1* and *A2* are present in this particular sentence and also an adjunct *AM-TMP* which is a temporal modifier of the event. Any sentence will have annotations for as many propositions (or predicate-argument structures) as predicate senses it has.

2.1.2 Task Description

The SRL task consists of analyzing the predicate-argument structures expressed in a given sentence[2]. In Machine Learning, although there are unsupervised approaches (Grenager and Manning, 2006; Lang and Lapata, 2010; Titov and Klementiev, 2012), it is generally conceived as a *supervised problem*, where a model learns to predict the predicates and arguments from a set of annotated examples. For each target predicate in a sentence, all the dependent sub-phrases that fill a semantic role for the predicate in question must be identified and classified. Thus, the task can be subdivided in four main steps: i) predicate detection ii) predicate disambiguation, iii) argument identification and iv) argument classification.

Since the Penn Treebank was annotated with syntactic constituents, the span of semantic arguments is also based on the given constituent boundaries. Therefore, each argument is a

[2] There is a variation of the task called *implicit SRL* where the surrounding discourse is also taken into account to find roles related to a main predicate beyond the sentence boundaries (Ruppenhofer et al., 2009). Nevertheless, for the purposes of this work we focus on single sentences only.

sub-phrase of the sentence. This is what is known as **span-based SRL**. Later on, a different syntactic formalism was considered for the task, based on syntactic dependencies. In SRL, this mode of annotation is also known as **dependency-based SRL**, and it has demonstrated that it is more prone to generalize across languages (Hajič et al., 2009; Björkelund et al., 2009). The differences across annotation schemes are shown in Figure 2.1.

Importantly, both span-based and dependency-based task formulations evaluate the labeling of predicates and roles with respect to $Precision = TP/(TP + FP)$, where, for each class, *TP* are the True Positives and *FP* the False Positives; $Recall = TP/(TP + FN)$, where *FN* are the False Negatives; and finally the F1 measure is computed, which is the harmonic mean of Precision and Recall:

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{2.1}$$

For the span-based classification to be correct, the span boundaries (as well as the role label) must completely match, whereas for the dependency-based task, since only the head words of some of the syntactic dependencies are labeled, each individual word is considered to be as correctly labeled or not.

2.1.3 PropBank SRL Datasets

As mentioned earlier, PropBank emerged directly as a corpus-based schema with semantic annotations on top of the English Penn Treebank. However, SRL was popularized as a task at the *CoNLL 2004 Shared Task* (Carreras and Màrquez, 2004) and further refined for the *CoNLL 2005 Shared Task* (Carreras and Màrquez, 2005) which incorporated full-fledged syntax information and a bigger training corpus that became the official evaluation standard for span-based SRL.

On the other hand, the dependency-based SRL schema was formalized for the first time in the *CoNLL 2008 Shared Task* (Surdeanu et al., 2008), which aimed for a stronger syntax-semantic linking that directly benefited the semantic task. The original PropBank corpus annotations (based on syntactic constituents) were converted into syntactic dependency annotations by following the head finding rules of Magerman (1994). In this annotation version, only the *head word* of the syntactic children of a given predicate could have a role label assigned, and such labeled role would be the (syntactic and semantic) head of the phrase which represents the argument[3].

[3] While it is true that in many cases there is a mismatch between syntactic and semantic heads, for example with semantically empty words such as some auxiliaries, expletives or complementizers(Bender, 2013), Sur-

Monolingual Train	Language	Sentences	Total Predicates	Total Arguments	Avg Preds/sent	Avg Args/sent
CoNLL-05	EN [Span]	39,832	59,544	157,208	2.37	3.94
CoNLL-09 [Verbs]	EN [Head]	39,279	89,193	238,793	2.37	6.07
CoNLL-09 [Verbs+Nouns]	EN [Head]	39,279	179,014	393,699	4.59	10.02
CoNLL-09	DE [Head]	36,020	17,400	34,276	1.09	0.95
CoNLL-09	ES [Head]	14,329	43,821	99,054	2.85	6.91

Table 2.2 Monolingual annotated training sets with SRL annotations for different languages. We can immediately observe that non-English languages have considerably less annotations when compared to English training data.

In a like manner, the *CoNLL 2009 Shared task* (Hajič et al., 2009) extended the advantages of this unified dependency-based formalism to seven different languages, with the aim of expanding the PropBank SRL task for non-English data. The languages included in this task were chosen according to pre-existing annotated corpora: Catalan and Spanish from the AnCora corpus (Taulé et al., 2008), the Chinese Treebank 2.0 (Xue and Palmer, 2009) for Chinese, the Prague Dependency Treebank 2.0 (Hajič et al., 2006) for Czech, the SALSA corpus (Burchardt et al., 2006) for German, and the Kyoto University Text Corpus (Kawahara et al., 2002) for Japanese. The aims of this Shared Task were to convert such independently developed resources into homogeneous PropBank-style labels; however, for practical reasons, the conversions where performed in an automated manner and in some cases the language-specific annotations were preserved, resulting only in a partially compatible corpus.

For example, German possesses only core-labels that range from A0 $-$ A9, which were assigned based on the original FrameNet-like annotations in the SALSA corpus. Therefore, German predicate frames lack of a direct match with the analogous predicate senses on English PropBank and the roles do not match those from the English catalogue. Spanish and Catalan have more fine-grained PropBank-like tags, e.g. instead of having a single proto-agent $A0$ tag there are several: A0-AGT, A0-CAU, A0-EXP, etcetera, and defines roles that are not present in English such as: A0-NULL or AL-NULL.

Finally, a more important divergence occurring is the density of annotations available for non-English languages, generally having not only less sentence examples but, for each sentence, less annotated predicates and arguments, resulting in weaker training signal when training non-English models. For example, in the CoNLL-09 datasets, English contains around $39K$ sentences, and $89K$ verbal predicate-argument structures per sentence in the training set; whereas German has only $36K$ sentences, and $17K$ propositions per sentence; and Spanish $14K$ sentences and $40K$ propositions in their respective training sets. See Table 2.2 for more details.

deanu et al. (2008) report that their heuristic for converting annotations works remarkably well on nearly 99% of the cases with the advantage of being compatible at the syntactic and semantic levels.

2.1.4 Machine Learning with Features

The first automatic machine learning approach for the SRL task was conducted for English by Gildea and Jurafsky (2000) on the FrameNet corpus. In this work, they approached SRL as a classification task for constituents. They trained a complex pipeline composed of several sub-modules: first, they used an automatic parser (Collins, 1997) to obtain the syntactic parse of each training sentence and used it to extract various lexical and syntactic features such as verb-object pairs, phrase types of labeled constituents, and parse tree paths between predicates and arguments. Separately, they used the training corpus to compute statistical knowledge of the predicates, as well as information such as the prior probabilities of semantic role combinations and various lexical clustering algorithms (Hofmann and Puzicha, 1998) to generalize across possible fillers of roles. This set of pre-computed features was aggregated and combined to train different classifiers that act on linear interpolations of the different pre-computed probabilities for the corpus. This work assumed that the predicate token and predicate sense were already given. Thus, the first classifier was a binary classifier for argument identification, where each span or phrase inside a sentence was labeled as argument or non-argument. In a second step, the argument classification was performed by assigning a role label to each of the identified spans, given a sentence and a predicate.

In general, this was the seminal work that exposed the feasibility of learning to label semantic roles automatically with probabilistic knowledge drawn from a big corpus. Most of the subsequent systems integrated more robust ML frameworks that aimed to generalize beyond the raw feature probabilities seen in the corpus, and also aimed at capturing structural constraints such as repetition of roles (for example, a single predicate-argument structure can't have two *A0* roles) as well as integrating the predicate identification and classification, or even better, benefit from joint role and predicate labeling systems. These follow-up systems were also feature-based and kept using different kinds of pipelines to obtain part-of-speech tags and syntactic parsing information of the sentence, and afterwards, used that information together with statistical knowledge to assign semantic roles. Some of the common features across different works used for training systems were: the governing predicate, phrase type (NP, VP, PP) of the argument, the headword of the constituent, the path from the constituent to the predicate of interest, the named entity type, among many others (Xue and Palmer, 2004).

The promising results obtained with syntactic feature-based approaches led researchers to believe that this information was crucial for creating SRL classification systems (Pradhan et al., 2005; Johansson and Nugues, 2008; Merlo and Van Der Plas, 2009). Motivated by this, several works focused on improving the quality of the syntactic features. Pradhan et al. (2005) presented a PropBank semantic role labeler based on Support Vector Machine classifiers that included feature selection and calibration together with a combination of several parses that were trained using different *syntactic views* with the aim of improving SRL performance. Similarly, Koomen et al. (2005) and Surdeanu et al. (2007) proposed that an ensemble of classifiers could reduce the impact of the noise produced by automatic syntactic parsers in the SRL classifiers, reporting effective improvements with these solutions. Punyakanok et al. (2008) proposed to solve the task by using Integer Linear Programming with explicit constraints for labeling. Finally, Toutanova et al. (2008) proposed a system that predicted semantic argument frames as a joint structure, with strong dependencies among the arguments, and used this information to build a classifier that dramatically improved the SOTA at the time. In fact, the 80.3 F1 points achieved by this system remained unchallenged for seven years, until the advent of neural models.

Other approaches, such as Täckström et al. (2015), presented a dynamic programming algorithm for efficient constrained inference that automatically captured the majority of the structural constraints examined by Punyakanok et al. (2008). Their model showed significant improvements in efficiency and performance on both PropBank and FrameNet corpora; however, it still didn't outperform the SOTA at that time, and needed a big amount of hand-crafted features defined specifically for the training set. To close with the early SRL approaches, we mention that models aimed to learn both syntactic and semantic tasks jointly demonstrated to outperform their pipeline counterparts (Lewis et al., 2015), suggesting already that finding more sophisticated techniques to model both tasks at the same time would result in significant improvements.

2.2 Neural Network Fundamentals

As described in the previous section, feature-based approaches made improvements on SRL by combining different sets of features and using different probabilistic ML frameworks to make the learning more robust. However, the performance plateaued because of i) the increasing level of intuition necessary to develop more complex features by hand, which was consequently becoming more language-specific; ii) while lexical features (also on arguments) were known to be important, sparsity was an issue, which naturally increased when dealing with morphologically complex languages; iii) the scalability of lexical knowledge drawn from vocabulary dependent on seen training data or hand-crafted lexical resources such as WordNet (Fellbaum, 1998). Hence, a call for better lexical representations and the exploitation of lexical similarities and differences were clearly an issue that needed to be tackled for the improvement of the task performance.

The big break-through was thus the advent of deep learning models that were able to learn relevant features by themselves, as well as the learning of dense vector representations for lexical items. NLP was radically re-thought in terms of neural architectures suitable for different NLP tasks. For SRL, sequence labeling seemed the most natural setting, much akin to tagging or parsing, as a single input sequence could be labeled with BIO labels. More experience led to end-to-end architectures that dispensed with NLP pipelines.

When NLP deals with written language it is straightforward to model many of its tasks as a mapping from a sequence of words (input) to an associated sequence of task-specific labels (output). In ML, this particular paradigm is called *sequence labeling* and in the latest years neural network approaches, in particular the Recurrent Neural Network (RNN) and Transformer architectures, have consistently shown better performance than feature-based ML for sequence labeling. This is particularly true when enough training data is available. For this reason, nowadays the most common approach when solving these kind of tasks is through the use of neural network mechanisms, sometimes as feature selectors or as end-to-end neural architectures that completely avoid the need for explicit feature engineering. In the rest of this section we will introduce the basic theory and components that are used to build the neural architectures, in particular we describe the tools that will play a role at many stages of the work presented in this thesis.

2.2.1 Deep Learning for NLP

Artificial Neural Networks (ANNs) are a computational paradigm inspired by how the human brain learns by using dense interconnections of neurons. In practice, a computational neural network is composed of multiple interconnected units called *neurons*. Each neuron is an activation function that transforms the input. The neurons tend to be arranged in several layers, hence the most widely generalized term nowadays used is Deep Learning (DL).

DL can be characterized as a learning process to make **predictions** and also as a method for learning better **representations** of the data in order to optimize the predictions (Goldberg, 2017). DL approaches work by feeding numerical representations of the data into a network that applies successive mathematical transformations to the input, through a certain amount of layers, until a final layer is used to predict the output. In the supervised learning setting, the specific transformations produced by the network are learned from the given input-output mappings (training data), such that the network adjusts its parameters to approximate a general function that models the relationship between input data and its output labels. Therefore, if the network sees enough representative cases, the function it learns will be robust enough to not only approximate the seen data but also to accurately predict unseen cases.

The simplest neural network unit is called the *perceptron*; its mathematical expression is:

$$\mathbf{y} = NN(\mathbf{x}) = f(\mathbf{x}\mathbf{W} + \mathbf{b}) \tag{2.2}$$

where $\mathbf{x} \in \mathbb{R}^{d_{in}}$ is the input vector with dimensionality d_{in} (which is also referred to as the vector of input features), $W \in \mathbb{R}^{d_{in} \times d_{out}}$ is a learnable matrix of weights and b is a bias term; f is a non-linear function that permits the network to approximate more complex values that are mapped into the output vector $\mathbf{y} \in \mathbb{R}^{d_{out}}$.

A stacked arrangement of perceptrons form a Multi-layer Perceptron (MLP), which is also the simplest kind of Feed-Forward Neural Network (FFNN). In this setting, the neurons in each layer are connected to all neurons in the successive layer which can be stacked indefinitely and apply transformations to the data in a *forward* manner. For example, a 3-layer MLP can be expressed as:

$$NN(\mathbf{x}) = f''(f'(f(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2)\mathbf{W}_3 + \mathbf{b}_3) \tag{2.3}$$

In this example the network first computes the output of the first layer perceptron (with matrix W_1 and bias b_1) and given its output, the operation of the second layer is computed (with matrix W_2 and bias b_2), and so on, in an iterative manner. f, f', f'' are non-linear functions which allow the network to make the necessary transformations to approximate a complex function.

2.2.2 Distributed Word Representations

The initial difficulty when using Neural Networks in NLP tasks is to find a suitable method to represent discrete data (words and sentences) in a numerical representation $\mathbf{x} \in \mathbb{R}^{d_{in}}$ that can be processed by a neural network. A straight-forward and common approach for representing linguistic data in numerical terms is called *one-hot encoding*. This method treats each different word-type [4] as a *feature* and therefore sentences are transformed into sparse binary vectors containing a 1 if a given word is mentioned in the current sentence or 0 otherwise. The main drawback of this approach is having to use a considerable-sized vector to represent each sentence, with usually just a few relevant bits of information[5]. More

[4] A word-type is a word mention in the abstract, regardless of the context where it is being mentioned.

[5] The size of the vector is the *vocabulary size*, which comprehends all the known features (i.e. word-types) in a given training set. Therefore, if there are 5,000 unique words in a training set, a 5-word sentence is a vector $x \in \mathbb{R}^{d_{5000}}$ with 5 ones and 4,995 zeroes.

importantly, this method does not help to measure how similar two word or sentence vectors are, since all features are treated equally.

To account for the word similarity problem, several works considered to use the distributional property of language, which states that similar words normally occur in similar contexts (see (Turney and Pantel, 2010) for a detailed survey on distributional semantics). Most of these methods are known as **count-based methods**, since they compute co-occurrence matrices from large corpora to build a vector space containing vectors associated to each word-type in the known vocabulary. These methods also offer the advantage of reducing the vector size problem, since dimensionality reduction algorithms can be used to collapse the redundant information within vectors. This comes, however, at the cost of interpetability because the information is now compressed across meaningless dimensions (as opposed to the one-hot encoding where each dimension is a word-type). However, the biggest advantage of **distributed representations** is that such vectors can be used to identify similar words inside the common space to which they were compressed by finding the closest neighbors in the shared space of vectors.

A similar approach can be used to create distributed **neural word representations** through a neural Language Model (LM). The main advantage of a neural LM is that it learns simultaneously i) a distributed representation for each word and ii) the probability function for word sequences, since the model is trained to predict the next word given the previous sequence of words (Bengio et al., 2003). To learn the language modeling task, the network takes a one-hot encoded input of words and outputs the next word as seen in the sentences from a training set. Once the training is finished, the weights of the first layer of the network (which represent each word-type seen in the trianing set) have been implicitly adjusted to account for a distributed vector representations. A further non-LM neural approach to obtain continuous word representations was Word2Vec (Mikolov et al., 2013a), a technique for training neural networks that optimizes different objectives such as predicting the current word based on the context (its surrounding n-words), this is called **Continuous Bag Of Words (CBOW)**; or, conversely, predicting the surrounding words given the current input word (also called **Skip-gram**). Another popular neural-based word representation learning method is **GloVe** (Pennington et al., 2014), a count-based model with an objective function that seeks to learn word vectors in a similar way as Word2Vec with the addition of collecting and exploiting global co-occurrence statistics.

In summary, obtaining representations through any of these neural methods yields a better generalization compared to the count-based methods (neural methods are better when dealing with unknown words and sequences) and, more importantly, they successfully capture fine-grained semantic and syntactic regularities using vector arithmetic.

Because these solutions already use a neural architecture, they can be directly loaded as the first-layer (the features component) of a more complex network architecture such as RNN or Transformer. This initial layer added to the network is commonly referred to as the **embedding layer**, and it is simply a matrix $\mathbf{E} \in \mathbb{R}^{|vocab| \times d}$ that acts as a lookup table, mapping the discrete symbols (all words in the known vocabulary) into the previously learned continuous fixed-length vectors of dimension d. Importantly, the network can keep treating the information in this matrix as learnable parameters that can be adjusted accordingly with the whole architecture in order to maximize the specific task performance.

Equally important, the neural word representations can be trained to obtain shared vector spaces for more than one language, making it feasible to perform the same vector operations and retrieve word-type features for different languages at the same time. To achieve this, earlier approaches aimed at learning first monolingual representations and then use a post-hoc mapping method to project them into the same space, such as a linear transformation of vectors (Mikolov et al., 2013b), or directly using bilingual dictionaries (Faruqui and Dyer, 2014; Artetxe et al., 2017). Later approaches aimed to train models to learn directly with cross-lingual supervision (Joulin et al., 2018), or unsupervised learning (Lample et al., 2017; Ruder et al., 2019; Shareghi et al., 2019) resulting in word representations that can be useful for downstream cross-lingual tasks.

2.2.3 Recurrent Neural Networks

The FNNs described in Eq. 2.2.1 work very well if we already know the amount of features that we need to process (it works with a fixed-size sequence of features). However, for NLP problems, where words are the input features, we necessarily will deal with inputs of different sizes (sentences or documents are always varying in size). The ideal solution is then to use another kind of architecture called Recurrent Neural Networks (RNNs). RNNs (Elman, 1990) are neural models specialized in dealing with sequential data and are particularly powerful because they allow to represent arbitrarily long sequential inputs in fixed-sized vectors. On the abstract level, an RNN is a function that takes as input an ordered sequence of d_{in} -dimensional vectors $\mathbf{X} = \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$, $\mathbf{x}_t \epsilon \mathbb{R}^{d_{in}}$ and returns as output a series of d_{out} -dimensional vectors $\mathbf{O} = \mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T$, $\mathbf{o}_t \epsilon \mathbb{R}^{d_{out}}$ where every $\mathbf{o}_t$ summarizes the whole sequence up to $\mathbf{x}_t$. The recursive definition of RNN means that at each time-step *t* the RNN takes as input the previous vector $\mathbf{x}_{t-1}$ in the sequence and its own previous state $\mathbf{h}_{t-1}$ to upgrade the current state $\mathbf{h}_t$ as:

$$\mathbf{h}_t = f(\mathbf{x}_t \mathbf{W} + \mathbf{h}_{t-1} \mathbf{U} + \mathbf{b}) \tag{2.4}$$

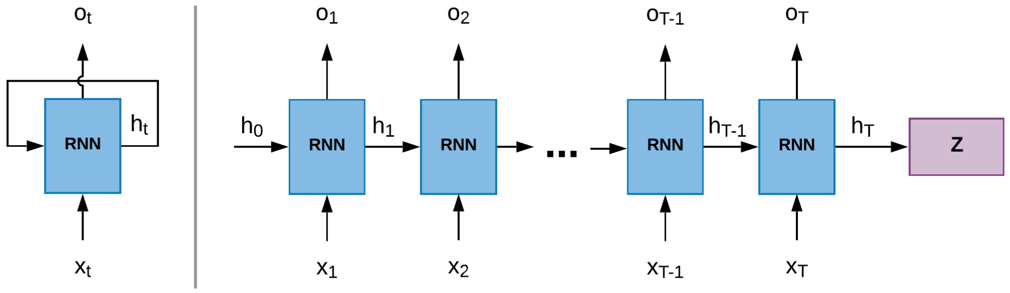


Fig. 2.2 A simple Recurrent Neural Network (left-hand side). On the right-hand side, the same network appears unfolded *through time*. At each time step *t* the network outputs a fixed-size vector $\mathbf{o}_t$ and holds a hidden state $\mathbf{h}_t$ which represents the sequence from $\mathbf{x}_0$ to $\mathbf{x}_t$. When the whole sequence is processed, $h_{t=T}$ is also called z, which is the *encoded representation* of sequence $\mathbf{X}$.

where the equation is similar to the feed-forward version (Eq. 2.2.1) with the addition of the matrix $\mathbf{U}$ which acts as the *memory* that correlates the changes of the network through time since it is modifying the previous time-step network state $\mathbf{h}_{t-1}$. Once the network processed the whole input sequence, the resulting vector $\mathbf{o}_T = \mathbf{z}$ is said to be the *encoded representation* of the entire sequence processed by the network. For a graphic description see Figure 2.2.

When we train an RNN we are obtaining an informative numeric representation of the sequential input data that can be used as a basis for making predictions, and for this reason, one can refer to an RNN as an **Encoder**. If the sequence $\mathbf{X}$ is a sentence (i.e. a sequence of word-tokens), the RNN will be able to hold in its final hidden state $\mathbf{h}_{t=T}$ the representation of the whole sentence, often denoted as vector $\mathbf{z}$.

RNNs, as defined so far, process the information left-to-right. This only preserves information from the *past* (i.e. the left part of the sequence). However, one can add information from the *future* tokens by running in parallel an RNN that processes the sequence from right-to-left $\mathbf{X}^{-1} = [\mathbf{x}_T, \mathbf{x}_{T-1}, \ldots, \mathbf{x}_1]$, in addition to the left-to-right RNN. This paradigm is called a **bidirectional encoder**, since it encodes information from *both directions* and it has shown to produce a better encoded representations. There are different ways of obtaining a more informed vector $\mathbf{z}$ using the bi-directional states. The most common approach is by concatenating or adding the left-to-right final hidden state $\overrightarrow{\mathbf{h}}_T$ and the right-to-left final hidden state $\overleftarrow{\mathbf{h}}_T$. Other common approaches are by *pooling*[6] the hidden states in a token-wise

[6] We refer to pooling here as a reduction operation that takes two or more $\mathbf{h}_t$ vectors and combines them by using operations such as addition, product, mean, max to obtain a single vector.

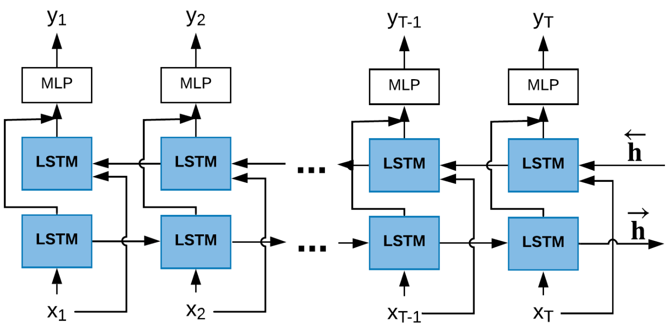


Fig. 2.3 A Bi-LSTM sequence labeler maps each token x_t from the input sequence into a corresponding label y_t.

manner (for example $\mathbf{h}'_t = \text{pooling}(\overleftarrow{\mathbf{h}}_t, \overrightarrow{\mathbf{h}}_t)$, and then combine all the $\mathbf{h}'_t$ states) or directly on the final hidden states $\mathbf{h}'_T = \text{pooling}(\overleftarrow{\mathbf{h}}_T, \overrightarrow{\mathbf{h}}_T)$.

Finally, an important problem emerges in practice with RNNs when sequences are very long. The *memory* of RNNs is quite limited for storing long-term information which produces the *vanishing gradient problem.*[7] This causes the network to incrementally *forget* the initial items of the sequence. To fix this problem, the additions of gated mechanisms such as Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) or Gated Recurrent Unit (GRU) (Cho et al., 2014) to the RNN mechanism results in a more complex way of capturing intricate statistical regularities when labeling long sequences. Therefore, in practice most architectures that implement the RNN paradigm use LSTMs or GRUs.

For consistency purposes, from now on when talking about RNNs, we will refer directly to them as **LSTM**, and **Bidirectional LSTM (Bi-LSTM)** when dealing with bidirectional networks, as this is the architecture that is more widely used in the literature and is also the one that we use in most of our experiments.

2.2.4 Sequence Labeling

In supervised learning, the term *sequence labeling* is used for describing tasks that involve the mapping of any sequence of inputs $\mathbf{x} = [x_1, x_2, ..., x_{T_n}]$ to a sequence of outputs (their respective labels) $\mathbf{y} = [y_1, y_2, ..., y_{T_m}]$ where $|\mathbf{x}| = |\mathbf{y}|$ and every y_t represents exactly one label from a predefined vocabulary L. This is a one-to-one mapping where each token in the sequence has a single label assigned. It is considered that both the inputs and the labels

[7] This problem arises because the non-linear functions used in practice (e.g. sigmoid) in the neurons squash the numeric values into a small region, preventing the neural network from learning when updating its weights too many times, since the values get smaller every time until they stop being significant.

form strongly correlated sequences, therefore the whole sequence needs to be classified at the same time in order to learn such correlations (Graves, 2012).

Taking this into account, a Bi-LSTM is a straight-forward neural **sequence labeler** (depicted in Figure 2.3). Since the encoding of information occurs step-by-step, one can make token-specific predictions for each time step. This is done in practice by adding an extra layer on top, and use it to predict the desired label y_t for each x_t. In practice, this layer is a MLP that transforms the LSTM output into a vector of L-dimensionality (where L is the number of available labels) together with a *softmax* function that normalizes the output vector into a probability distribution over predicted output labels. t:

$$\begin{gathered} z_t = (\text{BiLSTM}(x_t)\mathbf{W} + \mathbf{b}) \\ p(y_t|x_0, ..., x_{t-1}) = \text{softmax}(z_t) \end{gathered} \tag{2.5}$$

The softmax function is formally defined as:

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{L} e^{z_j}} \quad , \quad i = 1, \dots, L \quad , \quad \mathbf{z} = (z_1, \dots, z_L) \in \mathbb{R}^L \tag{2.6}$$

2.3 Encoder-Decoder Architecture

Although LSTMs work well for various NLP tasks, including sequence classification and sequence labeling, they are restricted to applying one-to-one mappings from input to output and cannot be used to map sequences of a given length *n* to sequences of a different length *m*. A straightforward task that needs such a setting is MT: When translating sentences across languages there are no direct word-to-word mappings between a *source language* and a *target language*.

The Enc-Dec architecture is a neural architecture that directly addresses this issue. It combines two LSTMs where the first one is an **encoder** which summarizes the input sequence into the fixed-size vector **z** (in the Enc-Dec formal definitions this vector is also called **c** since it holds the source *context*); the second one is the **decoder**, which generates an output sequence in an auto-regressive manner, that is, conditioned both on the representation of the previously encoded sequence **c** and the output tokens generated so far. By definition, this architecture models the mapping of a variable length source sequence into a different length target sequence, therefore it is also often referred as a seq2seq model.

A wider advantage of this architecture is that this mapping of variable-length sources to targets is not only tied to solve the language translation problem, but can be generalized to any sequence transduction scenario, to model the relationship between any kind of whole source-target sequence pairs without a restriction of length.

It is important to know a crucial problem that emerges when using this architecture: there is no hard constraint on the length of the output sequences, which makes evaluation difficult. In the case of sequence labeling, there is a one-to-one mapping across source words and the target labels that are applied to each word individually, therefore accuracy or F1 score can be used in a straightforward manner to evaluate the generated sequence against the gold standard sequence of labels. However, in the sequence-to-sequence scenario, there are many occasions where a generated target sequence does not necessarily resemble exactly the target reference and nevertheless it could be considered a valid sequence (this will be more noticeable when explaining MT).

In the rest of this section we use the task of MT to formally introduce the Enc-Dec architecture (Section 2.3.1). We then describe the related work that uses this architecture to solve more generalized sequence transduction problems, such as language generation and structured prediction (Section 2.3.2).

2.3.1 Machine Translation

The general task of MT is defined as translating a source sentence $E = e_1, ..., e_{T_x}$ (e.g. English) into a target sentence $F = f_1, ..., f_{T_y}$ (e.g. French). To perform translation, a system with parameters θ must learn the probability of F given E, therefore its task is to find the target sequence F with the maximum conditional probability given a source sequence:

$$p(F|E) \propto p(E|F;\theta)p(F;\theta) \tag{2.7}$$

where $p(E|F)$ is the *translation model*, $p(F)$ is the *target language model* and the parameters θ are learned from data consisting of aligned sentences in the source and target languages (sentences that are translations of each other); this is what we call a **parallel corpus**. Note that a parallel corpus has translation pairs that can be used as a *reference* to learn plausibility of translations; however, as it is well known, there is not a unique translation of a sentence, therefore having a reference does not imply that it is the only possible correct translation.

A whole research area seeks to find the best ways to evaluate the quality of translations (and in general any sequence transduction problem). The most widely used metric for MT is BLEU (Papineni et al., 2002), which computes a score that measures the partial overlap between the generated translation and the reference by using weighted n-grams. Other relevant metrics are TER (Snover et al., 2006), METEOR (Denkowski and Lavie, 2011), and

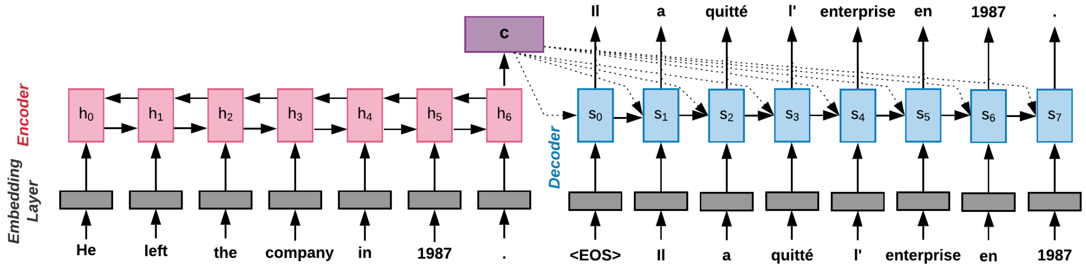


Fig. 2.4 The Encoder-Decoder architecture. Example when translating from English to French. A Bi-LSTM Encoder is used to obtain the context vector **c**. An LSTM decoder predicts the next word given **c**, the previous LSTM state and the previous word in the target sequence.

BERTscore (Zhang et al., 2020); however, as of the writing of this thesis, BLEU continues to be consistently reported on Enc-Dec research given that, so far, no metric has shown a clearly superior correlation with human evaluation on these tasks.

Before neural models emerged, statistical MT used phrase-based systems to approximate the translation model defined in Equation 2.7 by learning to factorize and weight the translation probabilities of pre-computed matching phrases in the source and target sentences (Marcu and Wong, 2002; Koehn et al., 2003). In practice, this was done with stand-alone sub-modules that were trained separately and ensembled in a pipeline to perform a translation.

Neural Networks started to be used as sub-modules for statistical MT, specifically to aid the computation of phrase probabilities (Schwenk et al., 2006) and later to obtain phrase representations as features for the statistical models (Cho et al., 2014). The first full-fledged NMT systems (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014) aimed to train a single large neural network that takes as input a source sentence and directly outputs a translation of that sentence into the target language, without the need for computing any extra features. This was the initial formulation of the Enc-Dec, where the two LSTMs are trained jointly to maximize Eq.2.7, and it works as follows:

The input sentence E (English) is transformed into a sequence of word-representations $\mathbf{x} = (x_1, ..., x_{T_x})$ through the embedding layer $\mathbf{E}_{src} \in \mathbb{R}^{|V_{src}| \times d}$. The **Encoder** takes this input sequence **x** and processes it. In general, the encoding step is defined as:

$$\mathbf{h}_j = f(\mathbf{x}_j, \mathbf{h}_{j-1}) \quad ; \quad \mathbf{c} = q(\{\mathbf{h}_1, \cdots, \mathbf{h}_{T_x}\}) = \mathbf{h}_{T_x} \tag{2.8}$$

where $\mathbf{h}_j \in \mathbb{R}^n$ is a hidden state at each time j of the source; f is frequently a Bi-LSTM network; and **c** is the source **context vector** that summarizes the information derived from the source hidden-states.

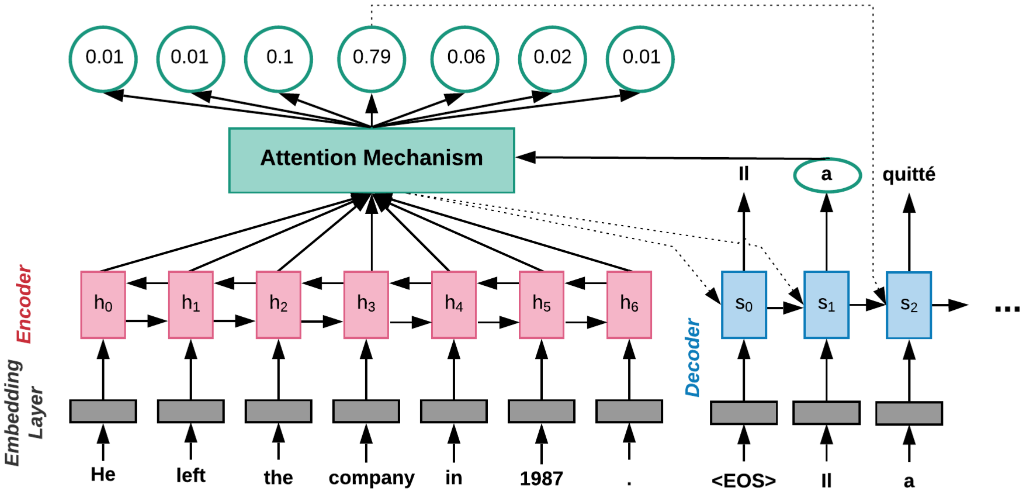


Fig. 2.5 The attention mechanism re-computes the context vector for each decoder time-step. This allows the Enc-Dec model to always *look back* at the entire source sequence, resulting in better performance, especially for longer sequences.

In the case of the output sequence F (French), it is also converted through an embedding layer $\mathbf{E}_{tgt} \in \mathbb{R}^{|V_{tgt}| \times d}$ into a target sequence of word representations $\mathbf{y} = (y_1, ..., y_{T_y})$. The **Decoder** is trained to predict the next word y_i given the context vector c as well as the information from the previously predicted target sequence such that:

$$p(y_i | \{y_1, \cdots, y_{i-1}\}, \mathbf{c}) = g(y_{i-1}, s_i, \mathbf{c}) \tag{2.9}$$

where y_{i-1} is the previously decoded token, s_i is the latest decoder hidden state (obtained with an LSTM[8]), and g is a MLP with a *softmax* layer that outputs a probability distribution over the target vocabulary from which the most likely next token is chosen.

The Enc-Dec as explained so far, however, still shows sub-optimal performance on longer sequences, even when using an LSTM. This happens because the encoder network is forced to compress all of the source information inside a single fixed-sized vector, resulting in loss of information that might lead to divergent or defective translations in such cases. To address this problem, a third component is proposed: the **attention mechanism** (Bahdanau et al., 2015; Luong et al., 2015). This mechanism allows the decoder, each time it generates a new target word, to (soft-)search for a set of positions in the source sentence where the most relevant information for the prediction is concentrated (Bahdanau et al., 2015). To obtain such an approximation, the decoder computation of each y_i is re-defined as:

[8] Note that this LSTM network is always uni-directional, since it is producing the target tokens from *left-to-right* in an auto-regressive manner, and since the probability of the next token is based on the previously produced tokens, the decoder cannot *look into the future*.

$$p(y_i|\{y_1, \cdots, y_{i-1}\}, \mathbf{x}) = g(y_{i-1}, s_i, c_i) \tag{2.10}$$

where s_i is the LSTM hidden state for time *i*, computed by

$$s_i = g(s_{i-1}, y_{i-1}, z_i) \tag{2.11}$$

notably with the addition of attention, the computing of the decoder state s_i takes into account a context vector c_i that is *re-computed* at every decoder time-step, whereas in the original decoder definition the context was always a fixed vector **c** that represented the entire source sequence.

The attention mechanism is the component that recalculates the context vector at each step. It is computed as a weighted sum of the source hidden states h_j, $j = 1, \ldots, T_x$:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \quad ; \quad \alpha_{ij} = \frac{\exp(e_{i,j})}{\sum_{k=1}^{T_x} \exp(e_{i,k})} \tag{2.12}$$

where

$$e_{ij} = a(s_{i-1}, h_j) \tag{2.13}$$

is the *alignment model*, which is meant to focus on the inputs around position j and the output at position *i* that are most relevant for the currently generated target tokens. In Bahdanau et al. (2015) the function a is a 1-layer MLP with a *softmax* layer on top which returns e_{ij} as the probability distribution denoting the importance of the source token j relative to the current target token i being decoded.

2.3.2 Other Sequence-to-Sequence Tasks

Given that the Enc-Dec architecture is a jointly trained network whose parameters are optimized based only on data without the need of specific features, it can be applied to any kind of problem that can be formulated as a mapping between source and target sequences, provided there is either access to big parallel data or a cheap way to produce *silver training data*[9]. An example of this occurs in language generation: having an Encoder to get an image representation and decoder that generates a novel caption based on it (Karpathy and Fei-Fei, 2017); or data-to-text generation, where abstract representations of data (for example from a knowledge base) can be transformed into a sentence in natural language (Chisholm et al.,

[9] The datasets that are produced with the assistance of automatic labelers and are not human-validated are called **silver data**. Since the labeling process is automatic, they contain noisy labels; however, this can be overcome by labeling bigger amounts of data and expecting the model to tune-out the noise.

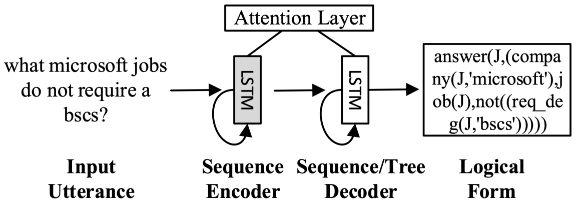

Attention Layer
what microsoft jobs do not require a bscs?
LSTM
LSTM
answer(J,(company(J,'microsoft'),job(J),not((req_deg(J,'bscs')))))
Input Utterance
Sequence Encoder
Sequence/Tree Decoder
Logical Form

Fig. 2.6 Dong and Lapata (2016) use a LSTM Enc-Dec network to map a sentence in plain text to its logical semantic form.

2017); or automatic summarization where a generated summary is decoded given an encoded large text (See et al., 2017).

In this line, a mapping from natural language text into structured sequences is also possible. Vinyals et al. (2015) demonstrate that constituency parsing can be formulated as a seq2seq problem by pairing the text to a *linearized* version of the parse tree. Following this approach, they use an Encoder-Decoder network with attention mechanism and only feed the network with large amounts of (sentence, parsed tree) pairs, without any extra feature set, and obtain results close to state-of-the-art. Similarly, the Enc-Dec framework has been used to aid the mapping of sentences to different semantic formalisms such as Semantic Parsing (Zettlemoyer and Collins, 2005) and Abstract Meaning Representation (AMR) Parsing (Banarescu et al., 2013). For example, Dong and Lapata (2016) built an Enc-Dec model for semantic parsing, where they map sentences to their linearized logical semantic form. This particular work demonstrated the surprising capability of networks for sequence transduction from text to deeply embedded hierarchical structures and preservation of balanced structures on the output, since parentheses proved to be well learned by the network. For AMR, Konstas et al. (2017) effectively constructed a two-way mapping: generation of text given an AMR representation (text to structured representation) and AMR parsing of natural language sentences (structured representation to text), again without relying on any external knowledge base or trained parsers, but only using parallel training data.

Finally, Zhang et al. (2017) went one step further by proposing a cross-lingual end-to-end system that learns to encode natural language (i.e. Chinese source sentences) and to decode them into sentences on the target side containing open semantic relations in English. This approach takes advantage of a high-resource language such as English with a high-quality parser to produce silver training data: first, they use an automatic parser to label English sentences and second, they translate the same English sentences to Chinese and end up with a parallel corpus of (Chinese, Parsed-English) pairs. They used this data to train a seq2seq model that directly learns to translate from a sentence in Chinese to the parsed English representation. The common denominator of the tasks described above is the use of seq2seq models with large amounts of silver data. This architecture has shown to be robust enough for learning even with noise in the training corpus, provided there is a cheap method to obtain a big amount of silver data that helps the model to generalize beyond the noise. Moreover, this architecture is very robust for cross-lingual tasks, as well as text-to-structure mappings, since it relies solely on parallel data to learn the patterns for the mapping, regardless of the complexity of the structure that needs to be decoded.

2.4 Multi-Way Machine Translation

Machine Translation is normally applied to a pair of languages, by modeling $p(E|F)$: the probability of a target sequence *F* conditioned on a source sequence *E*. However, this kind of mapping is strictly specific to a given language pair, where models learn to map words or sub-phrases from the specific source and target pair, presumably overfitting to the correlations of the two selected languages. In general, the basic assumption for trying models beyond the one-to-one translation is that, even when many languages differ lexically, they are closely related on the semantic level when working with parallel corpora.

It is not trivial to extend this mapping to work on multiple pairs of languages. The availability of multi-way parallel corpora[10] allows for the possibility to generalize MT to take into account more than a single pair of languages at a time. In their proposal for multi-source NMT (many-to-one translation), Zoph and Knight (2016) train a $p(E|F,G)$ model directly on trilingual data, using two source sentences (F,G) simultaneously as information for decoding a target sequence *E* and show positive BLEU score improvements over strong single-source baselines, especially when the two source languages are more distant from each other, showing that having information from multiple languages helps the generalization of the model to decode better sequences.

Conversely, Dong et al. (2015), avoid the issue of language pair-specific translation by training a multi-task system that performs one-to-many translation. This system learns to encode a source sequence *E* and decode either *F* or *G*; this is trained by alternating the target languages. The model then learns a *shared encoded representation* and *shared attention mechanism* that the decoder can use to condition the target language generation.

Firat et al. (2016a) generalize the previous approaches into a fully multi-way NMT system that can perform many-to-many translations. In this case, there are *N* language-specific encoders and *M* language-specific decoders. The system is trained to share the attention

[10] A *multi-way* corpus is such that contains parallel data across multiple languages, namely the same sentence has equivalent translation in *N* different languages, therefore one is not tied to work only with one source and one target reference pair at a time.

mechanism, this means that it learns a common continuous representation space that is shared by all $N \times M$ languages, demonstrating that training a system with all combinations of languages generalizes better. Moreover, the use of shared parameters across many languages has potential for zero-resource machine translation, in which there does not exist any direct parallel examples between a target language pair, but where the shared-attention mechanism is language-agnostic enough to empower a translation across a previously unseen language pair (Firat et al., 2016b).

On a parallel approach, Johnson et al. (2017) introduce a simpler method to translate between multiple languages by using a single model that is encompassed by only one *universal* encoder and one *universal* decoder, that can share all parameters end-to-end and process sentences from any language pair while improving translation performance for all languages involved. To do so, they add an artificial token to the input sequence to indicate the required target language, a simple amendment to the data only while keeping the same Enc-Dec model proposed for one-to-one NMT by Wu et al. (2016). This method has the additional benefit of directly improving lower-resource languages since all languages use exactly the same set of parameters and finally, it makes zero-shot straightforward since the model can recognize $N \times M$ languages to encode and decode without bi-text restrictions.

2.5 Transformer Architecture

As explained when discussing the LSTM-based seq2seq models, the general task is always to *encode* an input sequence, obtain one or more vector representations of the input and at the next stage use a second network to *decode* a target sequence given the input context and the so-far previously generated target sequence to predict the next token. This means that at each step the decoder is auto-regressive, consuming the previously generated symbols as additional input when generating the next. While this is certainly already a robust architecture, it can be optimized for more efficient processing. For example, by avoiding the constraint of compressing the whole input and only decode at a later stage; or decoding the sequences piece-by-piece in a left-to-right fashion; or looking at the input sequence only through an indirect attention mechanism.

The transformer architecture is an enhanced seq2seq architecture that bypasses the recurrence constraints of LSTMs and relies entirely on a fully integrated attention mechanism to draw global dependencies between input and output (Vaswani et al., 2017). Given that it does not rely on recurrence, it allows for significantly more parallelization (which results in execution time optimization), while at the same time augmenting the expressiveness of the learned representations by virtue of more and better interconnected parameters. This allows

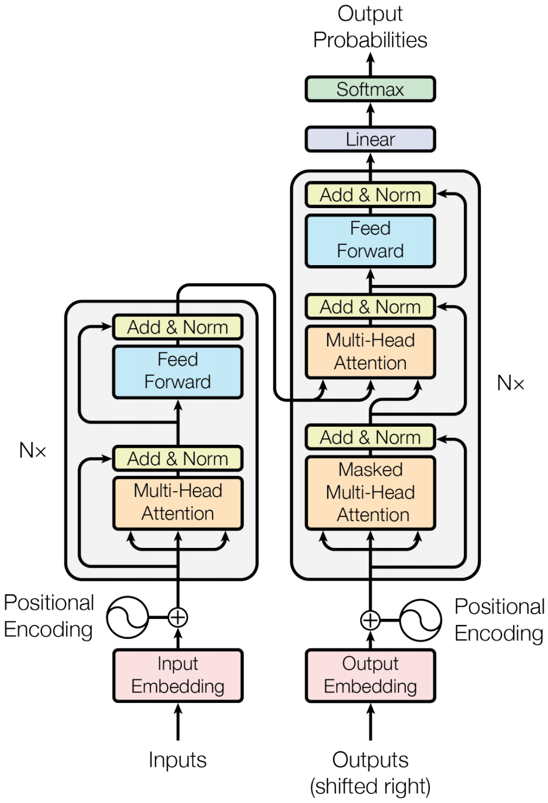


Fig. 2.7 The full transformer architecture as shown in Vaswani et al. (2017)

the transformer to learn from longer sequences, together with more complex correlations between input and output. It is worth noting, however, that the bigger size in parameters that transformers have, come at the drawback of needing considerably more training data to fully exploit the capabilities of the architecture, making it not always a suitable alternative to the LSTM-based seq2seq architecture.

The transformer architecture also includes an initial embedding layer to transform the discrete words into continuous vectors that represent them (they can also be initialized with the pre-trained word representations from Section 2.2.2). The subsequent layer is an **Encoder**, defined as a stack of N identical layers. Each layer has a multi-head self-attention mechanism[11], followed by a fully connected feed-forward network. Residual connections are employed (He et al., 2016) around each of the two sub-layers, followed by layer normalization (Ba et al., 2016).

[11] self-attention means that the encoder will *attend* the full input sequence itself for each processed input token. The attention mechanism is similar to the one explained in Section 2.3.1. It is called multi-head, since there are several copies of attention (each copy is a head) that can *focus* on different sections of the attended sequence.

The **Decoder** is, as the encoder, an N-stacked layered network, with the addition of a third sub-layer in each of the N components. This sub-layer performs multi-head attention over the input sequence representation (namely the output of the encoder stack), in the same way the attention in a LSTM seq2seq architecture does. Finally, another important difference is that the self-attention sub-layer in the decoder stack is masked to force it to only attend to previously decoded tokens, ensuring that the predictions for position i depend only on the known outputs at positions less than i. With these two additions, the Transformer fully imitates the behavior of the seq2seq models, with the advantage that there is no recurrence involved.

2.6 Contextualized Language Models

The continuous word representations described in Section 2.2.2 are computed to represent each word-type, and while the context is considered when computing the representation, once the word entries are learned, they are used at the word-level, regardless of the context. This is very convenient to represent words in an efficient way; however, it does not take into account the fact that the same word-type often has a different meaning depending on the context in which it appears. This is the well-known problem of word sense disambiguation in NLP. **Contextualized word representations** are intended to keep the context information within the word representation. This is obtained by training a neural network that captures correlations between each word-mention and its respective context. Unlike word-type vectors, which are essentially lookup tables that assign the same vector to any mention of a word-type, contextualized representations include both word-level and sentence-level information that *contextualizes* each word (all this expressed in the neural network parameters that are learned through diverse language learning tasks) (Smith, 2019).

Therefore, contextualized word representations were recognized as a tool not only for learning better representations of words, but as a potential tool for robust inductive transfer learning (Howard and Ruder, 2018) that could then be used as the base for training NLP models and avoid random initialization from scratch everytime that a new task needs to be learned, and as a result of this, improve SOTA results in different NLP tasks. Several general contextual LM architectures and training objectives were thus proposed to obtain these pre-trained representations such as GPT (Radford et al., 2018), ULMFiT (Howard and Ruder, 2018), ELMo (Peters et al., 2018), BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2020) among others.

Importantly, there are two strategies for applying pre-trained language representations to downstream tasks: **feature-based** and **fine-tuning**. In the first one, the idea is to train a deep neural language model and then extract the word representation vectors to use them as extra features for task-specific architectures; the latter is constructing a robust enough architecture where pre-trained word representations are further refined at training time directly to fit into specific tasks. This results in contextualized vectors that are also specialized in the tasks for which they were fine-tuned.

In some parts of this work we rely particularly on two contextualized word representations (namely ELMo and BERT), the latter being more used in detail, both as a source for word representations and as an architecture that we fine-tune for different task purposes. For these reasons we focus on explaining such architecture in detail.

2.6.1 ELMo

Embeddings from Language Models (ELMo) was the first model trained explicitly for capturing contextualized representations (Peters et al., 2018). This is a Language Model (LM) based on a multilayered Bi-directional LSTM. This means that the bi-directional network is trained to learn a forward LM, predicting the current token given the past context:

$$p(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} p(t_k | t_1, t_2, \ldots, t_{k-1}) \tag{2.14}$$

and a backward LM, predicting the previous token given the future context

$$p(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} p(t_k | t_{k+1}, t_{k+2}, \ldots, t_N) \tag{2.15}$$

Training both objectives on the same L-layer network ends up with word representations R_k for each word token x_k:

$$R_K = \left\{ \mathbf{x}_k^{LM}, \overrightarrow{\mathbf{h}}_{k,j}^{LM}, \overleftarrow{\mathbf{h}}_{k,j}^{LM} | j = 1, \ldots, L \right\} \tag{2.16}$$

where $\mathbf{h}_{k,j}^{LM}$ is the hidden state corresponding to word k at layer j. ELMo is then obtained as a linear combination of the intermediate layer representations in the biLM, thus capturing complex characteristics of word use such as syntax and semantics, and how these uses vary across linguistic contexts (i.e., to model polysemy).

2.6.2 BERT

Bi-directional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) is a particular architecture that can be used as a source for contextualized word vectors

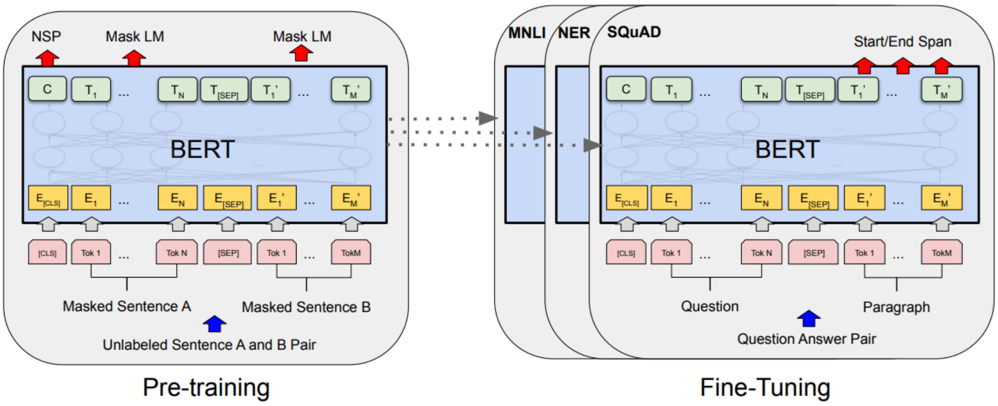


Fig. 2.8 The BERT architecture can be used for training a masked language model and, once it is pre-trained, the same weights can be used for fine-tuning on different down-stream tasks. Figure taken from Devlin et al. (2019)

(one can use the BERT pre-trained vectors as input features for an external architecture) as well as an architecture that can be entirely fine-tuned, where the BERT model itself can be further trained to solve specific NLP downstream tasks. BERT is based on the transformer architecture, which models the whole sequence context at the same time (as opposed to using Bi-LSTMs where the network is constrained to the token-by-token processing and a post-concatenation of left and right contexts), making it a robust architecture to produce context-informed predictions.

To allow looking at *both directions* at the same time, BERT is pre-trained as a masked language model. This means that for a given sentence, some of the tokens are randomly substituted by a $[MASK]$ token and the learning objective is to predict the word-type that was in the original sentence at that given position. In addition to this, the same architecture is also pre-trained to perform a *next sentence prediction* task, where given a sentence pair (S_1, S_2) the model makes a binary prediction to decide if S_2 is the following sentence of S_1 or not. These two tasks generate a robust architecture that models token-level as well as sentence-level relationships (See *pre-training* on the left side of Figure 2.8).

BERT is flexible enough to be fine-tuned as a classifier for several token-level and sentence-level complex tasks. For the sentence-level classification task, the architecture includes a special token $[CLS]$ that is used during training to hold the *class* of the encoded sequence. The architecture also includes the special token $[SEP]$ which acts as the separator to be able to encode different spans inside the input sequence (e.g. one can encode a question-answer pair by including a $[SEP]$ token between them). Some of the sentence-level tasks where BERT was fine-tined and presented SOTA performance are GLUE (Wang et al., 2018), Natural Language Inference (Bowman et al., 2015), and SQuAD (Rajpurkar et al., 2016, 2018). See *fine-tuning* on the right side of Figure 2.8. As for the token-level predictions, it is only necessary to add an output layer on top on the transformer to recreate state-of-the-art models sequence labeling results, for tasks such as Named Entity Recognition (Tjong Kim Sang and De Meulder, 2003).

2.6.3 Multilingual BERT

The pre-training of BERT is unsupervised, making it very easy to obtain results in a wide variety of languages, since the only prerequisite is to have written text available without any kind of labeled data: for the masked token prediction, one just randomly replace a word-token with the $[MASK]$ token; and for the next sentence prediction, one just needs to provide examples seen in the data as the positive class and any random pair as the negative class). This allowed to straightforwardly train a Multilingual BERT (mBERT) version. Surprisingly, the mBERT architecture is strong enough to learn in the same parameters simultaneously from different language pairs, specifically mBERT is trained on 102 languages, and has demonstrated SOTA performance also in many non-English NLP tasks.

Chapter 3

Related Work

Before neural networks became the norm, most semantic role labeling approaches relied heavily on lexical and syntactic indicator features. Through the availability of large annotated resources, researchers designed features and used a series of complex techniques, such as dynamic programming, or integer linear programming, to optimize global and local designed constraints and achieve high accuracy on the common datasets. However, results often fell short when the input to be labeled involved non-frequent predicates, or instances of infrequent syntactic linguistic phenomena and surface realizations, given that they did not appear frequently enough in the training data and didn't allow the statistical models to generalize well given the hand-crafted features.

Neural network components were added to SRL systems to overcome the feature-engineering bottleneck (FitzGerald et al., 2015; Roth and Lapata, 2016). One of the first examples of research in NLP shifting from purely featured-based approaches to neural network models is the neural multi-task system proposed by Collobert et al. (2011), where a single system (based on a Convolutional Neural Network) learned to solve several tasks without any explicit feature designed for them. Further, the first fully neural system specifically designed for the SRL task was proposed by Zhou and Xu (2015) who define it as a sequence labeling task, and train a a deep LSTM network as a semantic role labeler.

While eliminating language-specific features in principle could help a model to learn from data in any language, neural networks have shown to be successful on supervised tasks mostly when there is access to a large amount of high-quality annotated data. Only when this requirement is met, the network can exploit the patterns in the data obviating the need of features. This is the case of English SRL because the high-quality PropBank training corpus is large enough to give a strong signal to the neural model. Unfortunately, this is not yet the case for other languages, calling for methods to create more scenarios where neural networks can be used to improve performance such as data augmentation or joint multilingual systems.

For this reason, most of the neural semantic role labelers have been developed for English. Throughout this chapter we will first describe some of the most prominent semantic role labelers that give the current English SOTA results for PropBank annotations (Section 3.1). We will then explain the most common data augmentation methods used to enlarge datasets in a single language (Section 3.2) and also augmentation methods that make use of high-quality annotations in a source language (usually English) to label data in a lower-resource target language by cross-lingual label projection (Section 3.3). Finally, in Section 3.4 we describe more recent joint multilingual solutions that seek to train single polyglot models that leverage data from different languages at the same time with the aim of providing stronger training signals, especially to the lower-resource languages and thus improving the task results in those languages.

3.1 Neural Models for SRL

3.1.1 Neural Features for SRL

The use of neural components as an aid to obtain features that could be more resilient to infrequent phenomena not captured by the hand-designed features attracted interesting changes in how supervised tasks are approached, including SRL. In this line, Hermann et al. (2014) use pre-trained word embeddings as input features for a semantic frame identification classifier. Given a sentence and a marked predicate, they use a linear transformation to map the word embeddings of the predicate and predicate's children into a low-dimensional representation, where the frame labels are also embedded. Using this information, they train a ranker that assigns the most feasible semantic frame for the given predicate word. Following this idea, FitzGerald et al. (2015) use a feed-forward neural network that generates argument and role representations (related to their respective predicates) that are embedded in a shared vector space. With this, they skip the step of finding syntactic features and let the neural network automatically learn the correlations between predicates and arguments. The similarity of such learned representations can be measured by their dot product, and is used to score possible roles for candidate arguments by using a graphical model proposed by Täckström et al. (2015). This graphical model jointly models the assignment of semantic roles to all arguments of a predicate, subject to structural linguistic constraints. The original work used hand-crafted features, but FitzGerald et al. (2015) straightforwardly integrated the vector representations learned by the neural network as the input features and obtain even better results. Moreover, with this feature-free proposal, it is also possible to learn, in a single model, representations from different schemes such as FrameNet (Baker et al.,

1998) and PropBank (Palmer et al., 2005), as well as being able to work with span-based and dependency-based annotations. On all datasets, this model performed on par with the hand-engineered graphical model.

In a similar manner, Roth and Lapata (2016) propose to use lexicalized dependency path embeddings as features to better handle the problem of sparsity expressed in phenomena such as control predicates and sentences with rare dependency structures. In particular, this work aims to model the semantic relationships between a predicate and its arguments by analyzing the dependency path between the predicate word and each argument head word. It considers lexicalized paths, which are decomposed into sequences of individual items. For example, in the sentence *He had trouble **raising** funds.* one can automatically extract the dependency paths between the predicate and its arguments, resulting in the following paths:

(3) a. ***raising*** $\xrightarrow{NMOD}$*trouble*$\xrightarrow{OBJ}$*had*$\xleftarrow{SBJ}$*he*

b. ***raising*** $\xleftarrow{OBJ}$*funds*

In general, given a dependency path **x** with steps $x_k \in \{x_1, \ldots, x_n\}$, it is fed into an LSTM Encoder to obtain the representation $\mathbf{e}_n$. This representation is combined with a vector of binary features *B* through a linear hidden layer **h** which is then fed into the output layer **s** which computes the most probable class category for the given word w_n. This is expressed in the equation:

$$\mathbf{s}_c = softmax(max(0, \mathbf{W}^{Bh}\mathbf{B} + \mathbf{W}^{eh}\mathbf{e}_n + \mathbf{b}^h)) \tag{3.1}$$

The obtained vectors $\mathbf{s}_c$ are the features that are fed into the classic SRL system described in (Toutanova et al., 2008). The neural embedding features also proved to be transferable to other languages. In this case, Roth and Lapata (2016) report new SOTA for English and German and confirmed not only the flexibility but also the efficacy of neural features for SRL.

3.1.2 NLP (Almost) from Scratch

Collobert et al. (2011) propose a multi-task model that, given raw text inputs, learns several NLP tasks (e.g. Part-of-speech tagging, Chunking, Named Entity Recognition and SRL). This is done by first learning internal word representations based on vast amounts of mostly unlabeled training data (around 852 million words), instead of exploiting input features carefully optimized for each particular task. On the top of these learned representations, a deep task-specific network is trained on task-specific data to learn from the word representations as features and the labels at different linguistic level. A last Conditional Random Field (CRF) layer which is used to compute the most probable label for each token (thus each of the NLP tasks is treated as a *sequence labeling task* (See Section 2.2.4). The aim is two-fold: firstly, to evaluate the quality of the learned representations on each of the linguistically relevant tasks, and secondly, to completely bypass the feature engineering process for each task. This is also based on the intuition that the network should be powerful enough to infer all the intrinsic linguistic relations needed to solve each of the proposed tasks.

To standardize the sequence labeling for all tasks, they use the IOB (Inside Other Begin) notation, which marks relevant spans of text with the task-specific labels, for example:

(4) a. The account billed $ 6 million according to [Leading National Advertisers]$_{ORG}$
[The]$_{O}$ [account]$_{O}$ [billed]$_{O}$ [$6]$_{O}$ [million]$_{O}$ [according]$_{O}$ [to]$_{O}$ [Leading]$_{B-ORG}$
[National]$_{I-ORG}$ [Advertisers]$_{I-ORG}$

b. [The account]$_{A0}$ [billed]$_{V}$ [$ 6 million]$_{A1}$ according to Leading National Advertisers
[The]$_{B-A0}$ [account]$_{I-A0}$ [billed]$_{V}$ [$6]$_{B-A1}$ [million]$_{I-A1}$ [according]$_{O}$ [to]$_{O}$
[Leading]$_{O}$ [National]$_{O}$ [Advertisers]$_{O}$

where a. is labeled for NER and b. is labeled for SRL. Note that both tasks are initially span-based, and the IOB notation assigns one label per token without loosing the span ranges. This way it is possible to straight-forwardly treat each task as a standard sequence labeling. Importantly, for the case of SRL, the model processes one predicate-argument structure at a time. In order to have coherent IOB notation for complex sentences, they are repeated as many times as predicates there are inside it, and produce a label sequence for each predicate separately.

Specifically for SRL, this model reached 75.49 F1 score on the CoNLL-05 dataset, which is slightly below the SOTA models with hand-crafted engineering features for the task (which reported 77.92 F1 at that moment). Nevertheless, this work demonstrated the ability of deep neural networks to discover hidden representations from unlabeled data by only using a stochastic learning algorithm that scales linearly with the number of examples. More importantly, it demonstrated that the automatically learned representations are promising and strong enough to be transferred to downstream tasks in NLP within the same neural architecture, opening the path for experimenting with neural networks for end-to-end learning.

3.1.3 End-to-End BiLSTM Models for SRL

Zhou and Xu (2015) propose the first end-to-end neural system for SRL. This model takes only the original text as its *input features*, without any intermediate tag nor syntactic information, which then are processed by a deep bidirectional LSTM. At the top of the Bi-LSTM it locates a conditional random field (CRF) model for sequence label prediction.

Formally, the task is to predict a sequence of labels (or tags) $\mathbf{y}$ given a sentence $\mathbf{w} = \{w_0, \ldots, w_n\}$ and a predicate v. Each $y_i \in \mathbf{y}$ belongs to a discrete set of BIO tags $\mathbf{T}$, and $n = |\mathbf{w}| = |\mathbf{y}|$. Predicting the SRL structure becomes the task of predicting the most probable tag sequence over the space of all possible $\mathcal{Y}$:

$$\hat{\mathbf{y}} = \operatorname*{argmax}_{y \in \mathcal{Y}} f(\mathbf{w}, \mathbf{y}) \tag{3.2}$$

The word inside the sentence $\mathbf{w}$ is converted, through an embedding layer $\mathbf{E}$ into a sequence of word representations (which are pre-trained with a neural language modeling task). The model processes one token at a time $\mathbf{E}(w_i)$, to which 3 more features are concatenated: the word representation of main predicate w_j of the sentence $\mathbf{E}(w_j)$, a predicate context (the word representations inside the predicate's surrounding window) p_{ctx}, and a binary indicator p_{flag} (set to 1 if the current token to be processed is inside the predicate context-window or 0 otherwise). Thus the input for each layer l of the network at each time step is the following feature vector:

$$x_{l,t} = \begin{cases} [\mathbf{E}(w_i); \mathbf{E}(w_j); \mathbf{p}_{ctx}; \mathbf{p}_{flag}] & l = 1 \\ \mathbf{h}_{l-1,t} & l > 1 \end{cases} \tag{3.3}$$

The deep Bi-LSTM processes the whole sequence of inputs and finally the CRF layer computes the most probable tag sequence. This model, which only used those four simple features as input, obtains 81.07 and 81.27 F1-score on the CoNLL-05 and CoNLL-12 span-based datasets respectively, out-performing by an important margin the previous systems that were based on parsing results and feature engineering.

Later, He et al. (2017) also approach the span-based SRL task in a very similar way: A deep Bi-LSTM was trained for SRL as a sequence labeler; however, it introduced four improvements with respect to Zhou and Xu (2015)'s model:

- A simplified input layer (the input is only a word representation with a **binary predicate indicator**, $\mathbf{P} = 1$ if the token is a predicate or $\mathbf{P} = 0$ otherwise).
- Introduced **high-way connections**, meaning that they interconnect non-consecutive layers on the deep BiLSTM network.
- They used **recurrent dropout**, a technique where some of the input tokens are randomly masked (i.e. the word representation, or corresponding hidden state is set to 0), this *noise* mechanism acts as a regularizer to improve generalization.

- Instead of the CRF layer, it uses a constrained **decoding A* algorithm** with BIO constraints.
- It used an ensemble approach to improve further the latest SOTA.

In this case, the layer-specific inputs $x_{l,t}$ are:

$$x_{l,t} = \begin{cases} [\mathbf{E}(w_t); \mathbf{P}(t = v)] & l = 1 \\ \mathbf{h}_{l-1,t} & l > 1 \end{cases} \tag{3.4}$$

The Bi-LSTM is trained to minimize the following equation:

$$f(\mathbf{w}, \mathbf{y}) = \sum_{t=1}^{n} \log p(y_t|\mathbf{w}) - \sum_{c \in \mathcal{C}} c(\mathbf{w}, y_{1:t}) \tag{3.5}$$

where the first term is the negative log likelihood of the label sequence conditioned on the input, and the second term is an optional set of decoding constraints $\mathcal{C}$ (e.g. structural consistency, syntax constraint, etc), inspired by previous feature-based SRL models (Punyakanok et al., 2008; Täckström et al., 2015).

Since this model does not include a CRF layer, the A* decoding algorithm is the one that manages the sequence constraints (e.g. a inner tag *I-A0* cannot appear before a *B-A0* opening tag, or a single sentence cannot have two *A0* roles, etcetera).

This model was also tested on the span-based CoNLL-05 and CoNLL-12 PropBank test sets, obtaining, with the final ensemble of models, a F1 score of 84.6 and 83.4 which showed a considerable improvement over the SOTA at that moment.

For the case of dependency-based SRL, which was typically more linked to syntax given that only syntactic heads are annotated, Marcheggiani et al. (2017) propose a neural model that labels the syntactic heads by only relying on Bi-LSTM hidden states (encoder) without any syntactic supervision. An role classifier is applied on top of the encoder to predict the tag for each word in the sentence. As in previous work, one predicate-argument structure is predicted at a time, processing one sentence as many times as predicates it has.

In this work the word representation is the concatenation of four vectors: a randomly initialized word embedding $x^{re} \in \mathbb{R}^{d_w}$, a pre-trained word embedding $x^{pe} \in \mathbb{R}^{d_w}$, a randomly initialized part-of-speech embedding $x^{pos} \in \mathbb{R}^{d_p}$ and a randomly initialized lemma embedding $x^{le} \in \mathbb{R}^{d_l}$, which is processed by a 4-layer Bi-LSTM. The classifier is a log-linear model:

$$p(r|v_i, p) \propto exp(W_r v_i) \tag{3.6}$$

where v_i is the hidden state calculated by BiLSTM$(x_{1:n}, i)$; p refers to the predicate of interest and $\propto$ is the proportionality. This model is equivalent to the CRF layer used by Zhou and Xu (2015).

The authors reported their architecture on 4 languages of the CoNLL-09 Shared Task (Hajič et al., 2009), namely English, Chinese, Czech and Spanish. They slightly improved the SOTA at that time for those four languages: from 86.7 to 87.7 in English, from 79.4 to 81.2 in Chinese, from 80.2 to 80.3 in Spanish, and from 85.4 to 86 in Czech. Again, the biggest advantage of this model is that the neural model didn't need any lexico-syntactic features nor separately trained parsers to achieve these results.

3.1.4 Syntactically Informed Neural SRL

Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling. Even though syntax-agnostic neural architectures managed to surpass the results for feature-based systems, Marcheggiani and Titov (2017) experiment with a neural model that incorporates syntactic information using graph convolutional networks (GCNs) (Duvenaud et al., 2015; Kipf and Welling, 2017) and achieve better SOTA scores for English and Chinese.

They use initially the same Bi-LSTM encoder as in Marcheggiani et al. (2017) to obtain the word representations BiLSTM$(x_{1:n}, i)$, which are subsequently used as the initial layer of a k-layered GCN encoder. This encoder incorporates the syntactic tree of the same sentence by treating it as a labeled directed graph where $L(u, v)$ represents the syntactic label that relates word v (governor) to word u (dependent). For each word v, its hidden state $h_v^{(k+1)}$ for the next layer is computed as:

$$h_v^{(k+1)} = ReLU(\sum_{u \in \mathcal{N}(v)} g_{v,u}^{(k)}(V_{dir(u,v)}^{k} h_u^{(k)} + b_{L(u,v)}^{(k)})) \tag{3.7}$$

where each u is a word belonging to the syntactic neighborhood $\mathcal{N}$ of v; $g_{v,u}^{(k)}$ is a learned gating mechanism for each word pair, $h_u^{(k)}$ is the hidden state of each u word in the neighborhood, and $V_{dir(u,v)}^{k}$ and $b_{L(u,v)}^{(k)}$ are the learnable parameters.

By incorporating the syntactic information, the authors aim to profit from the syntactic information that is already present in the CoNLL-09 datasets. They find that syntax helps to slightly improve the scores from Marcheggiani et al. (2017) (by 1 F1 score approximately); however, because their model is overparametrized (the network has too many degrees of freedom, especially considering that the size of SRL training data are very small) the improvements were achieved only on the bigger datasets available: Chinese and English.

Linguistically-Informed Self-Attention for Semantic Role Labeling. For the case of span-based SRL, Strubell et al. (2018) explore the possibility of adding syntax information to a Transformer to focus its learning capacities for the SRL task. Their enhanced Transformer is called LISA: linguistically-informed self-attention. This architecture combines a multi-head self-attention mechanism with multi-task learning. The model is trained to i) jointly predict parts of speech and predicates; ii) perform syntactic parsing; and iii) use a dedicated attention head to attend to syntactic parse parents, while iv) assigning semantic role labels. To incorporate syntax, one self-attention head is trained to attend to each token's syntactic parent, allowing the model to use this attention head as an oracle for syntactic dependencies.

The input to the network is a sequence of token representations $X = \{x_0, \ldots, x_t\}$ initialized to pre-trained ELMo representations. These tokens are projected to representations of the same size as the output of the self-attention layers. The multi-head self attention consists of H attention heads, and the results of the H self-attentions are concatenated to form the final self-attended representation for each token.

As mentioned before, one head is specialized in syntax, this means that instead of computing the dot product of the key-query attention pairs (K, V), as done traditionally in the Transformer architecture (Vaswani et al., 2017), they score the compatibility between K_{parse} and Q_{parse} using a bi-affine operator (Dozat and Manning, 2017) (denoted as U_{heads} to obtain attention weights:

$$A_{parse} = \mathrm{softmax}(Q_{parse} U_{heads} K_{parse}^T) \tag{3.8}$$

this head is provided with auxiliary supervision (the syntactic heads of the training corpus) to model the probability of token t having parent q as:

$$P(q = \text{head}(t) | \mathcal{X}) = A_{parse}[t, q] \tag{3.9}$$

The last layers of the model use the transformer representations to predict semantic roles. Each transformer token representation t is projected to a predicate-specific representation t_{pred} and a role-specific representation t_{role}, which are provided to another bilinear transformation U_s for scoring. So, the role label scores s_{ft} of token t with respect to frame f (i.e. the predicate) are given by:

$$s_{ft} = (s_f^{pred})^T U s_t^{role} \tag{3.10}$$

The last layer computes a locally normalized distribution over role labels for token t in frame f using the *softmax* function. At test time, constrained Viterbi decoding is used to emit valid sequences of BIO tags.

This work also finds that performing multi-task learning and incorporating syntactic information is beneficial for neural models especially when a big training corpus with gold data is available. They tested their approach on the span-based CoNLL-05 (Carreras and Màrquez, 2005) and CoNLL-12 (Pradhan et al., 2012) for English and improved SOTA by 2 F1 points with respect to only using a syntax-agnostic transformer architecture (Tan et al., 2018). Unfortunately, this approach is also heavily parametrized, being difficult to apply to lower-resource languages.

As we have seen in this review of semantic role labelers, big improvements were made by focusing on neural architectures that get rid of the intensive task of language-specific feature-engineering. While in principle this should imply that the end-to-end neural models can be applied to any language, the evidence shows that the best performing neural models are those which have access to a big corpus that provides robust training signals (i.e. corpora annotated with gold part-of-speech, syntactic parsing, lemmas and semantic roles). We have seen that in the cases where this is not true, the feature-based models exhibit a better performance. We take this as evidence that there is still a need for generating more training data for lower-resource languages. In the following two sections we will go through the related work for monolingual data augmentation and cross-lingual annotation projection. Both techniques aim to alleviate the lack of big training corpora in lower-resource languages, which ultimately can be used as auxiliary to generate more training data and once a bigger corpus is available, perhaps a big parametrized neural model such as the ones that have worked for English can be also used to improve the results for SRL in languages other than English.

3.2 Monolingual Augmentation Methods

Above we showed the progress in deep neural approaches to SRL, which are exclusive for English. For other languages it was not possible to replicate such performance results given much smaller training resources. Therefore, now we will explain the work that aims to get more data for other languages. Because the manual construction of large-scale labeled corpora is an expensive process in terms of time, human and economic resources, several semi-automatic methods have been proposed. Any method which can help to reduce the manual effort involved with resource creation for new languages, as well as methods that can leverage unnanotated data as means of training signal, constitute an important step towards improving the performance of the SRL task.

There are augmentation methods at the monolingual level aimed at obtaining larger labeled corpora to train large models in a semi-supervised manner where the original annotated data is augmented with automatically generated sentences that are based on the original gold annotations. This can be achieved by obtaining e.g. more diverse syntactic realizations or coverage of uncommon predicate and role sets without the need of manually annotating these new sentences. In this line, Vickrey and Koller (2008) propose a joint system for sentence simplification and semantic role labeling that proves to be more robust across syntactic variations at inference time.

Fürstenau and Lapata (2009) developed an algorithm that augments a small number of manually labeled instances with unlabeled examples whose roles are inferred automatically via (monolingual) annotation projection. The projection is formulated as a generalization of the linear assignment problem. This method finds a role assignment in the unlabeled data such that the argument similarity between the labeled and unlabeled instances is maximized. Experimental results on semantic role labeling show that the automatic annotations produced by this method also result in performance improvements.

Woodsend and Lapata (2017) developed a method to automatically extract rules for rewriting from comparable corpora and bi-texts to generate multiple versions of sentences annotated with gold standard semantic role labels. They re-train a semantic role labeler with the official CoNLL-2009 benchmark dataset augmented with their *rewritten* sentences and show performance improvement.

More recently, Cai and Lapata (2019) use an LSTM-based semantic role labeler that is jointly trained as a sentence learner (it simultaneously learns POS tagging, dependency parsing and predicate identification) in order to use data without semantic role labels as additional training signal when learning the SRL task. This technique consistently improves the performance in Chinese, German and Spanish portions of the CoNLL-09 benchmark dataset.

3.3 Cross-lingual Annotation Projection

Beyond machine translation, parallel corpora can be exploited to relieve the effort involved in creating annotations for new languages, especially when the resources available for them are scarce or lower quality. One of the most common methods to automatically overcome scarcity of annotations on a different language is **annotation projection** (Yarowsky et al., 2001). The automatic induction of annotations is a common technique to take advantage of existing models and annotations in a resource-rich source language and transfer them to a lower-resource target language. Given that an overwhelming majority of NLP research is done in English, it is feasible to use its resources as a source of annotations and develop cross-lingual methods that allow to map all this existing knowledge into other languages in a reliable manner. Importantly, this technique relies on a pre-existent availability of a parallel corpus. A parallel corpus contains a translated equivalent sentence to the source into *N* target languages, and the key idea of annotation projection is to use this parallelism to transfer the information. It works as follows:

1. Given a pair of sentences **E** (English) and **L** (new language) that are translations of each other, obtain annotations for **E**. These annotations can be obtained in two ways: i) take an annotated source corpus and obtain high-quality translations of the source text to create the parallel corpus, ii) take an already available parallel corpus and use a high-performance source model to label the source side.

2. Generate reliable alignments from the source annotations to the target candidates. The level of these alignments can be at any granularity inside the text (words, phrase chunks, discursive units, etc.), but all of the alignments ultimately rely on the availability of word alignments, which are links between individual words of both sides that indicate translational equivalence.

3. Once the alignments $\mathbf{E} \to \mathbf{L}$ are obtained, the annotations from **E** are induced onto **L** by following the alignments.

4. The new labeled sentences can then serve as data for training a model for **L** that is independent of the parallel corpus.

In the specific case of transferring predicates and semantic roles, the annotation projection paradigm faces important challenges: firstly, the automatic alignment methods often produce noisy or incomplete alignments hence, when the annotated roles are span-based, it is sometimes impossible to completely recover source and target aligned word spans; secondly, on top of their overall translational equivalence, the semantic structure to be projected must be shared between the two sentences. Clearly, if the role-semantic analysis of the source sentence **E** is inappropriate for the target sentence **L**, simple transfer through alignments will not produce valid semantic role annotations on **L** (Pado, 2007).

Importantly, even when having a human translated parallel corpus and gold standard word alignments, there is a phenomenon present when translating from one language to another, namely **translation shifts**. This occurs because there are cases of idiosyncratic lexical preferences in two different languages or when free translation is used to better adapt to the target language use. For example:

(5) a. If Mr. Mason had used less derogatory language to *articulate* his amateur analysis, **would the water be quite so hot**?

b. Hätte Mason in seiner amateurhaften Analyse eine weniger abwertende Sprache verwendet, **würde er dann so tief in der Tinte sitzen**?

where both sentences are perfect translations from each other; however, the English predicate *articulate* disappears on the German sentence. More importantly, the English sentence uses a specific idiom that is then adapted to find an equivalent German expression, loosing the lexical correspondence to fully preserve the meaning on the target side.

In these cases there cannot be a method that solely relies on alignments, and subsequent filtering is needed to rule-out the implausible transfers into the target language. A naive approach to this problem could be to use translation dictionaries and discard all expressions that do not match, but by following this hard-matching approach we encounter again coverage problems, and ideally what we want is to capture the *meaning in context* of the translated expressions (and not only lexical matches) in order to obtain a larger amount of plausible target annotations. On the other hand, if one relaxes too much the matching constraints, the resulting target annotations will be very noisy (as we risk transferring labels that shouldn't be present in the target language).

As this is still an on-going area of research, below we list some of the most common approaches to effectively apply annotation projection to SRL.

3.3.1 Cross-lingual Annotation Projection of Semantic Roles

Padó and Lapata (2009) assess whether English semantic role annotations can be transferred successfully onto German, by using English-German parallel sentences taken from Europarl (Koehn et al., 2003). They automatically assign FrameNet labels by aligning constituency trees. The alignment of arguments inside sentences is treated as an optimization problem. They find that the two languages exhibit a degree of semantic correspondence substantial enough to warrant projection. Since the FrameNet annotations are span-based, they also tackle the problem of annotated semantic roles with arbitrarily long word spans (therefore mere word alignments are not enough to reliably transfer the labels). To tackle this, they construct semantic alignments between syntactic constituents of source and target sentences and formalize the search for the best semantic alignment as an optimization problem in a bipartite graph.

Their method works as follows: Consider each bi-sentence as a set of linguistic units: source $u_s \in U_s$ and target $u_t \in U_t$. A semantic alignment A between U_s and U_t is a subset of the cartesian product of linguistic units (i.e. $A \subseteq U_s \times U_t$). Provided with an optimal **A** and the role assignment function for the source sentence a_s, projection consists simply of transferring the source labels r onto the union of the target units that are semantically aligned with the source units bearing the label r:

$$a_t(r) = \{u_t || \exists u_s \in a_s(r) : (u_s, u_t) \in \mathbf{A}\} \tag{3.11}$$

The task then is to find such optimal semantic alignment **A** among the set of all admissible alignments $\mathcal{A}$. This is done by solving the following equation:

$$\mathbf{A} = \operatorname*{argmax}_{A \in \mathcal{A}} \prod_{(u_s, u_t) \in A} sim(u_s, u_t) \tag{3.12}$$

In this work, the word alignments, obtained with GIZA++ (Och and Ney, 2003), are used as the proxy for computing the similarity measure between source and target constituents, and in general this problem is solved as a *bipartite graph* optimization problem, where U_s and U_t are the two partitions of the graph G which is initially fully-connected. The task is then to find the subgraph G' which keeps both partitions connected while minimizing the weight of the connections in G' which is analogous to solving equation 3.12.

The constituent based projection works as follows: represent source and target sentences as constituent sets $U_s = \{C_s^1, C_s^2, \dots\}$ and $U_t = \{C_t^1, C_t^2, \dots\}$ respectively. Use the word overlap between c_s and c_t with Jaccard's coefficient (based on the word alignments) as the constituent-based similarity function. The overlap function is formally defined as:

$$o(c_s, c_t) = \frac{|al(c_s) \cap yield(c_t)|}{|al(c_s) \cup yield(c_t)|} \tag{3.13}$$

Since the Jaccard's coefficient is symmetric, in order to take into account the alignments in both directions (source-to-target and target-to-source) the mean of both directions is taken as the final similarity measure:

$$sim(c_s, c_t) = \frac{(o(c_s, c_t) + o(c_t, c_s))}{2} \tag{3.14}$$

On top of the similarity-based alignment they introduce three filters to reduce the noise of alignments:

- Forcing the alignments to be inside a given span (only keep the contiguous word alignments of a constituent).
- Only perform alignments of content words: adjectives, adverbs, or verbs, nouns; and avoid processing words for which the word alignment tool didn't assign an explicit alignment.

- Reduce the size of the target tree by only considering the set of *likely constituents* to be labeled which are the children constituents of some ancestor of the predicate. This heuristic is according to the findings of Xue and Palmer (2004).

To evaluate the semantic parallelism and accuracy of this transfer method, the authors have independently annotated the source and target side of 1,000 parallel sentences with FrameNet semantic roles. The performance was measured using precision and recall, treating the German annotations as gold standard. They found that about 72% of the time English and German sentences evoked the same frame and around 91% of the time they contained the same set of roles (for the subset of sentences already holding the same semantic frame). This demonstrated that for languages that are close such as German and English it is possible to transfer semantic frames provided that the appropriate filter for predicate equivalence are used. This work showed successfully that the semantic correspondences in a parallel corpus can be used as a means for generating labeled corpora in a target language without explicit annotations on the target side.

3.3.2 Cross-lingual Validity of PropBank

In a separate line of work, van der Plas et al. (2010) studied the feasibility of directly applying the English PropBank frame definitions (Palmer et al., 2005) into a different language, in this case French. They hypothesize that the level of abstraction of a well-defined semantic lexicon/ontology should be already cross-lingually valid. To prove this, they manually annotated 1,000 French sentences by directly using the guidelines from the English predicate and label definitions and use it as a gold standard for further experiments.

There are two key differences of this work with Padó and Lapata (2009): i) this is based on PropBank semantic frames, which means the projection of frames is predicate- and sense-specific as opposed to frame-specific; ii) it uses dependency parsing as support, instead of constituents, therefore this is dependency-based semantic labeling (only the head word of each argument holds the role label). The manual annotation proceeded as follows:

1. For each predicate they find in the French sentence, annotators translate it to English and look for it in the frame file to be able to label the predicate token with the right verb sense.

2. If the translated verb is not found then a dummy label is assigned. Since this implies a non-parallelism between the English frames and the French sentence, these examples will be discarded.

3. Argument identification is performed for the cases that already have a predicate frame assigned. The annotator is guided to select the heads of phrases during the annotation process and then decide if they correspond to a role in the corresponding English frame definitions.

4. After discussions and individual corrections, the *F1 agreement* scores are between 91% (predicates) and 95% (arguments). This indicates that the task is well-defined.

The task of labeling predicates proved to be more difficult than labeling semantic roles. The inter-annotator agreement was 59% for PropBank verb senses, but if they are measured using VerbNet classes it increases to 81%, meaning that translations of predicates occur to similar-enough related lexical entries. Disagreement was resolved by comparing annotations at the verb class level. Non-parallel cases (dummy labels) were found to be mostly due to idioms and collocations, they were discarded because they are translation shifts. Following this approach, they arrived to a corpus of 1,000 French sentences with high-quality labeling which can be used as a test set for larger-scale cross-lingual experiments using the PropBank framework.

3.3.3 Scaling-up Cross-lingual SRL Annotation

The research continued towards developing a method to automatically generate enough French PropBank-labeled data to train a labeler for that language (van der Plas et al., 2011). They construct automatically a large-scale parallel English-French corpus with semantic role annotations. To do this, they also use the Europarl corpus (Koehn et al., 2003). In this case, they train a joint syntactic and semantic role parser for English, then they obtain word alignments for each sentence with its French translation by using GIZA++ (Och and Ney, 2003). Separately, they train a French syntactic parser and finally they use the word-alignment information and the syntactic relationships in both languages to transfer the semantic roles from English to French.

To increase the quality of the annotations on the target side, they only consider the **intersection of alignments** as valid alignments. This means that given the parallel sentences $E \leftrightarrow F$, the English word x_E is aligned to the French word x_F if and only if $x_E \rightarrow x_F \wedge x_F \rightarrow x_E$ is true. Furthermore, to ensure a strong alignment quality, they adopt the **direct semantic correspondence** approach (Hwa et al., 2005), where a relationship transfer $R(x_E, y_E) \rightarrow R(x_F, y_F)$ is valid if and only if there exists a simultaneous valid alignment $x_E \rightarrow x_F$ and $y_E \rightarrow y_F$ (i.e. both the governors and dependents of the relationships are aligned). Likewise, the transfer of a semantic property $P(x_E) \rightarrow P(x_F)$ is made if and only if there exists a a valid alignment $x_E \to x_F$. The relationships in this case are semantic role dependencies and the properties are predicate senses.

Additionally, they propose the following filters to improve the alignment quality:

1. **Filter low-frequency:** remove a sentence pair if it contains a predicate sense with a low relative frequency given its word (relative frequency less than 0.2).
2. **PoS tags filters:** only keep sentence pairs whose predicates are aligned to a target POS noun, verb or adjective
3. **Avoid non-literal translations:** only keep sentence pairs with a source to target predicate alignment and at least one of the roles that have an alignment to the target.

Even though this careful design of annotation transfer ensures high-quality alignments, this comes at the trade-off of very low density of annotations on the target side. This was measured by comparing the automatically annotated sentences to 500 manually annotated French sentences (van der Plas et al., 2010), and resulted in F1 score of 55 for predicates and 65 for arguments, which is close to the inter-annotator agreement of 59 and 74 for predicates and arguments respectively. However, in practice the consequence of having low-density of annotations is that systems trained on this data assign scarce annotations and normally only identify the frequent predicates and roles seen in the construction of the corpus. On the other hand, this work is already an important step that shows feasibility of applying cross-lingual projections at large-scale for PropBank SRL annotations into a target language. More importantly, because the performance was measured using a human-annotated test set, this work shows that the automatic transferability of English PropBank labels is valid at least for related target languages.

3.3.4 Global Methods for SRL and Predicate Labeling

van der Plas et al. (2014) propose to learn a global transfer method for SRL and combine it with the direct transfer sentence-level method. The aim of the global method is to learn semantic relationships from the entire source corpus, as opposed to the direct transfer approach (such as the ones described above) where a token-by-token alignment is done only inside individual sentences. With a global approach they aim to obtain a more stable semantic representation across syntactically different sentences, address the translation shifts and word alignment noise problems that are present in the previously described approaches (Pado, 2007; Padó and Lapata, 2009; van der Plas et al., 2011), which are direct transfer methods. They use again Europarl (Koehn et al., 2003) and after filtering for sentence length

(40 tokens) and for avoiding translation shifts, they end up with a parallel corpus of 276,000 instances. In particular, two separate models that learn from this corpus are built: i) a model for predicate identification and labelling and ii) a model for semantic role assignment.

The transfer of predicates is modeled as a cross-lingual Word Sense Disambiguation (WSD) task and exploits information gathered from the whole corpus such as a bi-lingual lexicon and alignments obtained with using GIZA++ (Och and Ney, 2003). They compute the co-occurrence counts for alignments between English and French predicates, and given the probability they compute an association score for alignment, which they use to decide when and to which target tokens should the predicates be transferred to at inference time.

For the transfer of roles, the model determines the most suitable semantic role label r for a given argument of a given predicate p, based on its syntactic dependency label d. They compute the maximum likelihood estimates (MLE) and count occurrences of (p, d, r) triples, computed first in a large English corpus with gold semantic and syntactic annotated data and then applying that model cross-lingually to French .

This work finds that using the direct transfer method combined with the global knowledge improves coverage: 39$\%$ of predicates are recovered vs 29$\%$ from the direct transfer, and an F1 score of 45$\%$ which is an improvement over the global-only 42$\%$ and a big improvement over the direct transfer baseline which attained 37$\%$. As for the role labeling, only the accuracy obtained is reported which is 68 for the global method and 73 for the combined method, as opposed to the direct transfer baseline of 35.

This work shows that a global approach provides useful information for correcting and complementing the annotations from traditional direct transfer methods. Because direct transfer is a high-precision method, its combination with global methods (which are high in recall) improved previous results. The major advantage of the purely global approach is that it does not need parallel data or alignments at inference time (it uses the probabilities computed during training). However, as it was demonstrated in this work, the combination of direct and global transfer provides the best scores, loosing the advantage of the purely global method. Finally, while it is true that this approach generalizes better than the purely direct transfer, it is still restricted by the vocabulary and syntactic information found in the specific training corpus, a problem that can be straightforwardly addressed by neural methods.

3.3.5 Generating High Quality Proposition Banks for Multilingual SRL

In search of higher-confidence projection and denser annotations, Akbik et al. (2015) propose a combination of annotation projection and bootstrapping for refining the projections. Their proposed method is applied on English as a source language and a broad amount of target languages: Arabic, Chinese, French, German, Hindi, Russian, and Spanish. This work also claims to address better the translation shifts across the diverse languages.

Given a parallel corpus, the general steps followed in this work are:

1. Run a syntactic parser and semantic role labeler to assign PropBank labels to the source language S; in practice, they used the ClearNLP (Choi and McCallum, 2013) toolkit[1].

2. Assign syntactic labels to the target language T using a SOTA parser for each T.

3. **Filtered Projection:** first apply a *direct projection*(van der Plas et al., 2011), and then apply strict filters to the intermediate assigned labels, to only keep the high-confidence labels. This results in a T corpus with very low recall but very high precision.

4. **Bootstrap learning:** iteratively add new labels to the sentences by training classifiers for T and obtaining silver labeled data, improving coverage with each iteration.

Filtered Projection: The common errors obtained by the direct projection methods are analyzed in depth in order to design the constrains for developing the filters. They are inspired by the most common mistakes that were found during the error analysis and aim to either fix or avoid the projections that fall into one or more of the following cases:

- **Verb Filter (VF):** drops the transfer of a predicate if the word alignment tool gives a $Verb \rightarrow NonVerb$ alignment. This is to ensure that only verbal predicates are labeled on T (since only verbal predicates from S are being projected).

- **Translation Filter (TF):** aims to avoid translation shift errors on verbal predicates. Specifically, it is a *translation dictionary* that allows projection if and only if the T predicate is a valid translation of the S predicate (i.e. it is in the dictionary). The dictionary holds the *k* most commonly observed $s_i \rightarrow (t_0 \ldots t_k)$ translations.

- **Reattachment Heuristic (RH):** targets to obtain only syntactic head-to-head alignments. This means that when a source argument is lexically aligned to a non-head argument, a heuristic is used to move the T argument label to the ancestor which is the immediate child of the predicate.

The authors report that the **VF** filter increases the predicate precision from 45% to 59% without impact to recall (since the original aim is to only label verbal predicates) and argument precision from 43% to 53%, subsequently, applying the **TF** filter increases predicate precision to 88%, however, the recall impact is significant, dropping it to 71%.

[1] https://github.com/clearnlp

Note that the translation filter is very strict, since it is lexically constrained to the coverage of the dictionary, a consequence of this is that the filter might be blocking valid transfers of predicates, ending in low density of T predicates (and hence of arguments, since a missing predicate means the whole predicate-argument structure won't be present in T).

Note that other important sources of error that are not covered by the filters are: Gold Labeled data errors (some mistakes done during test set annotations), semantic labeler errors (noise from the *original* labels assigned by the SOTA systems), alignment errors (noise from the Berkeley aligner (DeNero and Liang, 2007) system), and parsing errors (noise form the SOTA – at that time – syntactic parsers).

Bootstrap Learning: Given that the constraints for filtering labels are very strict, the resulting data is scarcely labeled, but with very good quality. Since the parallel corpus used for the transfer is large enough, even with the low-density labels it is already possible to train an SRL system on the target language. Then it is possible to use this SRL to relabel the T corpus, effectively overwriting the projected labels with potentially less noisy predicted labels. This process is repeatedly applied until no further re-labeling is detected.

They train the SRL system of Björkelund et al. (2009). Since the precision of labels generated by the SRL system is lower than the precision of labels obtained from filtered projection, the precision of the training data is expected to decrease with the increase in recall. To optimize precision and avoid overtraining, the bootstrapping step is done for 3 iterations. They used as training data the Europarl (Koehn et al., 2003) and UN (Ziemski et al., 2016) parallel corpora. Its performance is tested for French with the manually annotated French corpus from van der Plas et al. (2010), and for the rest of the languages it is estimated by manually annotating 100 random sentences. This resulted in roughly 90 F1 points of predicates being correctly annotated and 70-80 F1 points for arguments (varying slightly across languages). For the assessment of the rest of T languages studied in this paper, the precision of predicate labels is over 95% and the recall is around 85%. For argument labels, the precision is at least 85% and the recall is between 66% to 83%.

Whereas the improvements of both precision and recall are impressive with respect to the hard-filter approach, the method is still bound to the existence of several resources that were used as a pipeline, namely there should be high-quality source and target syntactic parsers, big-enough parallel corpora, a well trained word aligner and curated lexical dictionaries for both source and target languages. On top of this, the coverage for predicates, that depends on fixed dictionaries, directly impacts the density of final annotations on the target side, especially if the method is used for out-of-domain data at inference time, calling for re-running this method each time one wished to apply it for different types of data.

On the other hand, this method was followed to generate two important (not manually annotated) resources for lower-resource languages, derived from this bootstrapping technique: the Universal Proposition Banks[2] datasets available for Chinese, Finnish, French, German, Italian, Portuguese, and Spanish, and a pre-trained open-source software, ZAP: An Open-Source Multilingual Annotation Projection Framework Akbik and Vollgraf (2018), that can be used out-of-the-box for Spanish, French and German. Particularly, we will use ZAP as a baseline when we evaluate the models that we present in this thesis.

3.3.6 Other Recent Approaches

Transferring Semantic Roles Using Translation and Syntactic Information. Aminian et al. (2017) define and use a customized cost function to train over noisy projected instances. This is shown to be an alternative to the manually-defined rules to filter projections. They propose to use bootstrapping following Akbik et al. (2015), however, the authors report that relabelling all training instances (including the already labeled data) instead of only labeling unlabeled raw data give better results on coverage of annotations on the target side..

They also introduce a weighting algorithm to improve annotation projection based on cues obtained from syntactic and translation information. For each aligned source argument s_i that is projected to a t_j, a cost function λ_i^{dep} is defined according to the dependency of the source and target words $dep(s_i)$ and $dep(t_j)$ as:

$$\lambda_i^{dep} = \begin{cases} 1 & \text{if } dep(s_i) = dep(t_j) \\ 0.5 & \text{otherwise} \end{cases} \tag{3.15}$$

With this the claim is to avoid translation shift projection and therefore improve quality during the boostrapping rounds. They report that by following this technique, they reach 63.8 F1 score (an improvement of 1.3 over the baseline) on role projection from English to German. On a closer look at projection, the authors report a precision of around 60% for the $A0$ role (the most frequent one in the dataset), while having a recall of 50% after 6 re-labeling iterations. Importantly, they ignore the projection of the modifier $AM-$ roles to German since this particular role does not appear in the CoNLL-09 German dataset. In contrast, one of the aims of our thesis is to include such role definitions in lower-resource languages such as German, this to have a more complete set of annotations that closely can follow English PropBank definitions.

From Raw Text to Semantic Roles. Aminian et al. (2019) propose a full method that performs first annotation projection from English to other target languages and uses this

[2] https://github.com/System-T/UniversalPropositions/

automatically created data to train a group of neural semantic role labelers. For the annotation projection step they assume a parallel corpus, perform word-alignments using GIZA++ (Och and Ney, 2003) to project the annotations.

The novelty of this approach is in the avoidance of any intermediate annotations such as supervised lemmas, dependency parse trees, and part-of-speech tags, they train a SRL system using the projected predicate-argument structures with two separate components: i) a joint argument identifier and classifier , and ii) a classifier for predicate sense disambiguation. They use BiLSTM encoders, and a role+predicate specific decoder that, instead of using explicit lower-level annotations (such as lemmas and POS), benefit from the encoded representations to learn the features in an unsupervised fashion.

For the predicate disambiguation they use an external system from Björkelund et al. (2009). For the joint argument classifier, given a sentence with n tokens and m predicates, m separate predicate BiLSTM encoders are run to extract contextualized representations for each token related to each predicate. The input of an encoder $\mathbb{E}$ for each token is the concatenation of a randomly initialized word embedding, a pre-trained word embedding, a character representation of the word (obtained by running a char-LSTM encoder on the token), and a predicate lemma embedding (active if the current token i is a predicate or a zero-vector otherwise). Therefore, each token i related to a predicate j is represented as:

$$x_{ij} = [x_i^{re}; x_i^{pe}; x_i^{char}; x_{ij}^{le}] \forall i \in [1, \cdots, n]; j \in [1, \cdots, m] \tag{3.16}$$

They use this as an input for each of the predicate-specific encoders $E_{i,j}$ and assign a label for each token inside the sentence.

The novelty of this work is that the cross-lingual transfer of dependency-based SRL annotations is end-to-end. This model is agnostic to linguistic features, as it is character-based, and can be trained on projected text on a target language without annotated data. The model achieved competitive performance compared to bootsrapping techniques; however, the evaluation was conducted on the Universal Proposition Banks (Akbik et al., 2015) which contains test data that was produced automatically also with bootstrapping techniques. It would be desirable to assess the effectiveness of this method with human-validated test sets. Moreover, the performance of the presented method is still poor in coverage, all the languages exhibit F1 projection scores around 60, which still shows that cross-lingual transfer methods based on word alignments hit a coverage limit and calls for an upgraded projection method that can improve coverage. We will show in Chapter 7 that our proposed projection method fulfills these two goals as we test it on a human-validated dataset and we avoid relying on pre-trained word alignments such as most of the methods presented here do.

3.4 Joint Multilingual Models

The models presented above follow, in general, a similar line of work: obtain an available parallel corpus, first using a monolingual SRL model on the source side and afterwards apply a cross-lingual method supported on lexical and syntactic information learned on the parallel sentences. On the contrary, the following models aim to use all the data available in several languages as a training signal since the beginning. The hypothesis is that if we are already using a semi-standardized label-set for different languages, the semantic roles should be applied to the parallel sentences. Even when this is not always the case, these commonalities should be enough to help a model from a lower-resource language to profit from the quality of labels from a high-resource language.

The final goal of a joint multilingual approach is to create models that obtain performance gains by improving the statistical strength for all languages (i.e. sharing parameters), which should benefit the semantic role labelers of resource-poor languages. The hypothesis is that multilingual models should optimize the information sharing across languages, instead of manually designing explicit alignments and filters that transfer labels from a source to a target. Importantly, the last two papers that we describe here, which are neural approaches (Sections 3.4.3 and 3.4.4), are concurrent work to what we present in this thesis.

3.4.1 Multilingual Semantic Role Labeling

Björkelund et al. (2009) propose a single multilingual approach that learns from all languages in the CoNLL-09 Shared Task (Hajič et al., 2009). They propose a pipeline of three independent, local logistic-regression classifiers that given a sentence S and a predicate p i) identify the predicate sense, ii) identify the arguments of the predicates, and iii) classify the arguments. They use the local models to generate a pool of candidates, which are then processed by a global re-ranker that applies a linear combination of the local classifiers' probabilities and a set of proposition features. The global re-ranker chooses the best sense and set of labels for each (S, p) pair.

To address the multilingual nature of the data, they implemented a feature selection procedure that obtains the best feature-set for each individual language, obtaining important gains over a generic set of features. To work with the label divergences across languages they made some special adaptions for what to consider core labels: in Catalan and Spanish, all the labels prefixed by $A0$ to $A3$; in Chinese and English, only the labels $A0 - A4$; and in Czech, German, and Japanese all the labels were considered core labels.

This work already showed the potential of training systems that optimizes the task for several languages at the same time. It reached the second place at the SRL-only CoNLL-09

Shared Task (Hajič et al., 2009), and SOTA for Chinese and German. Unfortunately, in the years that came after this, very little work was done on exploiting the multilinguality of the task, and aside from German and Chinese, no further progress was made on the performance for non-English languages.

3.4.2 Bootstrapping Semantic Role Labelers from Parallel Data

Kozhevnikov and Titov (2013) aim to facilitate the construction of SRL models for resource-poor languages, while preserving the annotation schemes designed for each target language. They propose a co-training of two monolingual SRL models (they use the model from Björkelund et al. (2009)), which are initially trained on monolingual data. Next, they aim to use information of both models to learn a role correspondence model (RCM) on a $(English, Target)$ parallel corpus[3]. Initially, the parallel corpus is annotated with semantic roles using the independent monolingual models, and then they use the RCM to refine these annotations via a joint inference procedure. The refined predictions are expected to propagate the superior information quality from English to the weaker language, thus improving the initial predictions.

The task of the RCM is to jointly use the source model f_s and target model (conditioned on the source) f_st scoring information, given the source S_s and target S_t sentences, and source p_s and target p_t predicates to identify the target language role assignment r_t that maximizes the objective:

$$L(r_t) = \lambda_t f_t(r_t, S_t, p_t) + \lambda_{st} f_{st}(r_t, r_s, p_s, p_t) \tag{3.17}$$

where $r_s = \operatorname{argmax}_r f_s(r_s, S_s, p_s)$ is the role assignment of the source-side arguments as predicted by the monolingual model and λ are the weights associated with the models.

For training the monolingual models they used the CoNLL-09 datasets (Hajič et al., 2009), for the parallel corpus, Europarl (Koehn et al., 2003), and GIZA++ for obtaining source-to-target alignments. Concretely, they evaluate their model on four language pairs: English (EN) vs German (DE), Spanish (ES), Czech (CZ) and Chinese (ZH), using the CoNLL-09 test sets. Consistent improvements are observed over a self-training baseline (this is, re-training the SRL monolingual model with the Europarl sentences of the corresponding language). For example, EN-CZ self-training has an accuracy of 62.15 vs 63.11 using the joint model (+0.96); whereas German self-training yields 68.34 vs 70.13 using the joint approach (+1.79). This work shows again that using data from different languages as training signal can yield improvements on the SRL task. Importantly, the authors also report that

[3] Where the English source is the better informed language and the target is a lower-resource language.

iterating more than one time through the joint re-labeling, ends in poorer performance for all languages, which shows that there is too much noise from the automatically labeled data and this is limiting the generalization capabilities across languages of this approach. Another important aspect that hampers improvement is the domain mismatch between the monolingual training data and the cross-lingual dataset, which affects the performance of models' predictions, as reported by the authors.

3.4.3 Polyglot Semantic Role Labeling

Mulcaire et al. (2018) try the straight-forward approach of training a single neural architecture using data from CoNLL-09 (Hajič et al., 2009). They only combine the languages bi-lingually (English + a lower-resource language). This is inspired by a successful neural multilingual dependency parsing model (Ammar et al., 2016) that uses the universal dependencies corpus (Nivre et al., 2016) to train a single model, and achieves better results in several languages compared to their monolingual baselines. In this work, the authors re-implement the span-based SRL model of He et al. (2017), and try three different modifications for language combination during training:

Simple Polyglot Sharing. Use pre-trained multilingual embeddings in the first layer of the model and train it on the union of data from two languages. Because English data is considerably bigger than the rest of languages in the CoNLL-09 corpus, they use stratified sampling to give the two datasets equal effective weight during training.

Language Identification. Concatenate a language ID vector to each multilingual word embedding and predicate indicator feature in the input representation. This vector is randomly initialized and updated in training. These additional parameters provide a small degree of language-specificity in the model, while still sharing most parameters.

Language-specific LSTMs. In addition to the language ID vector and processing every example with a shared biLSTM as in previous models, train language-specific 2-layer biLSTMs only on the examples belonging to one language. Each of these language-specific biLSTMs are stacked on top of the shared deep-biSLTM that is used in the two previous variants. The aim is to give the model a greater parameter space to learn both language commonalities and language specificity at the same time.

The authors compare their polyglot variants to a monolingual baseline (using the system of He et al. (2017) only with each language-specific training data), and to SOTA neural systems. They find inconsistent results for each language, where the first variant is the best for German and Japanese, the second variant is the best for Czech and Spanish, and the third variant is the best for Catalan and Chinese. Importantly, the monolingual baselines are quite low as compared to the SOTA of each language, this is understandable since such a big neural system can't learn a strong-enough signal when trained with the small datasets of non-English languages. For example, the German SOTA F1 score is 80.10, the monolingual baseline reported is 66.71 and the best polyglot (English+German) yields 69.97, which is an improvement from the baseline but still quite far from the SOTA. The authors also find that the improvements are mainly due to the core-roles that are both more frequent and also shared across languages, namely $A0 - A4$, which indicates the importance of having more reliable training data for lower-resource languages.

The fact that the polyglot approach works better than the monolingual baselines for lower-resource languages shows that neural models benefit from having more training data and from the English higher-quality labels; the neural architecture manages to generalize and transfer that knowledge into the weaker language. On the other hand, the fact that the polyglot model didn't outperform the monolingual SOTA shows the weaknesses of the incompatibility of annotations present in the CoNLL-09 datasets[4] (a finding that we will also analyze in our own work, see Chapter 6), such as a partial share of predicate senses and role labels, lower density of annotations in non-English languages, and fewer exposure to different sentence realizations in the non-English training sets.

3.4.4 Syntax-aware Multilingual Semantic Role Labeling

Inspired by the performance gains that more recent models for English obtained (Marcheggiani and Titov, 2017; Strubell et al., 2018) by leveraging syntactic information and enhanced word representations within a neural architecture for SRL, He et al. (2019) propose a language-independent neural model for dependency SRL that leverages syntactic information present in the individual training sets for each language in the CoNLL-09 dataset. With this technique, and with the help of multilingual contextualized representations (Che et al., 2018; Devlin et al., 2019), they improve the SOTA for the seven languages included in that dataset. This model is language-independent in the sense that the rules extracted for syntactic pruning depend only on the language-specific training data (as opposed to handcrafted rules, where the pruning might not work for all languages); the multilinguality of the model comes from the replacement of the embedding layer, either with the language-specific pre-trained embeddings or the multilingual contextualized embeddings, without changing anything else in the architecture. However, the model does not leverage SRL data from different languages at the same time, despite being called *multilingual*.

Concretely, they adapt the deep BiLSTM Encoder of Cai et al. (2018) and add an additional layer that integrates a novel method for pruning argument candidates guided by

[4] As we mentioned earlier, their work is inspired in a previous successful approach (Ammar et al., 2016) that uses Universal Dependencies, which, unlike SRL, is a completely standardized dataset across languages.

language-specific syntactic rules. As a last layer, they implement a biaffine scorer (Dozat and Manning, 2017) for the tasks of argument identification and classification. For each sentence and predicate pair, the model encodes at each time-step the concatenation of five vectors: i) a randomly initialized word embedding, ii) lemma embedding, iii) part-of-speech embedding, iv) pre-trained word embedding and v) predicate-specific indicator embedding (whether the current token is a predicate or not). The pruning rule is based on the well known property from the feature-based SRL models (Gildea and Jurafsky, 2000; Xue and Palmer, 2004), that the distances between predicate and its arguments on syntactic tree are within a certain range for most languages. Inside each language-specific training set, for each predicate p and their labeled arguments a_i, the distances from p and each a_i to their nearest common ancestor is calculated and saved as a tuple. Only the top-k frequent distance tuples (d_p, d_{a_i}) are kept as probable for argument labeling. During encoding, after the BiLSTM processed the whole sequence, the argument pruning layer drops the hidden representations corresponding to the tokens that do not appear as top-k candidates for the predicate of interest, and only the rest is processed by the last layer of the network.

The biaffine scorer takes as input the BiLSTM hidden states of predicate h_p and candidate arguments h_{a_i} filtered by the argument pruning layer. It computes the probability of the corresponding semantic labels using a *biaffine transformation* defined as follows:

$$\Phi_r(p, a_i) = (h_p)^T \mathbf{W}_1 h_{a_i} \mathbf{W}_2^T (h_p; h_{a_i}) + \mathbf{b} \tag{3.18}$$

where ; represents concatenation operator, $\mathbf{W}_1$ and $\mathbf{W}_2$ are the weight matrices of the bilinear and the linear terms respectively, and **b** is the bias item.

This model, when used together with pre-trained multilingual BERT representations (Devlin et al., 2019) in the embedding layer, improves the SOTA for the 7 languages in the CoNLL-09 dataset. In most cases it improves the F1 score by a considerable amount such as Catalan (80.3 to 85.1), Chinese (84.3 to 86.42), Czech (86 to 89.66), Japanese (78.2 to 83.76) and Spanish (80.5 to 84.6), however, in English and German the improvement was less significant (80.1 vs 80.9) for German and (90.4 vs 90.8) for English.

This architecture primarily reflects two aspects of neural SRL: namely that syntactic-based argument pruning still helps in neural models and that the strong lexical information encoded in the contextual representations are a big boost for improving SRL performance, even when the training data is smaller.

Chapter 4

An Extensible Model for Semantic Role Labeling

Thus far, we have described how end-to-end neural models considerably improved the state-of-the-art results for SRL in recent years. The majority of these neural models treat the problem of SRL as a supervised sequence labeling task, using deep architectures that assign a label to each token within the sentence. Given the complexity of these architectures, most of the described improvements are limited to English, for which resources are more plentiful and of higher quality. Even in those cases where training resources for another language exist, they are more restricted both in terms of the number of sentences, as well as in terms of the diversity of annotated predicates and arguments. Moreover, the available resources are not compatible with each other, because they were produced independently.

We also listed various works that address the data scarcity problem for non-English languages. Annotation projection in particular is a widely used method for augmenting both monolingual and bilingual data. However, such methods rely on preexisting parallel corpora, and on a pipeline of statistical models such as word aligners, a source semantic parser as well as source and target syntactic parsers, each of which introduce noise and propagate errors. Importantly, the filtering measures taken to reduce noise tend to be overly cautious, limiting the amount of data that can be produced. In addition to requiring parallel data and relying on automatic parsers and noise reduction strategies, annotation projection methods are designed to work for one language pair at a time. Hence, the need for improved solutions to the issue of resource scarcity for languages other than English.

In this thesis, we present a model that seeks to be a unified and extensible solution to overcome the issues listed above. The general solution that we propose is to simultaneously translate and transfer the labels by using a single joint model that accomplishes both things within the same step. In order to do this, we propose to reformulate the SRL task as a seq2seq task. Our approach is based on related work on two different tasks: *low-resource NMT*, which has shown the positive impact on target predictions by adding more than one language during training (Zoph and Knight, 2016; Johnson et al., 2017; Firat et al., 2016a), and *structured prediction* using seq2seq models (Dong and Lapata, 2016; Zhang et al., 2017), where a text input is mapped into a structured output. Additionally, it is directly related to concurrent work on *joint multilingual labeling* models (Mulcaire et al., 2018; He et al., 2019) where a single architecture is proposed to solve the same task for different languages.

We divide our effort to solve SRL as a seq2seq task and address resource scarcity in three stages of experimentation: monolingual, multilingual and cross-lingual.

In Chapter 5, we introduce the seq2seq formulation of SRL and a corresponding Enc-Dec model. We analyze how well such an architecture performs in a classical English **monolingual setting** by benchmarking the proposed system against existing monolingual SOTA sequence labeling models for SRL on well-known labeled evaluation data (the CoNLL-05, CoNLL-09, and CoNLL-12 datasets).

After establishing the feasibility of seq2seq SRL in a monolingual setting, Chapter 6 describes enhancements to the Enc-Dec model that let us fully exploit the architecture on languages other than English, and make the architecture flexible enough to apply it in multilingual and cross-lingual scenarios. Specifically, to experiment with non-English data, we take two lower-resource languages, German and French[1], and show how the availability of data impacts the performance of the same architecture in the same task as compared to English, the high-resource language. To overcome the performance gap for the lower-resource languages, we define the **multilingual setting** as the combination of data from more than one language to train a single model. Note that in this setting, we are not translating; instead, we are exploiting the data available in different languages. We do this by concatenating all the available training data from more than one language, i.e. as a data augmentation technique, and using the seq2seq model as a multilingual labeler. For example, if the model receives an English sentence as input, it generates a labeled English sentence; whereas if it receives a French input then it generates a French labeled sentence.

Furthermore, we define the **cross-lingual setting** as the joint task of generating a labeled sentence which is in a different target language with respect to the source input: in this case the model receives an input sentence in English and can translate and produce a labeled sentence in German or in French. Our architecture can learn to translate input sentences to another language while transferring semantic role annotations by leveraging machine translation parallel data and making use of existing cross-lingual SRL datasets for training.

[1] We are aware that these languages are not generally perceived as *low-resource* in NLP; however, since we are comparing the data availability and task performance to English, the lower performance in the task for these languages calls for the availability of more resources, thus we consider them here as lower-resource languages.

Importantly, the cross-lingual setting poses a complex evaluation scenario that was not present on the SRL (monolingual and multilingual) labeling task. Since we are proposing a model that translates and labels at the same time by using a generative decoder, we have to evaluate this setting as a data generation task. Here, the generated target sequences at inference time can only approximate reference targets, hence we lack defined gold data to compare our labeled outputs with. We define intrinsic and extrinsic evaluation scenarios for the cross-lingual outputs and also we include human-evaluation to assess the quality of the novel generated labeled data. With this, we show that our model can be used for augmentation of labeled data on the lower-resource languages.

Finally, we assess the difficulties that emerged by training and evaluating our cross-lingual model with the so-far available datasets. Even when we found our Enc-Dec to be robust enough for generating useful labeled sentences, we hypothesize that having more homogeneous data across languages can result in higher-quality multilingual and cross-lingual learning. Moreover, we managed to train our cross-lingual model with a limited parallel labeled corpus, thus we can benefit from defining a method for obtaining more parallel training data for cross-lingual SRL. For this reason, we explore alternatives to bilingual annotation projection for generating our own high-quality cross-lingual resources. Based on the latest advances in machine translation and multilingual contextualized language models, we propose a portable semi-automatic method for creating more parallel labeled training data that allows us to obtain uniformly labeled datasets across languages while minimizing human intervention in the data creation process. Our method for creating training data uses existing SOTA Machine Translation to generate high-quality parallel data and the multilingual BERT language model (mBERT) to emulate word alignments and annotation projection without the need of training specific bilingual transfer models. Additionally, we create manually validated test sets for German, Spanish and French in order to experiment and explore the dataset we obtained by following our method.

Chapter 5

Encoder-Decoder Architecture for SRL

5.1 SRL as a Sequence-to-Sequence Task

In this chapter we give the definition SRL as a seq2seq task; we then demonstrate and evaluate the architecture in a monolingual setting. We start with the monolingual setting for the sake of both simplicity and comparison with SOTA results. In this setting, the source and target sequences are always in the same language. We use the datasets that were used to establish the SOTA for the SRL neural sequence labeling approaches (Collobert et al., 2011; Zhou and Xu, 2015; He et al., 2017) which are the span-based datasets for English, CoNLL-05 (Carreras and Màrquez, 2005) and CoNLL-12 (Pradhan et al., 2012).

We propose a straight-forward implementation of a seq2seq Enc-Dec model with attention (Bahdanau et al., 2015) to perform SRL. Our model learns to map an unlabeled *source* sequence of words $\mathbf{S} = (w_1 ... w_{|S|})$ into a *target* sequence $(\mathbf{T} = y_1 ... y_{|T|})$ consisting of both word tokens and SRL label tokens (see Figure 5.1), therefore, even when source and target are the same sentence in lexical terms, $|\mathbf{S}| \neq |\mathbf{T}|$ because the target sequence contains the additional tokens that describe its SRL structure (See Table 5.1).

The source sentence, represented as a sequence of dense word vectors $(x_1 ... x_{|S|})$ obtained through a word embedding layer, is fed to a multi-layer Bi-LSTM encoder that produces a series of hidden states that represent the input. The decoder then uses this information to recursively generate target tokens y_i (which can be a word or a label) step-by-step, conditioned on the source, by attending the encoder's hidden states as well as the so-far generated tokens.

Because we are dealing with monolingual data, our architecture also includes a copying mechanism (Gu et al., 2016). This helps the model avoid lexical deviations in the output without taking away the freedom to generate words and SRL labels based on the context. The attention-based generation and copying mechanism compete with each other so that the model learns when to copy a token directly from the source and when to generate the next

Source-1:	The trade figures *<PRED>* **turn out** well , and all those recently unloaded bonds spurt in price .
Target-1:	*(#* The trade figures *A1)* *(#* **turn out** *V)* *(#* well *A2)* , and all those recently unloaded bonds spurt in price .
Source-2:	The trade figures turn out well , and all those recently *<PRED>* **unloaded** bonds spurt in price .
Target-2:	The trade figures turn out well , and all those *(#* recently *AM-TMP)* *(#* **unloaded** *V)* *(#* bonds *A1)* spurt in price .
Source-3:	The trade figures turn out well , and all those recently unloaded bonds *<PRED>* **spurt** in price .
Target-3:	The trade figures turn out well , and *(#* all those recently unloaded bonds *A1)* *(#* **spurt** *V)* *(#* in price *AM-ADV)* .

Table 5.1 A single sentence with three labeled predicates is converted into three different source-target pairs. The symbol *<PRED>* in each source marks the predicate for which the model is expected to generate a correct predicate-argument structure.

token. This is a big aid for the monolingual SRL task since it helps the model converge faster by learning to copy the source words into the target sentence and to generate the appropriate labels interleaved in the target.

5.1.1 Sequence Linearization

At the start of the process, we convert each of the SRL structures of every sentence into a linearized sequence that can be processed by the Enc-Dec architecture as a target sequence corresponding to the source sentence in plain text. To ease the linearization process, we restrict role labeling to a single predicate per sentence. If a sentence has more than one predicate, we create a separate copy for each predicate as in Collobert et al. (2011); Zhou and Xu (2015) and most subsequent work. In addition, for each sentence copy, the predicate whose roles are to be labeled is preceded by a special token *<PRED>* which marks the position of the predicate under consideration (see Table 5.1). This helps the decoder focus on generating the argument labels that related only to that specific predicate. Therefore, if a sentence has n_p predicates we process the sentence n_p times, each one with its corresponding predicate-argument structure. Because this process is entirely reversible, we can convert the system outputs back to the original format and then compute the results using the official CoNLL-05 Shared Task evaluation script.

5.1.2 Vocabulary

We assume a unique vocabulary that is shared by the encoder and decoder. The vocabulary comprises the N most frequent words occurring during training[1], the out-of-vocabulary token, and the special symbol used to mark the position of the predicate, thus $\mathcal{V} = \{v_1, ..., v_N\} \cup \{UNK, < PRED >\}$. In addition, we employ a set $\mathcal{L} = \{l_1, ..., l_M\}$ with all the possible labeled brackets and a set $\mathcal{X} = \{x_1 ..., x_S\}$, a per-instance set containing the S words from

[1] This is determined separately for each of the different training datasets.

the current source sequence being processed by the model. Thus, our total vocabulary is defined for each instance as $\mathcal{V} \cup \mathcal{L} \cup \mathcal{X}$.

The label set $\mathcal{L}$ contains one common opening bracket *(#* for all argument types to indicate the beginning of an argument span, and several label-specific closing brackets, such as *V)* which indicates the span comprises the main predicate or *A1)*, which indicates in this case that the span for argument *A1* is ending, and so forth for the rest of the role labels available in each dataset.

5.2 Monolingual Encoder-Decoder for SRL

5.2.1 Encoder

We use a two-layer Bi-LSTM encoder that outputs a series of hidden states $h_j = \left[\overrightarrow{h_j}; \overleftarrow{h_j}\right]$ where each h_j contains information about the context surrounding the word x_j. We refer to the complete matrix of encoder hidden states as **M**, since it acts as a memory that the decoder can use to attend or copy words directly from the source.

5.2.2 Attention

We use the global dot product attention mechanism from Luong et al. (2015) to compute the context vector c_i as:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \quad ; \quad \alpha_{ij} = \frac{\exp(e_{i,j})}{\sum_{k=1}^{T_x} \exp(e_{i,k})} \tag{5.1}$$

where $e_{i,j}$ is the dot product function between decoder state s_{i-1} and each encoder hidden state h_j. The decoder uses this mechanism as a soft-matching support that learns during training which source words are the most relevant when generating each target token.

5.2.3 Decoder

Our decoder is a single-layer recurrent unidirectional LSTM with copying mechanism (Gu et al., 2016) that emits an output token y_t from a learned distribution over the vocabulary at each time step *t* given its state s_t, the previous output token y_{t-1}, the attention context vector c_t, and the memory **M**. To get this distribution it is necessary to compute two separate modes: one for generating and one for copying.

To obtain the **probability of generating** y_t we use the context vector produced by the attention to learn a score ψ_g for each possible token v_i of being the next generated token. We define ψ_g as:

$$\psi_g(y_t = v_i) = W_o[s_t; c_t], \quad v_i \epsilon \mathcal{V} \cup \mathcal{L} \tag{5.2}$$

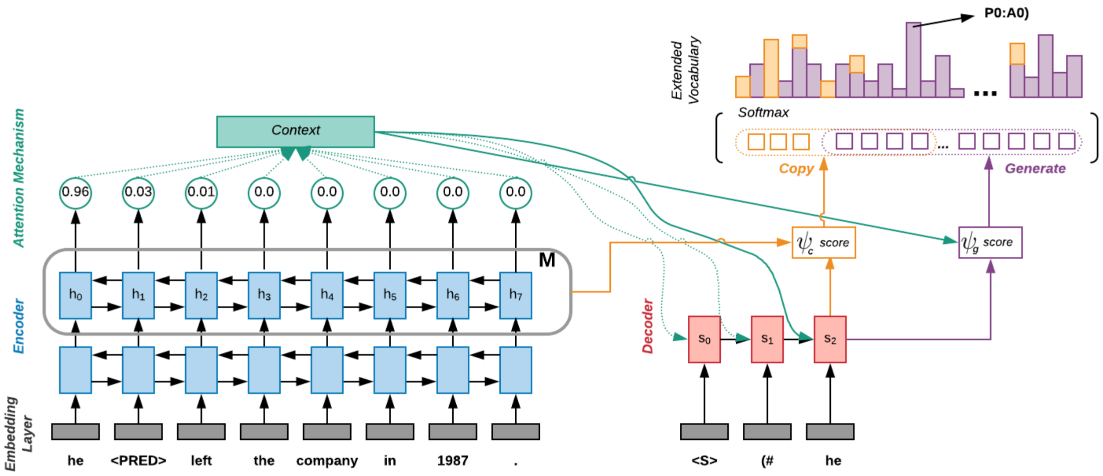


Fig. 5.1 Our sequence-to-sequence model for SRL. A score for copying and a score for generating tokens is computed at each time step and a joint softmax determines the probability of the next token over the extended vocabulary of words $\mathcal{V}$, labels $\mathcal{L}$ and current instance words $\mathcal{X}$.

where $W_o \epsilon \mathbb{R}^{N \times 2d_s}$ is a learnable parameter and s_t, c_t are the current decoder state and context vector respectively. This means that the model computes a generation score for both words and labels, based on what it is attending on at the current step.

For the **probability of copying** y_t we compute the score ψ_c of copying a token directly from the source as:

$$\psi_c(y_t = x_j) = \sigma(h_j^T W_c) s_t, \quad x_j \epsilon \mathcal{X} \tag{5.3}$$

where $W_c \epsilon \mathbb{R}^{d_h \times d_s}$ is a learnable parameter, h_j is the encoder hidden state representing x_j, s_t is the current decoder state, and σ is a non-linear transformation; we used `tanh` for our experiments.

Having access to these two scoring methods, the decoder has two competing modes: the generation mode, used to generate the most probable subsequent token based on attention; and the copying, used to choose the next token directly from the encoder memory $\mathbf{M}$, which holds both positional and content information of the source. A final mixed distribution is calculated by adding the probability of generating y_t and the probability of copying y_t:

$$p(y_t|s_t, y_{t-1}, c_t, \mathbf{M}) = p(y_t, \mathbf{g}|s_t, y_{t-1}, c_t) + p(y_t, \mathbf{c}|s_t, y_{t-1}, \mathbf{M}) \tag{5.4}$$

We use a *softmax* layer to convert the two scores into a joint distribution that represents the mixed likelihood of generating and copying y_t. Again following Gu et al. (2016), we define this as:

$$
\begin{aligned}
p(y_t, \mathbf{g}|\cdot) &= \begin{cases} \frac{1}{Z} e^{\psi_g(y_t)} & y_t \epsilon \mathcal{V} \cup \mathcal{L} \\ 0 & otherwise \end{cases} \\
p(y_t, \mathbf{c}|\cdot) &= \begin{cases} \frac{1}{Z} \sum_{j:x_j=y_t} e^{\psi_c(x_j)} & y_t \epsilon \mathcal{X} \\ 0 & otherwise \end{cases}
\end{aligned}
\tag{5.5}
$$

where Z is the normalization term shared by the two modes, $Z = \sum_{v\epsilon\mathcal{V}} e^{\psi_g(v)} + \sum_{x\epsilon\mathcal{X}} e^{\psi_c(x)}$. Since a single *softmax* is applied over the copying and generating modes, the network learns by itself when it is proper to copy a word from the source and when it needs to generate a label.

5.3 Experimental Setup

5.3.1 Training

Since we process as many copies of the sentence as there are predicates, the final amount of sequence pairs available for training is approximately 94 thousand for CoNLL-05 and 185 thousand for CoNLL-12 training sets. We keep linearized sequences up to 100 tokens long and lowercase all tokens. Because of this limit, we omit 30 (CoNLL-05) and 900 (CoNLL-12) sequences from training. Additionally, we initialize the model with pre-trained 100-dimensional GloVe embeddings (Pennington et al., 2014), which are updated during training.[2] All the tokens that are not covered by GloVe or that appear less frequently than a given threshold[3] in the training dataset are mapped to the UNK embedding. Our vocabulary size is set to $|\mathcal{V}| \approx 20K$ words for CoNLL-05 and $|\mathcal{V}| \approx 18K$ words for CoNLL-12.

During training, the objective is to minimize the negative log-likelihood of the target token y_t for each time-step for both generate mode (given previous generated tokens) and copy mode (given source sequence X). We calculate the loss for the whole sequence as:

$$
loss = -\frac{1}{T_y} \sum_{t=0}^{T_y} \log P(y_t | y_{<t}, X) \tag{5.6}
$$

We use the Adam optimizer (Kingma and Ba, 2014), a learning rate $l_r = 0.001$ and gradient clipping at 5.0. Both encoder and decoder have hidden layers of 512 LSTMs. We

[2] We experimented with word2vec word embeddings (Mikolov et al., 2013a) but found GloVe6B (trained on Wikipedia2014+Gigaword5) embeddings to perform better. Available at `https://nlp.stanford.edu/projects/glove/`

[3] We used a threshold of 10 for CoNLL-05 and 15 for CoNLL-12.

	CoNLL-05			CoNLL-12	
	Dev	WSJ	Brown	Dev	Test
Seq2seq (attention-only)					
same length	29.19	29.98	32.24	-	-
brackets	95.25	94.93	94.24	-	-
Seq2seq (w/ Attention & Copying)					
same length	96.71	97.15	97.24	97.46	96.07
brackets	99.91	99.82	99.88	99.97	99.93

Table 5.2 Quality of reproducing words and SRL brackets with seq2seq: Attention-only vs. Attention & Copying, on CoNLL-05 and CoNLL-12 datasets: percentage of correctly reproduced sentence length and percentage of balanced brackets.

use dropout (Srivastava et al., 2014) of 0.4 and train for 4 epochs with batch size of 6. The fast convergence of the model is due to the copying mechanism rapidly adapting to the fact that source and target contain always the same words, and that the decoder is basically learning to generate the label-tokens interspersed among the content words.

5.4 Monolingual English Results and Error Analysis

5.4.1 Effect of Copying Mechanism

To assess the effect of the copying mechanism on our outputs, we train a model using attention only, and compare performance with a model that uses both attention and a copying mechanism. Notably, the attention-only model learns to properly generate balanced brackets without further constraints, meaning that every opening bracket has a corresponding closing bracket within the sequence. This suggests that tasks using a similar notation as ours, or different SRL formalisms, could be easily extended to use a Enc-Dec architecture in a similar fashion as ours. However, due to the decoder's generative nature, many target sequences produce repetitions or diverge from the source in both length and token sequences. For example:

(6) a. **Source:** Some say november.
Target: *(#* some *A0) (#* say *V) (* november A1) .
Sys Out: *(#* some *A0) (#* say *V) (* say hi A1) .

b. **Source:** Now, the network has **opened** a news bureau in the hungarian capital.
Target: *(#* now *AM-TMP)* , *(#* the network *A0)* has *(#* opened *V) (#* a news bureau

A1) (# in the hungarian capital *AM-LOC)* .

Sys Out: *(#* now *AM-TMP)* , *(#* the network *A0)* has *(#* opened *V) (#* a news bureau in the soviet capital *A1)* .

This behavior is expected, since the system has to learn to generate not only the labels at the correct time-step but also to re-generate the complete sentence accurately, and part of the model's generalization includes lexical divergences. While this is a disadvantage compared to the sequence labeling models where the words are already given, it also shows a potential use of Enc-Dec to generate labeled sentence variations if intended to be used in this manner.

By adding the copying mechanism, the model avoids such unwanted divergences and successfully regenerates the source sentence in the majority (up to 99%) of cases, as shown in Table 5.2. Such behavior also enables us to measure the performance of the model as an argument role classifier against the gold standard, because we are forcing to reproduce exactly the same lexical tokens and a one-to-one word mapping can be assumed. Thus, we can benchmark its labeling performance against previous neural architectures built to solve the SRL task.

5.4.2 Semantic Role Labeling Results

Table 5.3 shows the overall labeling performance of our copying-enhanced seq2seq model in comparison to previous neural sequence labeling architectures. For sequences that do not fully reproduce the input, we cannot compute appropriate scores against the gold standard. We compute two alternative scores for these cases: *oracle-min*, by setting the score for these sentences to 0.0 F1, and *oracle-max*, by setting their results to the scores we would obtain with perfect (= gold) labels. With these scores, we can better estimate the loss we are experiencing by non-perfectly reproduced sequences (see Table 5.2.)

As seen in Table 5.3, our model achieves an F1 score of 76.05 on the CoNLL-05 development set, and 73.4 on CoNLL-12 using *min-oracle* (min) evaluation, and 77.29 and 75.05 with *max-oracle* (max) evaluation, respectively. While these scores are still low compared to the latest neural SRL architectures, they are above the relatively simple model of Collobert et al. (2011). Note also that in contrast to the stronger models of FitzGerald et al. (2015); Zhou and Xu (2015) and He et al. (2017), our architecture is very lean (only 2-layer encoder and a single layer decoder) and does not employ explicit structured prediction constraints (e.g. Conditional Random Field or A* decoding), to impose on the label assignment. Our simplified architecture allows us conclude that an Enc-Dec such as the one used for MT can be straightforwardly applied without further SRL specific additions.

	CoNLL-05				CoNLL-12	
	dev	test	WSJ	Brown	dev	test
Collobert	72.29	74.15	-	-	-	-
FitzGerald	78.3	-	79.4	71.2	79.2	79.6
Zhou & Xu	79.55	81.27	82.84	69.41	81.07	81.27
He	81.6	81.6	83.1	72.1	81.5	81.7
Ours (min)	76.05	76.7	78.13	66.28	73.4	73.61
Ours (max)	77.29	77.87	79.23	68.39	75.05	75.43

Table 5.3 F1 measure for argument role labeling of our seq2seq model w/ Attention & Copying on CoNLL-05 and CoNLL-12 dev and test sets, compared to Collobert w/o parser, FitzGerald single model, Zhou & Xu, and He single model .

5.4.3 Error Analysis

We analyze the output of the proposed model for a deeper investigation of the kind of errors that an unconstrained seq2seq model makes when performing SRL. The analysis is performed specifically on the CoNLL-05 development set.

Argument Spans. The model needs to generate labeled brackets at the appropriate time-step, that is, the prediction of correct spans for arguments. To evaluate performance of this function, we measure how much overlap exists between the generated spans and the gold ones. This is equivalent to computing unlabeled argument assignment. We found that 77.5% of the spans match the gold spans completely, 21.2% of spans are partially overlapping with gold spans, and only 1.2% of the spans do not overlap at all with gold.

Argument Labels. Recall that, even with the copying mechanism, our model is labeling the sentences as in a translation task. Every generated target token is *conditioned* on the source and the past. It learns to use information from relevant words in the source sequence, aligning the labels to the argument words via learned attention weights as it is shown in Figure 5.2. This means we can examine what the model attends to when generating the labeled bracket.

The confusion matrix in Figure 5.3 shows predicted vs. gold labels for all correctly assigned argument spans (i.e., the spans that match the gold boundaries). We observe that the model does very well for *A0* and *A1* gold roles, and that it produces only few mis-classifications for *A2*. However, it frequently predicts core argument roles *A0–A3* for non-argument roles, and also tends to mix predictions among non-core arguments. Since *A0* and *A1* roles are most frequent in the data, this suggests that the seq2seq model would benefit from more training data, particularly for less frequent roles, to better differentiate them. Notably, this kind of error is more prominent for spans that start with prepositions.

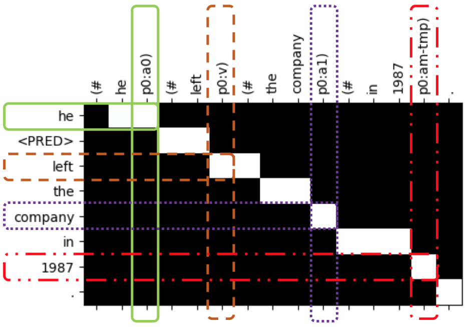


Fig. 5.2 Example of the alignments learned by the attention mechanism.

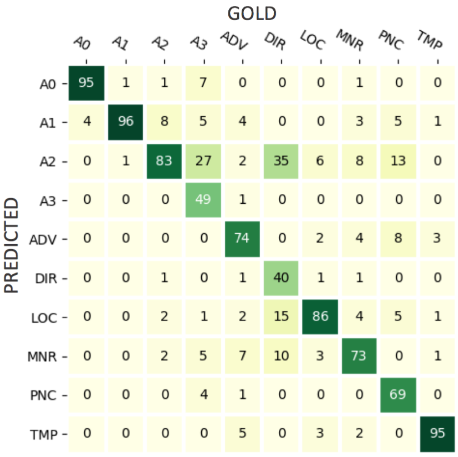


Fig. 5.3 Confusion matrix showing percentage of predicted labels compared to the gold labels on the CoNLL-05 development set.

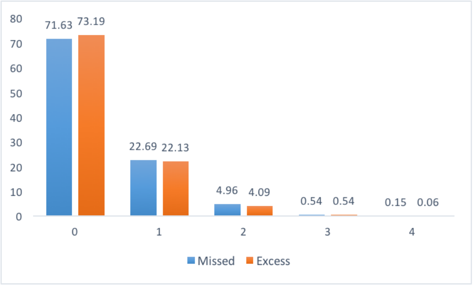


Fig. 5.4 Percentage of sentences with 0,1,2 or more missing (blue) or excess (orange) arguments (seq2seq w/Copying, CoNLL-05, dev set).

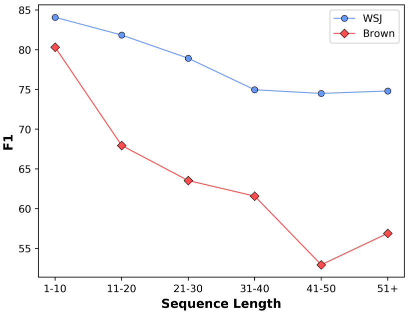


Fig. 5.5 Performance of the model based on the number of tokens that the sequence has.

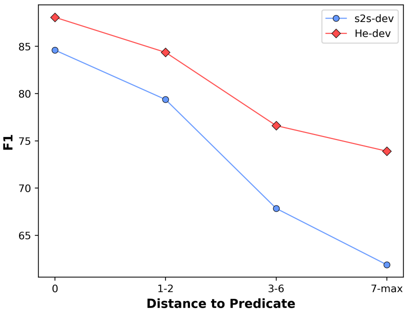


Fig. 5.6 F1 score of arguments in buckets of increasing distances from their predicate, with distance normalized by sentence length (CoNLL-05, dev). We compare our model with He et al. (2017).

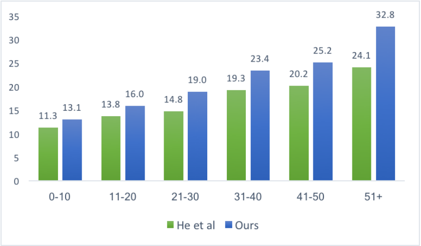


Fig. 5.7 Error ratio of arguments in different regions of the sequences (CoNLL-05, dev).

Role co-occurrence and role set constraints. Despite the absence of more refined decoding constraints, our model avoids generating duplicated argument labels in most of the sequences. We find duplicated argument labels in less than 1% of the sequences. Figure 5.4 shows that for the majority (about 70%) of sentences, there are no missing or excess arguments; about 24/20% of sentences experience a single missing/excess role, and only 5/4% of the sentences experience a higher amount of missed/excess roles. Overall, missed vs. excess arguments are balanced.

Sequence Length. The seq2seq model encodes both words and labeled brackets within a single sequence. This increases the length of the sequences that need to be processed. It is a well known problem that sequence length affects performance of recurrent neural models, even with the use of attention. To measure the labeling performance difficulty experienced by our model with increasing sequence length, we partition the system outputs in six different bins containing groups of sentences of similar length (see Figure 5.5). As expected, the F1 score degrades proportionally to the length of the sequence, especially in sentences with more than 30 tokens.

Distance to predicate. He et al. (2017) show that the number of labeling errors is proportional to the surface distance between the argument and the predicate. In our model, the distance between argument words and the predicate is even longer because of labeled brackets embedded in the sequence. Figure 5.6 displays the F1 score for different token distances between predicate and the respective argument. We see that the seq2seq model follows the same trend as the sequence labeling model, despite the fact that our model has access to the hidden states from the encoded input sentence; however, the real distance between predicate and argument in the decoder is also bigger.

Distance from sentence beginning. With each token that the model generates in decoding, the distance to the end position of the encoded sentence representation grows. While intuitively we would expect the model performance to degrade with larger distance to the input, it is also true that the model could be more prone to making mistakes at the beginning of the sequence, when the decoder has not yet generated enough context. To investigate this, we traced the ratio of errors that occur in several ranges of the sequence. We can see in Figure 5.7 that the first intuition was correct, the distance to the encoded representation is proportional to the mistakes that the model makes. We compare the error ratio to He et al. (2017) and show that the seq2seq system follows a similar trend but, understandably, degrades faster with sequence length.

Chapter 6

Multilingual and Cross-lingual Models for Semantic Role Labeling

Having established a stable formulation of SRL as a seq2seq task and achieved promising results on English monolingual datasets, we propose an enhancement and generalization of the Enc-Dec architecture, in order to leverage SRL data for multiple languages simultaneously. In this chapter, we detail the additions made to the model to successfully apply it for the target lower-resource languages. We describe experiments using English, German, and French data, and show that training a single multilingual model on all the available data in different languages yields a significant improvement in the lower-resource languages without overly affecting labeling performance for English.

Moreover, we experiment in a cross-lingual setting where our model functions as a generator of unseen labeled sentences by creating translations into lower-resource target languages (German and French) with interleaved SRL labels; as opposed to (monolingual) SRL labeling, where we were copying the source words and producing suitable labels (cf. Chapter 5). The biggest advantage of training a seq2seq generator in this manner is that we avoid the need for preexisting parallel corpora as well as explicit syntactic or semantic annotations at inference time, which are resources needed for label projection techniques. Next, we show that it is possible to augment the training set of a lower-resource language with sentences generated by our cross-lingual model and obtain improved F1 scores on the benchmark datasets, which leads us to conclude that the data generated by our model has enough quality to be used as training data. We close the chapter with an assessment of the achieved improvements and evaluate the strengths and weaknesses of our joint approach for translating and labeling SRL data.

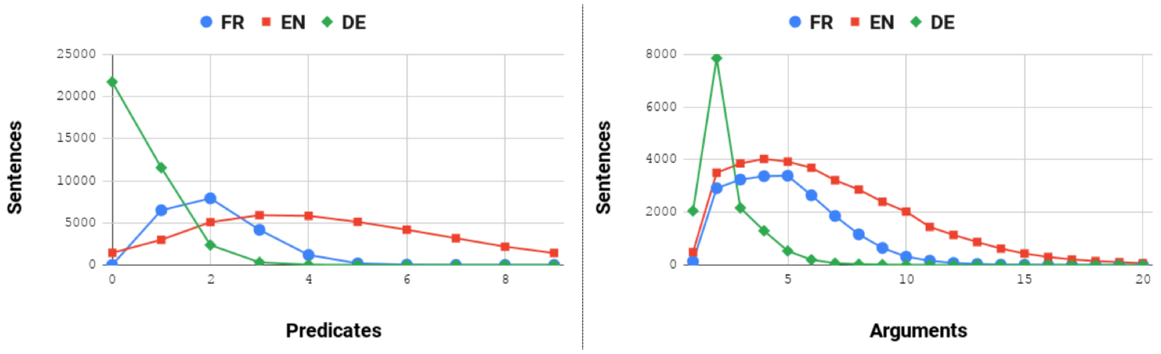


Fig. 6.1 Distribution of predicates (left-hand side) and arguments (right-hand side) in the three languages that we study here. We can see that non-English data has considerably less propositions per sentence and fewer argument labels.

6.1 Multilingual Semantic Role Labeling

In Chapter 5 we tested our seq2seq model (monolingually) on two English span-based SRL datasets (CoNLL-05 and CoNLL-12) and obtained results that, while having respectable performance, unfortunately fell short from the SOTA. In this section, we present enhancements to the Enc-Dec model which make it more robust, reaching SOTA scores for English, and more importantly, allow it to be extensible to multilingual experiments.

6.1.1 Multilingual SRL Datasets

The ideal scenario for training and evaluating the enhanced multilingual model would be to have data available in multiple languages that use the same definitions for semantic roles. This is not the case in practice, because the bulk of role semantics theory was first developed for English and interest in *transferring* such knowledge to other languages is a recent phenomenon. Therefore, our model must prove to be robust enough to leverage the cross-lingual divergences in existing annotations across languages. Importantly, we do not aim to optimize for a language-specific SRL characterization, but rather use the higher-quality predicate and role definitions available in English to improve performance in other languages.

As discussed in Section 2.1.3, the most widely used datasets for benchmarking SRL monolingual models are the ones defined for the CoNLL-05 (Carreras and Màrquez, 2005), and CoNLL-09 (Hajič et al., 2009) Shared Tasks, of which the latter, despite having aimed for a common SRL strategy for different languages, the subsets for each language diverge from one another in ways that can impact performance, even when using the same model architecture for each language. These divergences include: absence of nominal predicates in non-English languages, different number of sentences available, language-specific definition

Dataset	Language	Train		Test
		# Sents	w/ 1-Pred	w/ 1-Pred
CoNLL-05	EN [Span]	75,187	94,497	5,476
CoNLL-09	EN [Head]	39,279	180,446 (92,908)	10,626
CoNLL-09	DE [Head]	36,020	39,138	2,044
v.d. Plas	FR [Head]	20,012	40,827	2,036
One-to-One	Multilang [Head]	95,311	172,873	-

Table 6.1 Train and Test Data for Monolingual Baseline Models. We show the original number of sentences and the size of the "expanded" data with one copy per predicate. The last row shows the concatenation of the available annotated dependency-based verbal propositions in the three different languages, which we use to train the multilingual model.

of predicate senses, different amount of predicates per sentence and density of annotated roles per sentence.

Figure 6.1 shows the discrepancies in the number of predicate and argument annotations per sentence in the available training data for the four languages used in our experiments. The fact that most sentences in *ES*, *FR* and *DE* have 2 or fewer predicates signifies a considerably smaller number of propositions (training examples) available to the model. Additionally, the overall lower density of arguments per sentence (especially for German) directly impacts the number of labels seen by the model, which receives considerably more exposure to labeled arguments in English sentences compared to the other languages.

For our non-English **monolingual baselines**, we train and test our enhanced Enc-Dec model on the German (DE) dependency-based CoNLL-09 dataset, and the French (FR) PropBank labeled data from van der Plas et al. (2011). We test the model on English as well, using the CoNLL-09 dependency-based English (EN) monolingual data for comparison with German and French; and the English span-based CoNLL-05 data for comparison with the monolingual English results presented in Chapter 5. We describe and compare the specific training data used for each monolingual baseline in Table 6.1. Note that, for direct comparison with the English SOTA we train the English monolingual model with verbal and nominal predicates, and later for the multilingual experiments we only consider verbal predicates.

For the one-to-one **multilingual** experimental setting, we use the same data as in the monolingual baselines. Despite the divergences between the data sets, we expect the model to benefit from exposure to a larger amount of labeled sentences (direct training example augmentations), as well as a higher exposure to annotated arguments per sentence (for the lower-resource languages). We believe these two factors can boost performance compared to the monolingual baselines. Thus, the single multilingual model is trained with the concatenation of the monolingual training datasets of the four language pairs: $(\mathrm{EN}, \mathrm{EN\text{-}SRL})$[1], $(\mathrm{DE}, \mathrm{DE\text{-}SRL})$ and $(\mathrm{FR}, \mathrm{FR\text{-}SRL})$ that is listed in Table 6.1.

6.1.2 Multilingual Model

We propose some modifications to the Enc-Dec model described in Chapter 5. These modifications aim to make a more data-robust model capable of generalizing and making better use of multilingual data. In this setting we are not performing any language translation, but rather utilize all of the training data available from different languages to obtain a stronger training signal, with the aim of benefiting the languages with less labeled data available. This is *one-to-one* multilingual SRL: if the model receives an input sentence in German, it will produce the same German target sentence with labels (similar to the monolingual setting); likewise, if it gets a sentence in French, it will produce a labeled French sentence, and so on.

Our approach of using a single *universal* Enc-Dec for multilingual data is based on Johnson et al. (2017). In their case, because they have access to millions of parallel sentences for MT, they do not apply any architecture modification[2]. In our case, given the fact that our training data is some orders of magnitude smaller than theirs, we need to apply architecture modifications that guide the model for effective multilingual learning, as well as SRL labeling enhancement. We make the following modifications to our vanilla Enc-Dec:

Language Indicator Embeddings. We want the model to profit from the (partial) intersection of role labels used across languages, yet at the same time there are subtle differences in role labeling and how roles are linguistically marked in the different languages[3]. Hence, we define *N* different special tokens, namely the language indicators (e.g., <*EN*>, <*FR*>, <*DE*>), which represent each language with a randomly initialized vector that we fine-tune during training. The model can use at each time-step these language vectors to leverage language-specific properties when generating SRL annotations. Also, by using these embeddings in the decoder, we can help it to stay consistent regarding the language it generates.

Encoder. In our English-only experiments from the last Chapter, we used a 2-layer Bi-LSTM as an encoder. We now adopt the 8-layer Deep Bi-LSTM Encoder with highway connections from He et al. (2017) which has been shown to work well for SRL models. Again, following He et al. (2017), we define the encoder input vector x_i as the concatenation

[1] Since only English has nominal predicate annotations, for the multilingual model we omit them and only keep the verbal predicate argument-structures (which comprise half of the annotated predicates, namely 92,908 propositions).

[2] They only use a *translation token*, which is directly added to each training example. We will follow a similar approach in our cross-lingual setting.

[3] e.g. the role A2 (Beneficiary) can be PP in *EN* and *FR*, but dative NP in *DE* (DativeNP).

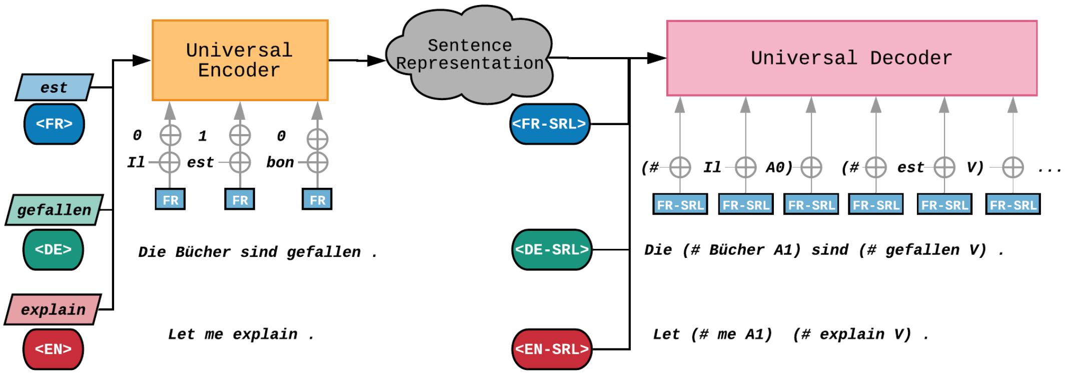


Fig. 6.2 We modify the Enc-Dec architecture from Chapter 5 in order to deal with multilingual and cross-lingual SRL datasets.

of a word embedding w_i and a binary predicate-feature embedding p_i indicating at each time-step whether the current word is a predicate or not. The encoder still outputs a series of hidden states $h_1, ..., h_{T_x}$ representing each token.

Thus, in all multilingual settings, at each time step t we feed the Encoder with a concatenation of the previous encoder state h_{t-1}, the word embedding w_t of the current token, the embedded predicate indicator p_t and the language indicator embedding l_t. The Encoder state update is defined as:

$$h_t = LSTM([h_{t-1}; w_t; p_t; l_t]) \tag{6.1}$$

Decoder. Likewise, on the Decoder side we concatenate the representations for both word tokens and label tokens with the language indicator vector to produce tokens in a specific language. For SRL-labeled output sentences the indicator token for the language embedding is *<DE-SRL>*, *<FR-SRL>*, ... depending on the target language. Formally, at each time step the decoder updates its state by taking into account the previous decoder state s_{t-1}, the previous generated token y_{t-1}, the language indicator embedding l_{t-1} and the attention context vector c_t:

$$s_t = LSTM([s_{t-1}; y_{t-1}; l_{t-1}; c_t]) \tag{6.2}$$

During training we use teacher forcing, feeding the gold target token instead of the previously generated token. We use a common vocabulary for the three languages and keep all tokens that occur more than 5 times in the combined dataset. We train the model with batches containing instances randomly chosen from the individual languages (this means that each batch will contain examples from different language pairs). The training objective

Parameter	Size
Encoder Embedding Size [GloVe, BERT, ELMo]	[300, 768, 1024]
Vocabulary Size	Freq > 5
Encoder Layers	8
Decoder Layers	1
Encoder Hidden	300
Decoder Hidden	500
Attention	1024
Language Embedding Size	200
Predicate Indicator	100
Epochs	30
Early Stopping Patience	5 epochs
Optimizer	Adam
Learning Rate	0.0001
Batch Size [GloVe, Other]	[32, 12]
Gradient Clipping	5
Dropout	0.1

Table 6.2 Hyperparameter configuration used for all the settings tested here: monolingual, multilingual and cross-lingual.

is the same as its vanilla counterpart, that is, minimizing the negative log likelihood of the generated target token by the decoder at each time step, as described in Equation 5.6.

To summarize, our vanilla Enc-Dec model tested in Chapter 5 here is enhanced with a deeper encoder (eight layers instead of two), predicate indicator embeddings for the Encoder and language-specific embeddings for both Encoder and Decoder. Note that all this additions increase the number of parameters of the model; we can afford to do this because by using all training data from the different languages at the same time, we have more data points to tune the network parameters and expect a better result for each language.

6.2 Multilingual Experiments and Results

6.2.1 Hyper-parameters

We evaluate the proposed model using monolingual datasets (SRL labeling) and a combination of datasets in several languages (multilingual SRL labeling). To narrow down the effects of data in the different settings, we use the same hyper-parameter configurations for both experiments. All versions were trained up to 30 epochs using Adam optimizer with a learning rate of 1e-4. We use early stopping (with patience of 5 epochs) based on the BLEU score

Type	Model	Word	CoNLL-05		CoNLL-09	
		Repres.	WSJ	OOD	WSJ	OOD
Span SRL	He 2017	GloVe	84.6	73.6	-	-
	He, 2018	ELMo	83.9	73.7	-	-
	Tan, 2018	GloVe	84.8	74.1	-	-
	Strubell 18 [LISA]	GloVe	84.6	74.5	-	-
	Strubell 18 [LISA*]	ELMo	86.5	78	-	-
	Ouchi 2018	ELMo	**88.5**	79.6	-	-
	Ours [Vanilla]	GloVe	79.2	68.4	-	-
Dep SRL	Roth 2016	DPE*	-	-	87.7	76.1
	Marcheggiani 2017	Dyer*	-	-	87.7	77.7
	Cai et al 2018	GloVe	-	-	89.6	79
Dep and Span SRL	FitzGerald 2015	GloVe	80.3	72.2	87.8	75.5
	Li 2019	ELMo	87.7	-	90.4	-
	Ours [Mono]	GloVe	80.4	70.5	85.5	75.7
	Ours [Mono]	ELMo	88.3	**80.9**	**90.8**	**84.1**

Table 6.3 CoNLL-09 and CoNLL-05 Test Sets for English. Our model with ELMo shows SOTA performance on both types of SRL. LISA* only reports ELMo with predicted predicates; DPE*: dependency path embeddings; Dyer*: Dyer et al. 2015.

of the development set. We represent both words and labels as tokens using N-dimensional vectors. We use pre-trained word embeddings for the 3 languages, which we fine-tune during training[4]. The dimension of embeddings change depending on the pre-trained representations used: 300 for GloVe, 768 for BERT and 1024 for ELMo. All models use the same Encoder (8-layer interlaced BiLSTM – 4 forward and 4 reversed layers) of 300-dimensional hidden size and a single-layer decoder with hidden size of 500. Our predicate feature embedding and language embedding are 100 and 200 dimensions respectively. Due to memory constraints, the batch size is smaller (12 instead of 32) for the models that use contextual representations. The details are fully described in Table 6.2.

6.2.2 Monolingual Baselines

For the monolingual experiment, we benchmark the proposed model against a wide variety of English *EN* models (both span- and dependency-based) that perform the role classification task with gold predicates, as well as against the vanilla version of the proposed system (cf. Chapter 5). The results of this comparison are given in Table 6.3. We find that with the modifications described in this chapter, our proposed seq2seq SRL model improves the existent SOTA for English, obtaining better F1 score for CoNLL-05 out-of-domain test data

[4] Except for the BERT embeddings, which showed better performance when left frozen.

Model	EN-Test	DE-Test	FR-Test
SOTA models*	90.4	80.1	73
Ours-EN [Mono + GloVe]	85.5	-	-
Ours-DE [Mono + GloVe]	-	61.9	-
Ours-FR [Mono + GloVe]	-	-	70.3
Ours-ES [Mono + GloVe]	-	-	-
Mulcaire 2018 [Multi + GloVe]	86.5	69.9	-
Ours [Multi + GloVe]	87	68.2	70.5
Ours [Multi + ELMo]	91.1	75.7	70.7
Ours [Multi + BERT]	89.7	77.2	72.4

Table 6.4 F1 scores for role labeling on dependency-based SRL data. EN and DE Tests: CoNLL-09; FR-Test: van der Plas et al. (2011). State of the art (SOTA) models* are: Cai et al. (2018) [GloVe] for EN, Roth and Lapata (2016) [Dependency-path Embeddings] for DE, van der Plas et al. (2014) [Non-neural] for FR.

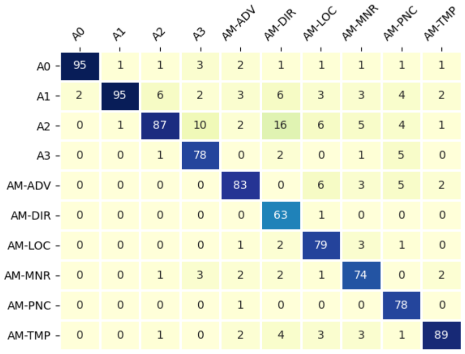
and for both English CoNLL-09 test sets. Importantly, when using GloVe word embeddings, the model is competitive, but still falls several F1 points below SOTA in most settings; on the other hand, when using ELMo embeddings, we achieve SOTA results for both span-based and dependency-based SRL with a single Enc-Dec architecture.

We then compare our system performance across languages for the other two lower-resource languages: *DE*, and *FR*, which only use the respective language dataset as training. The performance for *EN* is compared to performance for *DE* and *FR* settings in the top half of Table 6.4 as well as the respective SOTA results for each language. Results show that, as expected, *EN* setting performance is much higher than the other languages, given its higher density of annotations. In our case, the *EN* model performs at least 10 F1 points higher than the other monolingual models.

6.2.3 Error Analysis

We directly compare our enhanced seq2seq model with the one described in Chapter 5. We analyze again the English CoNLL-05 development and test datasets, as in Section 5.4.3, and perform a similar error analysis. We evaluate argument spans, argument labels on the whole dataset and related to their distance to the main predicate on the development set and overall performance bucketed by sequence length on the test set. We did it this way for fully comparison with both (He et al., 2017) and (Daza and Frank, 2018) reported experiments.

Argument Spans. We first measure the ability of the model to generate the labeled tokens that wrap the appropriate argument spans in the sentence (regardless of the label).



	A0	A1	A2	A3	AM-ADV	AM-DIR	AM-LOC	AM-MNR	AM-PNC	AM-TMP
A0	95	1	1	3	2	1	1	1	1	1
A1	2	95	6	2	3	6	3	3	4	2
A2	0	1	87	10	2	16	6	5	4	1
A3	0	0	1	78	0	2	0	1	5	0
AM-ADV	0	0	0	0	83	0	6	3	5	2
AM-DIR	0	0	0	0	0	63	1	0	0	0
AM-LOC	0	0	0	0	1	2	79	3	1	0
AM-MNR	0	0	1	3	2	2	1	74	0	2
AM-PNC	0	0	0	0	1	0	0	0	78	0
AM-TMP	0	0	1	0	2	4	3	3	1	89

Fig. 6.3 Confusion matrix showing percentage of predicted labels (rows) compared to the gold labels (columns) on the CoNLL-05 development set.

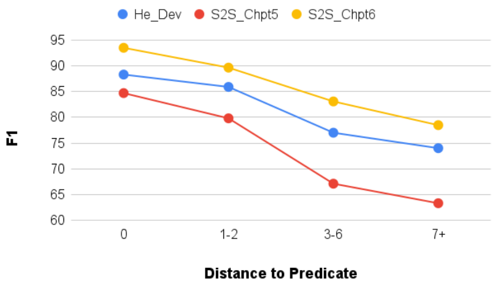


Fig. 6.4 Performance of the model, based on the distance (no. of tokens that separate the argument and the predicate).

This is analogous to measuring argument span identification in classic SRL systems. We found that 84% of the generated the spans match completely with the gold ones, compared to the 77.5% reported earlier (Chapter 5) and only 0.9% do not overlap at all, which is an improvement from the 1.2% reported in the previous model (as measured in the development set).

Argument Labels. Next, we take the subset of correctly predicted spans and analyze how many of their argument labels are correct. Figure 6.3 shows this in a confusion matrix comparable to Figure 5.3. In the current matrix, we also visualize the core labels and the most common modifier argument labels of the dataset. We see small but steady improvements for most labels when compared to the previous confusion matrix. For example, the model presented here correctly predicts A0 and A1 95% of the time (the same score compared with the model from Chapter 5); however, for other case like A2 we see 2% improvement, for AM-DIR we see a +23% chance, for AM-PNC we see 9% improvement, etc. (as measured in the development set).

Distance to predicate. In Figure 5.6 we showed the model performance for arguments according to their distance to the predicate. Here in Figure 6.4 we show the same graph including the SOTA model back then (He_Dev in blue), the model from Chapter 5 (S2S_-Chpt5 in red) and the current model (S2S_Chpt5 in yellow). We still see a similar trend (in the development set) of performance drop across the three models; however, the deeper new model is consistently better for all distances to the predicate. This confirms the argument for robustness given by our deeper encoder and the predicate embeddings which the previous model lacked.

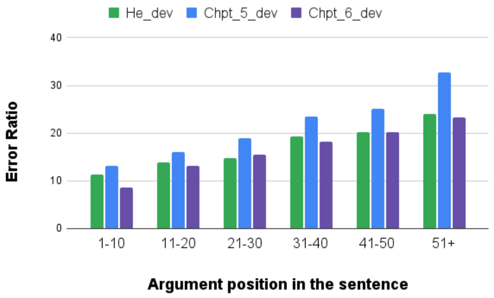


Fig. 6.5 Error ratio of arguments in different regions of the sequences (CoNLL05 development set.)

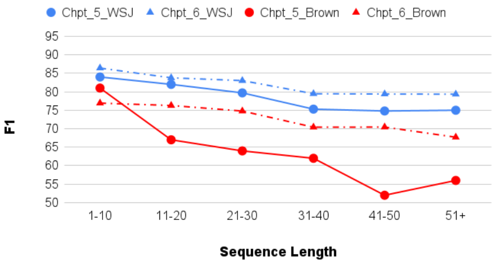


Fig. 6.6 Performance of the model (test set) based on the number of tokens that the sequence has.

Distance from sentence beginning. We also show the error rate of arguments present in different regions of the sentence in the development set (Firgure 6.5). We include the values showed in Figure 5.4 (the SOTA back then is He_dev in green and the previous model is Chpt_5_dev in blue) and add the datapoints from the current model, in purple. Again, the error rate is consistently lower in the enhanced seq2seq model, and this time even better than (He et al., 2017). Notably, the first seq2seq model did not outperformed (He et al., 2017) at any stage, however the enhancements done to the present model boosted the performance above such model.

Sequence Length. Finally, we analyze on the test set (both in-domain and out-of-domain) the robustness of the models with different lengths of sequences. Once more, we see that the deeper encoder and predicate embeddings of the current model provide more robustness on the source sentence representation and notably improves performance in the arguments that are on the last part of the sequences. This is show in Figure 6.6 we show the performance of both models: the lean seq2seq (with triangle points) vs the deeper model from this chapter (with circle points). For sequences longer than 30 tokens the previous model significantly dropped its performance (especially on the out-of-domain data), whereas the new version plateaus in performance even with the longer sequences.

6.2.4 Multilingual Experiment and Results

For the multilingual experiment, we train a single multilingual model on a concatenation of the training data for the three languages *EN*, *DE*, and *FR*; this architecture is identical to the one we used on the monolingual experiments. We use a common vocabulary for the three languages and keep all tokens that occur more than 5 times in the combined data set. We train the model with batches containing instances randomly chosen from the individual languages. This means that each batch might contain examples from different language pairs; our hypothesis is that this should leverage the training signal to optimize the sequence generation for all the languages at the same time. Note that the multilingual data has the CoNLL-09 style labeling, thus is not comparable with the span-based experiments we showed earlier. Nevertheless we still use the same robust architecture, and can measure how much does the combination of multilingual training data affect the results by comparing the monolingual baselines with their multilingual counterparts. Table 6.4 compares the one-to-one multilingual systems to each other in the bottom half of the table, as well as the polyglot SRL system of Mulcaire et al. (2018), which also leverages data from multiple languages during training; besides, improvements can be directly seen compared to the monolingual systems in the upper half of the same table.

Multilingual training yields improvement on each of the four languages studied here, when compared to the monolingual baselines. The largest benefit occurs for German, where we observe more than 6 points (F1) of improvement. In comparison to a concurrent work to ours, the polyglot system of Mulcaire et al. (2018), we obtain better results for English using GloVe and for both English and German when using ELMo embeddings. Finally, we observe that adding contextual representations to our model results in additional improvements across the board. Unfortunately, the important gains shown for lower-resource languages is not enough to improve the current SOTA. However, it is worth noticing that the German SOTA is a hybrid model that uses neural dependency path embeddings as features for a classic SRL system (Toutanova et al., 2008); whereas the French SOTA is based on corpus-specific maximum likelihood estimates (van der Plas et al., 2014). In contrast, our model is a straight-forward architecture that provides a flexible solution that is able to combine data for different languages as well as pre-trained word embeddings, which has the potential for better generalization.

6.3 Cross-lingual Semantic Role Labeling

After validating the robustness of our architecture when handling different languages simultaneously, we can now perform cross-lingual SRL. We consider that the positive results on monolingual and multilingual settings were good steps towards confirming that the Enc-Dec is suitable for the SRL task and more importantly, that it is possible to handle data from multiple languages at the same time without loosing labeling effectiveness in the individual language test sets. After this has been established, we can focus on the cross-lingual evaluation which by nature is more difficult. This happens because the cross-lingual training of an Enc-Dec involves several points of uncertainty, especially for evaluation purposes:

Cross-lingual Setting	# Sentences	w/ 1-Pred
EN - DE-SRL (Akbik, 2015)	41,993	63,397
EN - FR-SRL (Akbik, 2015)	20,012	40,827
EN - FR (UN)	100,000	-
EN - DE (Europarl)	100,000	-

Table 6.5 Data used for Cross-lingual Models: From the SRL parallel data available we take 90% for training and use the rest as a *Dev* set for our experiments. We add the non-labeled data (from UN and Europarl) during training to enforce translation knowledge.

Our cross-lingual labeling experiment is fundamentally different from the monolingual and multilingual experiments because here we are not only labeling sentences; instead, once the model is trained with a cross-lingual signal (e.g. with *EN* sources paired to their corresponding labeled German *DE-SRL* translated sentences). This is possible because of the generative properties of the decoder, which opens the possibility to generate new labeled sentences. This, however, also means that there is no control over the resulting target sentences (a similar case occurs when training MT models, where the translation system aims to *approximate* a translation, but there is no single correct translation). Thus, in this setting we will evaluate what happens once we have access to a trained cross-lingual model, which we can use to produce entirely novel labeled sentences and use them as means for labeled data augmentation for lower-resource languages. In the following sections we explain where we obtain the data to train such a model (Section 6.3.1), and then we show how the system is trained (Section 6.3.2).

6.3.1 Cross-lingual SRL Datasets

While the datasets used for the multilingual experiment in Section 6.1.1 are available in different languages, they are not parallel datasets, and therefore not suitable for our cross-lingual experiments. For these experiments, we therefore use the dependency-based labeled SRL corpus that was used as training data to generate the Universal Proposition Banks Akbik et al. (2015)[5]. This data was requested to the authors, who created it via annotation projection and active learning methods on parallel corpora from *EN* to *DE*, *FR* respectively. Importantly, these sentences are already pre-filtered to ensure that the predicate sense of the source predicate is preserved in the target sentence[6].

[5] We do not use the Universal Proposition Banks because they are not parallel

[6] This is the main reason for choosing an available dataset as opposed to artificially creating parallel data. It is important to have filtered parallel data, since we seek to avoid translation shifts in our training data, and finding this correspondence between source and target sentences is not trivial

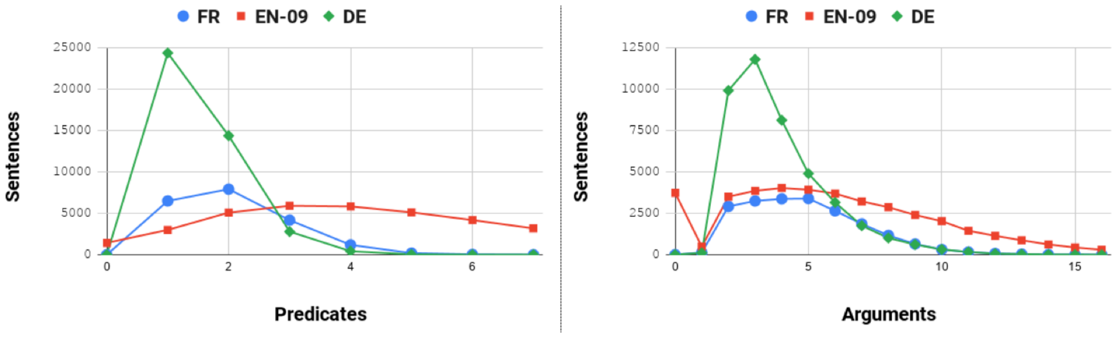


Fig. 6.7 Distribution of predicates (left-hand side) and arguments (right-hand side) in the available cross-lingual datasets. We include the distribution of the English CoNLL-09 data as an *ideal* distribution, and observe that the available data has considerably less propositions per sentence and fewer argument labels available.

Also, since the role labels were projected from automatically PropBank-parsed English sentences, the same label set is used across all languages, including predicate senses. This training data is a subset of sentences taken from underlying existing Machine Translation (MT) parallel corpora: Europarl (Koehn et al., 2003) for *EN-DE* (about 63K labeled parallel sentences), and UN (Ziemski et al., 2016) for *EN-FR* (about 40K labeled parallel sentences). Since we only had access to the labeled sentences (target-side), we constructed our parallel training pairs *EN* to *FR-SRL* and *EN* to *DE-SRL* by finding the original source English counterparts in the full set of parallel sentences. Once we found the source sentences that correspond to the labeled targets, we need to know what predicates are present in the source sentence (to populate the predicate-indicator embeddings that the Encoder uses).

We used Flair (Akbik et al., 2018) to obtain predicted PropBank frames on the English source sentences and then found the alignment to the (already) labeled predicate on the target side by looking for the matching predicate sense; if no equal sense was found in target, the proposition was dropped. Because the sense was predicted for the source, and it is also labeled in the target, we treat this as a valid cross-lingual predicate-argument structure. By following this method, we ended with 63,397 labeled training examples for (*EN-DE-SRL*) and 40,827 for (*EN-FR-SRL*). We show in Figure 6.7 the density of our training corpus, and compare it with the English CoNLL-09 corpus, which is our guide for a *successful* training data set for neural network models. We observe that the amount of labels is also quite low compared to English, however, by combining both datasets in a single model we expect to still obtain good quality target labeled data.

Further, besides the parallel SRL-labeled data, we choose a subset of 100K parallel (non-labeled) sentences for each language pair from the mentioned MT datasets (Europarl and UN corpora) and use them as training examples to improve the translation quality of

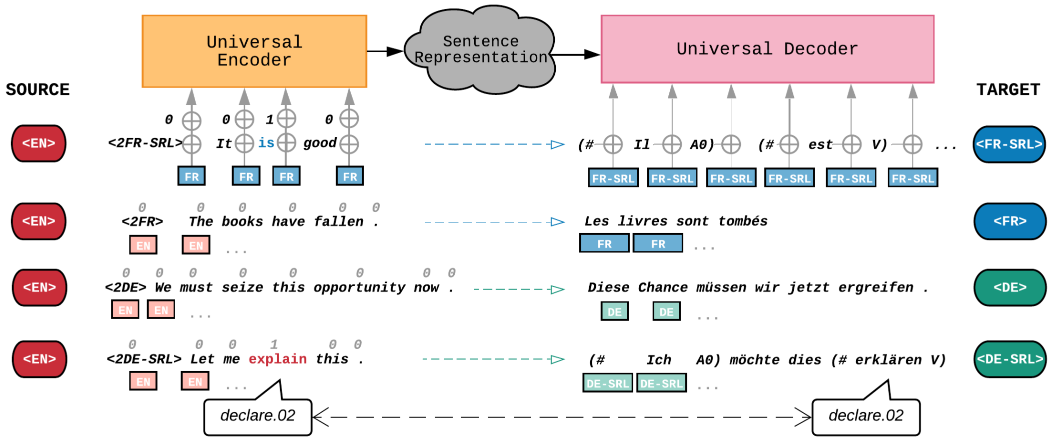

Universal Encoder
Sentence Representation
Universal Decoder
SOURCE
TARGET
<EN>
<2FR-SRL> It is good
0 0 1 0
FR FR FR FR
(# Il A0) (# est V) ...
FR-SRL FR-SRL FR-SRL FR-SRL FR-SRL FR-SRL
<FR-SRL>
<2FR> The books have fallen .
Les livres sont tombés ...
<FR>
<2DE> We must seize this opportunity now .
Diese Chance müssen wir jetzt ergreifen .
<DE>
<2DE-SRL> Let me explain this .
(# Ich A0) möchte dies (# erklären V)
<DE-SRL>
EN EN ...
DE DE ...
DE-SRL DE-SRL ...
declare.02
declare.02

Fig. 6.8 For cross-lingual SRL, we utilize the same multilingual architecture as in Section 6.1 (Figure 6.2), but this time we train it using cross-lingual datasets.

the model. We use 90% for training and the remaining 10% as the development set. The available cross-lingual data is described in Table 6.5.

6.3.2 Training Cross-lingual SRL

When performing cross-lingual SRL, the proposed model needs to accomplish two tasks: in addition to generating appropriate SRL labels, it needs to properly translate sentences from the source into the target language. We train a single SRL model on a concatenation of the parallel datasets described in Section 6.3.1 which we synthesize in Table 6.5. Note that the model is trained on four different language source-target pairs (each pair is a row in the table). The inclusion of MT data is intended to reinforce the translation knowledge of the model and improve the source to target alignment (via the attention mechanism), so that it can generate fluent properly labeled target sentences. Because the model treats this additional data as a separate language-pair, this does not directly impact the density of labels on the target side for the labeled outputs. Following the same strategy as for multilingual training, we feed the model with alternating batches of randomly chosen instances from the various language pairs (each batch contain multiple language pairs). Note that in this training schema, the amount of MT data that we can add is restricted by the amount of labeled multilingual data, since we do not want the non-labeled language pairs to dominate too much and therefore affect the labeling capabilities of the model.

Finally, we also deviate from the multilingual architecture by adding a **translation token** to each training example. Following Johnson et al. (2017), this involves prefixing the source sequence with a special token that indicates the expected language of the target sequence.

For example, if the source is in *EN* and the target is a German sentence with SRL labels (*DE-SRL*), the source sentence will be preceded by the special token <*2DE-SRL*>.

6.4 Cross-Lingual Experiment and Results

6.4.1 Evaluation Strategy

In this section, we evaluate the cross-lingual setting of our Enc-Dec model, which was trained with the same hyper-parameters given in Section 6.2.1 and described in Table 6.2 in order to preserve consistency through all experiments. We trained two versions of the model, where the only difference is the pre-trained embedding layer: one uses GloVe embeddings (Pennington et al., 2014) and the latter uses mBERT (Devlin et al., 2019).

Due to the generative capacity of the cross-lingual model, it can be used for augmentation of labeled data for lower-resource languages. This poses an evaluation challenge: the decoder may generate output in the target language, which are unseen, but whose predicate-argument labels are valid and useful. Because the copying mechanism is not in place here to enforce exact reproduction of the source words, as was the case in the monolingual and one-to-one multilingual settings, there is the possibility that the generated sentences do not match the gold reference. Since there is no gold standard to straight-forwardly evaluate the model outputs with, we need an alternative method to judge the validity of the predicate-argument structures of these unseen items.

This setting, where the output is intended to approximate target references as closely as possible but will not necessarily be identical, resembles the setting in classical MT, language generation, as well as label projection. Inspired by approaches used in this field, we develop an evaluation strategy that assesses the correctness of SRL labels as well as the quality of the translations and the usability of the generated data as training examples. We assess the performance of our system in three evaluation settings: i) an automatic **intrinsic evaluation** using BLEU score (Papineni et al., 2002) as a proxy for translation and expected argument label quality, ii) an **extrinsic evaluation** using labeled sentences generated by our system to augment the training set for a resource-poor language, iii) a small-scale **human evaluation** where we evaluate the automatically assigned SRL labels against 226 sentences that were manually judged and labeled by human annotators to give an estimation of the quality of the generated data.

	German (*DE*)			**French (*FR*)**		
Model [Filter]	**Full Seq**	**Word**	**Label**	**Full Seq**	**Word**	**Label**
XL-GloVe [All]	18.86	17.17	25.52	28.99	17.36	32.76
XL-BERT [All]	27.22	27.36	29.59	33.59	22.48	37.17
XL-GloVe [$\geq$ 10]	30.58	36.71	51.68	38.99	43.79	61.73
XL-BERT [$\geq$ 10]	36.95	41.36	55.73	42.66	46.52	65.32

Table 6.6 Cross-lingual (XL) system results using BLEU score on individual languages from the *Dev* set. We compute BLEU on labeled sequences (Full Seq), and separately for words and only labels. We also show scores when applying a filter on Full Seq of BLEU ≥ 10.

6.4.2 Intrinsic Evaluation

BLEU Scores. In the absence of a gold standard against which to test our output, we make use of BLEU scores (Papineni et al., 2002), a common metric in MT research (Sutskever et al., 2014; Bahdanau et al., 2015; Firat et al., 2016a; Johnson et al., 2017) to automatically assess the degree of closeness between outputs and reference instances. BLEU scores give only a rough estimate of the quality of translations, which is why we do it as a first step to measure the cross-lingual task. We perform this evaluation on the development set, since for our inference step we do not know the expected target labels. We split the BLEU measurement in three different sub-cases: i) translation quality and labeling quality as a whole, computing BLEU for the full system output sequences against the target reference sequences, we call this *full labeled sequences* (both word and label outputs), ii) we strip the labels from both system output and target reference and compute its BLEU score (*words only*), and finally iii) We keep only the generated labels and the reference labels and compute BLEU on them (*labels only*).

For the GloVe version, we can see that the quality of predictions of words is similar for both the German and French instances. Notably, both systems have great difficulty with German than with French. We also observe that adding multilingual BERT is very helpful for obtaining even more fluent and correct labeled outputs (according to BLEU) resulting in ca. +9 points in German and +5 in French on the full sequences. This is very important given that we have a small training set compared to classic NMT scenarios.

Output Filtering. The average BLEU score is not very informative about the individual sentences that contain high-quality labeled sentences, which are the ones that ultimately can be used as training data. Therefore, we also make use of the development set to search for a BLEU threshold for which we manage to keep the best quality sentences while also keeping enough quantity of examples, we call this the *sentence quality threshold*.

We will use this threshold for filtering out the excessively noisy output instances while retaining a large enough number of candidate novel training instances. Because the overarching goal is to improve SRL for lower-resource languages, we assess only the quality of predictions in German and French. We experiment with discarding items with BLEU scores below a certain threshold[7] as a method of improving output quality. A threshold of BLEU ≥ 10 was found to give the best trade-off between an increase in average BLEU score (presumably reflecting higher sentence quality) and decrease in available predictions (above the threshold). The lower part of Table 6.6 shows the scores when restricting the evaluation to sentences with score ≥ 10. By keeping only the filtered subset of sentences we achieve an improvement of approx. 10 BLEU points on average on the full sequences (*Full Seq*), and almost double the score for *labels only*. This holds for both the GloVe and BERT versions and for both languages; however, given that the BERT version of the model gives significantly better outputs, for the next two evaluation setups, we will always use the XL-BERT outputs to perform the closer dissection of the generated data. Moreover, we only keep the filtered outputs with quality threshold ≥ 10. In the next section we explain how this is used for an extrinsic evaluation setup.

Additionally, we also measure the improvement as measured by BERTScore (Zhang et al., 2020): The unfiltered DE dataset obtains an average BERTScore of 67.63, whereas the filtered version (BLEU higher than 10) goes up to 72.99 points; on the FR dataset, we see an unfiltered score of 73.89 and the filtered version goes up to 76.22. We cannot use these scores as a hard filter (like with BLEU), since the scores tend to be saturated on the upper numbers (i.e. close to 1)[8] and while it is useful to rank the similarity of sentences, establishing a partition based on it would be too risky.

6.4.3 Extrinsic Evaluation

Threshold Application. We use our cross-lingual model with pre-trained mBERT embeddings as a label data generator by applying it on 100K *EN* sentences from Europarl and 100K UN corpora not seen during training[9] and let the model predict *DE-SRL* and *FR-SRL* as target languages respectively. This results in previously *unseen* German and French labeled sentences. Since there is no guarantee that the generated sentences should preserve the source predicate meaning, we first filter all outputs by keeping only those that come close to the original sentence meaning. We approximate this by back-translating the generated outputs and applying the quality filter (BLEU $\geq$ 10) on them. We perform the back-translation by

[7] Concretely we experimented with thresholds of 5, 10, 20 and 30

[8] We give more details on the behavior of BERTScore on the next chapter where we use it more actively

[9] Note that these are taken from a different subset than the parallel sentences used during training.

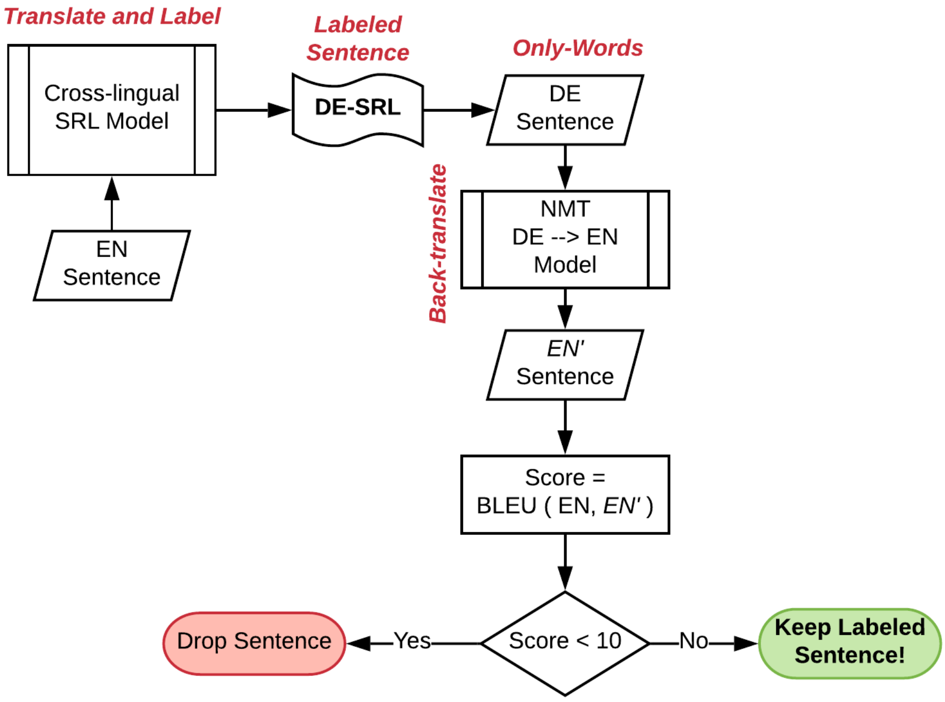


Fig. 6.9 We apply a back-translation filter to the model outputs to exclude the translations whose meaning is not as close to the source, increasing the probability of preserving the source meaning on the target side.

Model	Added Type	Added Size	Total Size	F1 Test
DE [Mono]	-	-	39K	61.90
DE [Mono]	LabelProj	44K	83K	62.37
DE [Mono]	OurGen	10K	49K	62.40
DE [Mono]	OurGen	20K	59K	62.46
DE [Mono]	OurGen	30K	69K	62.81
DE [Mono]	**OurGen**	**44K**	**83K**	**63.57**
FR [Mono]	-	-	73K	70.30
FR [Mono]	LabelProj	32K	105K	70.45
FR [Mono]	OurGen	10K	83K	70.33
FR [Mono]	**OurGen**	**32K**	**93K**	**70.52**
FR [Mono]	OurGen	All	105K	70.39

Table 6.7 We retrain the monolingual systems *DE, FR* using the original training sets (no added data) shown in Table 6.1 and compare it to performance of models trained on increasing amounts of generated data added to the original data. We also compare to the stronger baseline *LabelProj* where we add data created by label projection (Akbik et al., 2015)

(stripping the labels and keeping only the words) using a pre-trained *DE-EN* and *FR-EN* model from OpenNMT Klein et al. (2017). See Figure 6.9 for a graphic explanation of this process.

The logic behind this is that if the back-translation is close enough to the source, the generated target sentence preserves a fair amount of the original sentence meaning[10]. By following this strategy, after applying the quality filter, we end up with a parallel dataset of 44K generated sentences for *(EN, DE-SRL)* and 32K for (*EN*, *FR-SRL*).

Data Augmentation. We use the filtered generated data to augment the original training sets of our two resource-poor languages, namely *DE* and *FR* (we augment the CoNLL-09 train set for German and the training set of van der Plas et al. (2011) for French). We train our monolingual Enc-Dec model with the augmented data in steps of 10K, until we have added the complete generated data set and measure the increase in F1 score when training different models with incrementally augmented data. Additionally, we show a comparison of the improvement achieved when adding the same amount of sentences produced by ZAP[11], a SOTA label projection framework (Akbik and Vollgraf, 2018), to have a better measurement of the gains obtained by our method.

[10] BLEU score is used as a naive approach to avoid excessively noisy data but we could also develop, for example, a semantic similarity metric to also keep sentences that are close enough to the original predicate sense meaning.

[11] https://github.com/zalandoresearch/zap

As it can be observed in Table 6.7, adding our German data shows improvement in F1 score in the German dataset, despite the fact that the CoNLL-09 label scheme contains arguments not seen in our training data (namely A5-A9). We still observe improvement because the frequency of the major roles (*A0* and *A1*) is considerably higher than that of the unseen minority arguments. In the case of French, the improvement is not as significant, but the effect of adding projected data follows a similar trend. More importantly, our training data results in better F1 on the test set when compared to the SOTA label projection software.

6.4.4 Human Evaluation

To provide an in-depth quality assessment of the generated sentences, we also conduct human evaluation. We create a small-scale gold standard consisting of 226 sentences that is given to two annotators. The exact guidelines that the annotators followed can be found in Appendix A. To select a representative sample from our newly generated labeled sentences,[12] we analyze the distribution of labels in the data and apply stratified sampling to cover as many predicates as possible and as many role label variants as possible. We judge these sentences on the quality of the generated language and annotate them with PropBank roles. Because manual annotation is costly and German proved to be more challenging according to the BLEU-score based comparison (Table 6.6) we conduct this manual evaluation only on the German data.

Translation Quality. We ask two native speaker annotators to score each output sentence (they see only the words, not the labels) on a scale of 1-5 for *Quality* (where 1: 'is completely ungrammatical'; 5: 'is perfectly grammatical') and for *Naturalness* (where 1: 'The sentence is not what a native speaker would write'; 5: 'The sentence could have been written by a native speaker'). We obtain a high average score of 4.4 for *Quality* and 4.2 for *Naturalness*.

SRL Performance. To avoid the need for trained PropBank annotators, we use an annotation method based on the question-based role annotation method of He et al. (2015), Annotation with this technique entails using question and answer pairs in order to label the predicate-argument structure of verbs. The process consists of several sub-tasks: i) to generate questions targeting a specific verb in a sentence and to mark as answers a subset of words from the same sentence, ii) to choose the head word of each selected subset and iii) to assign a PropBank label to this head according to a table that correlates WH-phrases with the most likely label, as depicted in Figure 6.10. The table that the annotators used as a

[12] i.e., the generated sentences for which we measured a BLEU score $\geq$ 10 against the source using back-translation.

SOURCE (EN): Well , that word means more power for the European Union .

WORDS-OUT (DE): Nun bedeutet das Wort mehr Macht für die Europäische Union .

Question	Answer	Head	Label
Was bedeutet mehr Macht für die Europäische Union ?	das Wort	Wort	A0 ✓
Was bedeutet das Wort für die Europäische Union ?	mehr Macht	Macht	A1 ✓
Für wen bedeutet das Wort mehr Macht ?	für die Europäische Union	für	A2 ✓
Discourse Marker	Nun	Nun	AM-DIS ✓

LABELED-OUT: (# Nun **AM-DIS)** (# bedeutet **V)** das (# Wort **A0)** mehr (# Macht **A1)** (# für **A2)** die Europäische Union .

Fig. 6.10 Indirect QA Annotation Example: The annotators see the German words that the system generated and a specific central predicate. Based on this, they have to create WH-questions whose answers are a subset inside the same sentence. A mapping from the question to the appropriate label is provided later.

guide to map their questions to a corresponding SRL label are given in Table 6.8). With this method we can compare the (gold) labels assigned by the annotator vs. the labeled-output of the system, because both sequences coincide at the word level and thus we can compute F1 labeling score on the subset of human annotated outputs.

We ask two linguistically trained annotators to perform this task independently and compute Krippendorff's Alpha (Krippendorff, 1980) on the role labels, which returns an inter-annotator agreement score of 82.83. We resolve conflicting annotations through discussion among the annotators. The resulting gold standard contains 737 annotated roles. Notably, the most prominent roles (as in the CoNLL-09 datasets) are *A0* and *A1* which are normally related to the agent and the patient in sentences, but the annotated data also includes modifier roles such as temporal, modal, discourse markers, among others[13].

Using our human-annotated sentences, we can determine that the automatic labeling performance of our cross-lingual SRL model (XL-BERT) achieves an F1 score of 73.21 (73.33 precision, 73.1 recall). We also evaluate performance of the label projection system of Akbik and Vollgraf (2018) on this data. We only consider arguments of the predicates that were annotated, and find that ZAP obtains a low F1 score of 56.03 (42.65 precision,

[13] The label distribution is given in the Appendix A

Role	**Question**
A0 [Agent]	Who? What?
A1 [Patient]	What? Who? How much?
A2 [Patient 2]	What? How much? Where?
A3 [Patient 3]	What? Who?
A4 [Patient 4]	-
AM-DIR [Direction]	To where?
AM-LOC [Location]	Where?
AM-MNR [Manner - modify verb]	How?
AM-TMP [Temporal]	When?
AM-EXT [Extent]	How much? How?
AM-PNC [Purpose]	Why?
AM-CAU [Cause]	Why?
AM-ADV [Adverbial - modify entire sentence]	Why?
AM-DIS [Discourse Marker / Vocatives]	#AM-DIS / #VOCATIVE
AM-MOD [Modals]	#AM-MOD
AM-NEG [Negation]	#AM-NEG

Table 6.8 WH-Questions used to elicit manual semantic role annotations.

81.7 recall). Akbik and Vollgraf (2018)'s label projection method shows more unstable results, with a very low precision. Most likely this is due to the fact that it uses a statistically learned predicate dictionary and also due to the word alignment noise; whereas XL-BERT shows much better, and more precise results than this baseline, presumably because it was pre-trained in a bigger amount of data and possess the lexical knowledge to identify more predicates. The results that our model achieves are overall acceptable and stable in terms of labeling quality, suggesting that the joint translation-labeling method was successful.

Chapter 7

X-SRL: A Parallel Cross-lingual Semantic Role Labeling Corpus

7.1 Overview and Motivation for the X-SRL Corpus

In the previous two chapters, we showed that SRL can be formulated as a seq2seq task, for English and for other languages as well. We addressed the data scarcity problem on lower-resource languages by training models that use all the available multilingual labeled data at the same time, obtaining better results when compared to monolingual versions. Moreover, we showed that the main advantage of using an Enc-Dec to perform SRL is the possibility to not only use the model as a labeler but also as a generator for new training data for the lower-resource languages.

To train our models we used an already available annotated corpus with cross-lingual annotations as training data. We noted that there are discrepancies when it comes to available training datasets for different seq2seq settings. Firstly, while there are monolingual datasets available in different languages they were constructed with independent processes and rules of annotation, sometimes even semi-automatic ones. As a result, they have differing densities of annotations per sentence, as well as differing labeling definitions. This fact limited the number of languages we could integrate into our multilingual setting, and may have as well limited the improvements for those languages we managed to integrate. Secondly, the existing parallel corpora that we were able to use for the cross-lingual seq2seq setting was artificially created by means of an independent label projection method, for which no original gold annotations exist. In order to achieve high enough precision in the output, the creators of such data employed strict filtering methods, resulting in low-density annotated sentences in the target languages. Thirdly, there is a discrepancy in domain between the monolingual datasets and parallel datasets, which are sampled from news-wire text and parliamentary speeches respectively, which might be affecting the integration when being combined with one another. In short, there are issues with compatibility and precision in the available multilingual and cross-lingual data that limits multilingual integration and restricted the proposed model's performance.

We hypothesize that, by relying on the latest advances in multilingual contextualized word representations, we can develop a method for creating higher-quality cross-lingual data that helps to improve SRL labeling quality as well as generate better training data in lower-resource languages, without relying on annotation projection statistical models. Even though our Enc-Dec was shown to be robust enough to produce useful annotated data with the available training data, if we develop a method for obtaining more cross-lingual labeled data to train it.

In this chapter, we develop a new method for obtaining more cross-lingual training data in non-English languages, and test whether our seq2seq architecture can exploit better the multilingual shared properties of this improved training data. Now that we have full control over the annotation scheme, we are able to target more languages. For comparison purposes, we develop data for the same lower-resource languages targeted in the previous chapter (German and French), as well as for a third language: Spanish. We did not involve Spanish before because we observed large differences in predicate senses and argument label definitions between the available SRL labeled Spanish data and the other monolingual datasets, which we believed could harm performance.

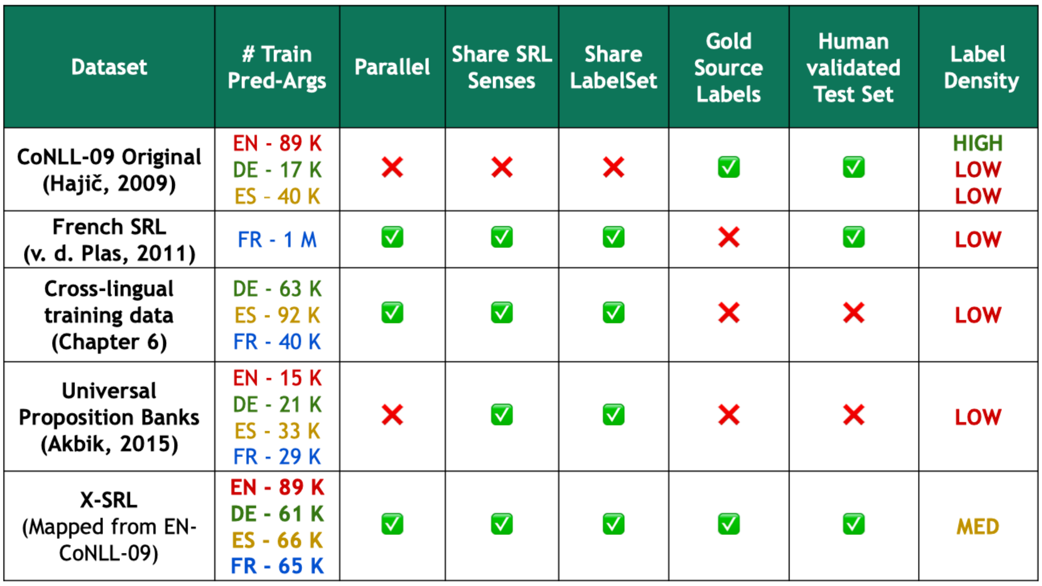
Ideally, we would like to have data with all of the following properties:

- Multi-way parallel sentences across languages: go beyond bi-lingual aligned data and have sentences that are parallel in several languages at the same time, to evaluate the differences in predicate-argument structures across languages.

- Shared domain across languages and the cross-lingual data and the high-quality SRL labels: we would like to have cross-lingual data on the news-wire domain, to directly match the domain of the available monolingual SRL annotations.

- Sentences with a similar density of annotations across languages: to improve multi-way compatibility of sentences, we aim to have annotated data that has a similar amount of labels across-languages.

- A human-validated test set with gold labels in non-English languages: the availability of a human-validated set can ease the evaluation of future cross-lingual methods.



Dataset	# Train Pred-Args	Parallel	Share SRL Senses	Share LabelSet	Gold Source Labels	Human validated Test Set	Label Density
CoNLL-09 Original (Hajič, 2009)	EN - 89 K DE - 17 K ES - 40 K	✗	✗	✗	☑	☑	HIGH LOW LOW
French SRL (v. d. Plas, 2011)	FR - 1 M	☑	☑	☑	✗	☑	LOW
Cross-lingual training data (Chapter 6)	DE - 63 K ES - 92 K FR - 40 K	☑	☑	☑	✗	✗	LOW
Universal Proposition Banks (Akbik, 2015)	EN - 15 K DE - 21 K ES - 33 K FR - 29 K	✗	☑	☑	✗	✗	LOW
X-SRL (Mapped from EN-CoNLL-09)	EN - 89 K DE - 61 K ES - 66 K FR - 65 K	☑	☑	☑	☑	☑	MED

Fig. 7.1 A comparison of available training data for the four different languages studied here. We can see that currently no available dataset fulfills all of the desirable characteristics for optimal multilingual and cross-lingual training, a gap that X-SRL aims to fill-in.

None of the SRL datasets available prior to our work satisfy these four things at the same time (see Figure 7.1). For this reason, we want to create our own parallel SRL labeled corpus. To create such corpus, which we name X-SRL, we propose the following:

1. Translate the official English dependency-based SRL corpus (CoNLL-09 corpus, comprised of around 40,000 sentences) to German, French and Spanish. This can be done using a SOTA MT system, to ensure rapid processing of the data with the best possible translation quality. We hypothesise that a good quality MT system will produce faithful lexical translations that will preserve the majority of the original source predicates on the target side.

2. Once we have the parallel sentences we will propose our own annotation projection technique to transfer the English high-quality labels to the target languages.

3. To validate the quality of our systems, we need a human-labeled test dataset. We also use MT to obtain parallel sentences to the CoNLL-09 English test set (around 2,400 sentences) and hire annotators with knowledge in translation that help us to analyze and in this case manually validate the quality of translations as well as decide which predicates and roles should be transferred to the target language.

By following these steps, we create the first cross-lingual and multi-way parallel dataset for SRL with homogeneous and similar amount of annotations across languages (see Figure

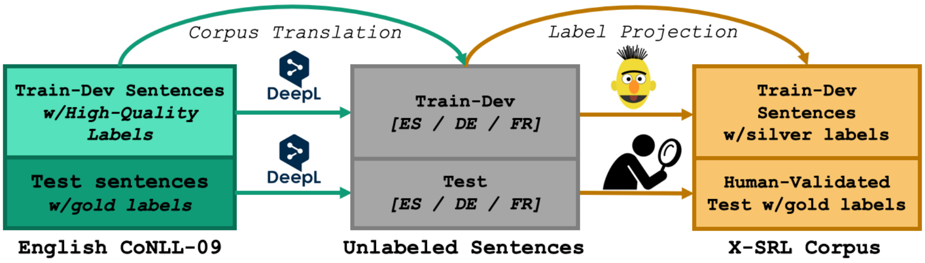

Corpus Translation
Label Projection
Train-Dev Sentences w/High-Quality Labels
Test sentences w/gold labels
English CoNLL-09
DeepL
DeepL
Train-Dev [ES / DE / FR]
Test [ES / DE / FR]
Unlabeled Sentences
Train-Dev Sentences w/silver labels
Human-Validated Test w/gold labels
X-SRL Corpus

Fig. 7.2 Method to create X-SRL. We automatically translate the English CoNLL-09 corpus, use a fast label projection method for *train-dev* and get human annotators to select the appropriate head words on the target sentences to obtain gold annotations for the *test* sets.

7.2). Having such a corpus allows to fully exploit the multilingual properties of SRL and improve performance by exploiting the best of our seq2seq architecture. We can use this knowledge to generate more training data using our cross-lingual system and possibly e.g., make a better SRL labeler for German. Finally, we will have a reliable test set that was validated by humans that will allow us to measure our performance more accurately.

In the following sections we describe how we obtain parallel sentences to the gold-labeled CoNLL-09 English corpus (Section 7.2). Next, we show how the human-validated labels (only for the test sets) were obtained in an efficient way (Section 7.3). We then describe the details of how we perform (automatic) annotation projection enhanced with simple filters for *train/dev* in Section 7.4. With this we achieve new large annotated SRL datasets for German, French and Spanish.

Notably, when building the X-SRL dataset, in line with the current PropBank SRL data available in different languages, we focus on verbal predicates only. We have already mentioned that the English CoNLL-09 data includes both verbal and nominal predicate annotations; by contrast, the remaining languages with PropBank SRL training data (including the CoNLL-09 non-English data) only provide annotations for verbal predicates. While we could attempt projecting the English nominal predicate annotations and create an X-SRL dataset that includes nominal SRL for all target languages – which would mean a big advance over the current situation – admitting nominal and verbal SRL annotations in a multilingual setting would confront us with many translation shifts. We could try to capture these for the manually curated test set, but we would run the risk of generating noisy or scarce target annotations when projecting them for the *train/dev* sections.

The reasons for this are complex: first, by including nominal SRL, we would be confronted with translation shifts in both directions, e.g. Noun-to-Verb or Verb-to-Noun translations. For these, we would have to verify whether they correspond to valid *verbalizations* or *nominalizations* on the target side. This would lead to considerable overhead and, most likely,

	EN			DE			ES			FR		
X-SRL	Sents	Preds	Args	Sents	Preds	Args	Sents	Preds	Args	Sents	Preds	Args
Train	39,279	92,908	238,887	39,279	60,861	134,714	39,279	68,844	154,536	39,279	67,878	154,279
Dev	1,334	3,321	8,407	1,334	2,152	4,584	1,334	2,400	5,281	1,334	2,408	5,388
Test	2,399	5,217	14,156	2,213	4,086	11,050	2,346	4,376	10,529	2,095	3,770	9,854

Table 7.1 Overall statistics for X-SRL.

noise in automatic projection. Also, translation shifts often involve light verb constructions, which require special role annotations. These would be difficult to assign in automatic projection. We thus defer the inclusion of nominal SRL to future work.

In Table 7.1 we provide the final statistics for our X-SRL corpus. Note that the English statistics are exactly the same as the CoNLL-09 English corpus, and the other three languages are the result of our projection and test set validation methods described in the following sections, where we give in detail the method for constructing each of the *train/dev/test* sections to end with the given amount of sentences and annotations for each language.

7.2 Dataset Translation

We aim to produce high-quality labeled corpora while reducing as much as possible the amount of time, cost and human intervention needed to fulfill this task. We use MT to perform dataset translation, obviating the need of human translator services or parallel corpora availability. As previous work Tiedemann and Agic (2016); Tyers et al. (2018) has shown, automatic translations are useful as supervision for syntactic dependency labeling tasks since they are quite close to the source languages; likewise, in Argumentation Mining, Eger et al. (2018) achieve comparable results to using human-translated data. One could argue that by automatically translating the English source, we may run into the problem of *translationese*. Translationese occurs when – in an attempt to reproduce the meaning of a text in a foreign language – the resulting translation is grammatically correct but carries over language-specific constructs from the source language to the target. While it would be interesting to study possible shining-through effects in our automatically translated target texts and any potential impact on SRL performance (e.g. by comparing a natural vs. translated test set), our main concern is to preserve the relevant *predicate-argument structures* in order to give a strong-enough signal to train our SRL systems, and our initial assumption relies on the evidence from the mentioned previous works (confirmed by our results) that obtaining relevant training data is possible with MT generated sentences.

In sum, we take as source the set of sentences in the English CoNLL-09 dataset, which are tokenized and annotated for part-of-speech (POS), syntactic dependencies, predicate senses and semantic roles. We use the DeepL[1] software to obtain translations of each sentence into the three target languages. For all target sentences we use spaCy[2] to tokenize, assign POS tags (language-specific and universal POS) as well as syntactic dependency annotations. This gives us a 4-way parallel corpus with syntactic information on both sides.

7.3 Test Set Annotation

7.3.1 Annotation Setup.

To confirm the quality of the translations delivered by DeepL, we hired 12 annotators with a background in translation studies and experience in $EN \rightarrow T$ translation (we hired 4 annotators for each language pair) to rate and validate the automatic translations of the test set[3] by following a guideline that explains the quality validation and the annotation processes[4]. First, we ask them to rate the translations on a scale from 1-5 (worst to best). On the basis of the obtained ratings, we apply a filter and keep only the sentences with *quality rating 3, 4, or 5*, since lower quality implies that the translations are ungrammatical.

Only on this subset of good-quality sentences we require them to do three more tasks: i) we show them the *labeled verbal predicates*[5] in the English sentence and ask them to mark on the target side the words that express the same meaning, ii) we show them a list of *key arguments* (which correspond to the labeled syntactic heads in the English sentence) and likewise, ask them to mark on the target side the expression that best matches each key argument's meaning (marking several words is allowed), and finally iii) we ask them to *fix minor translation mistakes* in order to better reflect the source meaning. Importantly, we ask annotators to flag as **special cases** any one-to-many mappings, and for predicates, any mapping that aligns a source verb to a non-verbal predicate in the target language. We also give the option to map source heads or predicate words to *NONE* when no relevant corresponding expression in the translated sentence can be found.

[1] `https://www.deepl.com/translator`

[2] `https://github.com/explosion/spaCy`

[3] Note that validating a translation that already exists is considerably faster than generating translations from scratch, therefore annotation time and budget dropped significantly.

[4] See Appendix B for the X-SRL annotation guidelines.

[5] We ignore all source nominal predicates.

(1) a. People aren't **panicking**.
b. La gente no está **entrando en pánico**.
The people not are entered in panic.

(2) a. The account had **billed** about $6 million in 1988, according to Leading National Advertisers.
b. Das Konto hatte 1988 etwa 6 Millionen Dollar **in Rechnung gestellt**, so die Leading National Advertisers.
The account had 1988 about 6 million dollars in invoice put, so the Leading National Advertisers.

(3) a. The economy does, however , **depend** on the confidence of businesses, consumers and foreign investors .
b. Die Wirtschaft **hängt** jedoch vom Vertrauen von Unternehmen, Verbrauchern und ausländischen Investoren **ab**.
The economy hangs however from-the confidence of businesses, consumers and foreign investors off .

(4) a. But while the **New York Stock Exchange** did n't fall apart Friday as the **Dow Jones Industrial Average** plunged 190.58 points.
b. Mais si la **Bourse de New York** ne s' est pas effondrée vendredi alors que le **Dow Jones Industrial Average** a chuté de 190,58 points.
But if the Exchange of New York not Refl is not collapsed Friday when that the Dow Jones Industrial Average has fallen by 190.58 points.

Fig. 7.3 Examples of translation shifts: (1) predicate nominalization on the target side, (2) and (1) source verb converted to a light verb construction on the target side, (3) a source predicate translates to a verb with separable prefix, and (4) instances of Named Entities being translated or not to the target language.

7.3.2 Annotation Agreement.

To approximate the inter-annotator agreement, we gave the first 100 sentences to all annotators of each language pair and compute Krippendorff's alpha[6] on this subset of sentences. We obtain $\alpha_{pred_{DE}}$=0.75, $\alpha_{pred_{ES}}$=0.73, $\alpha_{pred_{FR}}$=0.78 for *predicate* and $\alpha_{role_{DE}}$=0.79, $\alpha_{role_{ES}}$=0.70, $\alpha_{role_{FR}}$=0.79 for *role labels*. This shows that the annotation method can be trusted.

7.3.3 Linguistic Validation.

We run a second annotation round where two annotators with linguistic background revalidate the instances that were flagged as **special cases** by translators during the first round (more concretely, the possible *translation shifts*). Specifically, annotators in this phase decide, for each special case, if the annotated label should be deleted or corrected. The cases could fall into one or more of the following categories[7] (see Figure 7.3 for some examples):

- **Nominalizations:** A verbal expression (predicate) in English is translated to a nominal expression in the Target (see Figure 7.3, examples (1, 2)). Since we restrict our dataset to verbal predicates we discourage the annotation of nominal predicates even when they preserve the original sense.

- **Light Verb Constructions:** This is a special case of nominalization on the target side, where a noun that corresponds to a verb in the source language is an argument of a

[6] We use the NLTK implementation with binary distance to compute the agreement of labels.

[7] This validation was performed independently, according to the annotators' language expertise. However, the annotators discussed general policies and jointly resolved difficult cases.

so-called ’light’ verb with bleached, often aspectual, meaning. In example (2), the verb *billed* is translated to *in Rechnung gestellt* (literally: ‘in invoice put’). According to Bonial et al. (2015), the nominal argument of a light verb needs a special role annotation.[8] Since there is no easy automatic method to figure out the target senses, we leave these cases for future work and do not annotate them here.

- **Separable Verb Prefixes:** In German, specific verbs must split off their prefix in certain constructions, even though this prefix crucially contributes to their meaning. In example (3), the German verb is *abhängen* which means *to depend*, while the verb *hängen* means *to hang*. Since the labeling scheme that we are using only allows us to tag one word as the head, annotators were instructed to pick the truncated stem of the verb, given that the particle is a syntactic dependent of it.

- **Multiword Expressions (MWEs):** A single source word is translated to several target words that constitute a single unit of meaning. The translators were allowed to mark more than one target word if the source word meaning could be mapped to a MWE. For these cases, if they did not fall in any of the previous three categories, and since they were manually aligned for being equivalent in meaning, we transfer the source label to the syntactic head of the marked MWE.

- **Named Entities:** are treated as special cases of MWEs. Some NEs, but not all, are (correctly) translated to the target language, which can result in a change of the argument’s head. We see both cases in example (4). When NEs are translated to the target language, we need to select the appropriate head: *Exchange* is the head of the NE in English but *Bourse* should be the head in French. We re-locate the label to the NE’s syntactic head on the target side.

The linguistic analysis highlights the importance of providing a human-validated test set – as opposed to relying on automatic projection. While the English labels are considered to be gold standard, their transfer to any target language is not straightforward and must be controlled for the mentioned cases to be considered gold standard on the target side. Accordingly, we also consider filters or refinements for the automatic projection and finally, on the basis of our validated test set, we can evaluate how accurate our automatic projection is.

[8] The noun projects its predicate-specific role set and in addition includes the governing verb with a role ARGM-LVB.

QUALITY (Q)	EN	DE	ES	FR
5	2,399	718	1,758	1,358
4	0	902	407	463
3	0	593	181	274
2	0	164	46	184
1	0	22	15	119
# Sentences Q >2	2,399	2,213	2,346	2,095
# Kept Predicates Q >2	5,217	4,086	4,376	3,770
# Kept Arguments Q >2	14,156	11,050	10,529	9,854

Table 7.2 EN shows the original numbers for the English CoNLL-09 corpus. The other three languages show the quality distribution and *predicate* and *role* annotations kept after applying the quality and linguistic filters.

7.3.4 Test Statistics

Table 7.2 shows the statistics for the final quality distribution for each of the target language datasets according to the translators' ratings. The final test sets are composed by all sentences with quality level higher than 2. We observe that after applying this filter, the three languages have roughly similar amounts of good quality sentences (between 87% and 97%) as well as similar density of annotations for both predicate and argument labels. The number of sentences that are completely 4-way parallel is 1,714 (71.45% of the original EN corpus). This confirms the intuition that DeepL generates translations that are faithful to the sources. The number of *special cases* analyzed in the second validation step were 294 (DE), 332 (ES) and 1300 (FR), of which 105, 122 and 173, respectively, were considered to be translation shifts and thus were not taken into further consideration.

7.4 Label Projection Method

The next step is to find an efficient method to automatically transfer the labels in the *train/dev* portions of the data to the target languages without loosing too many gold labels. In contrast to the test set, we cannot perform human validation on the *train/dev* sets due to the size of the data; here we are mostly interested in getting automatically *good enough* labels to train models. Usually, label projection methods (Pado, 2007; Padó and Lapata, 2009; van der Plas et al., 2011; Akbik et al., 2015; Aminian et al., 2019) rely on the intersection of *source-to-target* and *target-to-source* word alignments to transfer the labels in the least noisy manner, and this way prefer to have higher precision at the expense of lower recall. We instead take a novel approach and rely on the shared space of mBERT embeddings

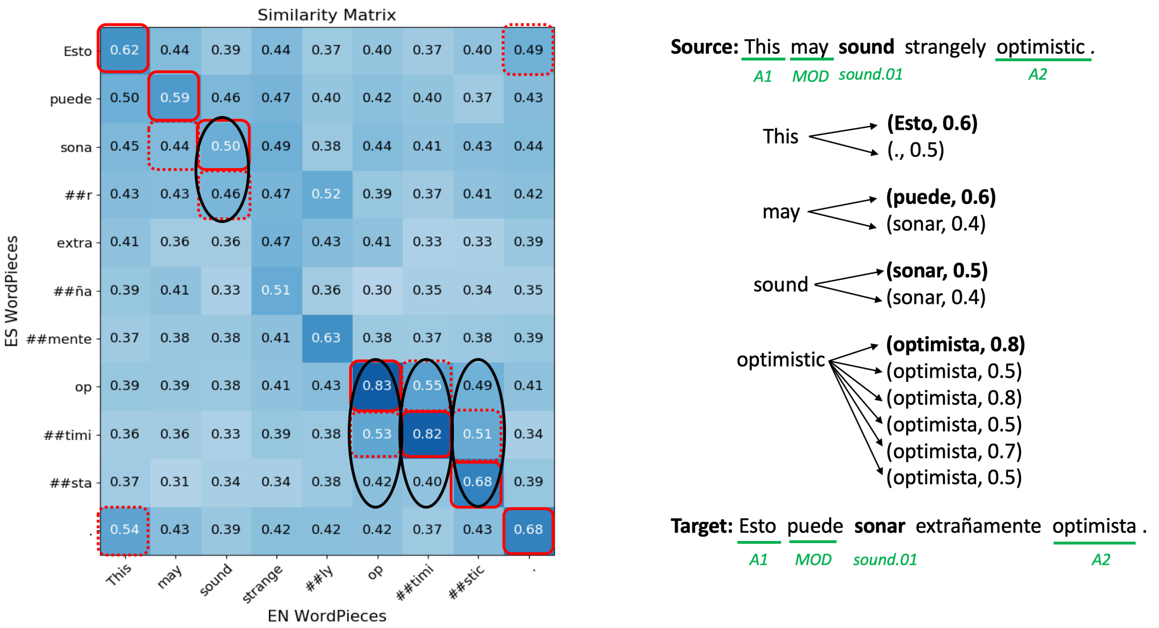


Fig. 7.4 We compute a pair-wise cosine similarity matrix to simulate word alignments. For each column, we look only at source word-pieces with an associated label and keep the top-k (k=2) most similar target-side word piece candidates (red squares). The black circles show the aligned full-word. By mapping word pieces to their full-words and applying filters we choose the final aligned target words for each source word.

Devlin et al. (2019). Specifically, we compute pair-wise cosine similarity between source and target tokens and emulate word-alignments according to this measure[9]. We show that using mBERT instead of typical word alignments dramatically improves the recall of the projected annotations, and enhanced with filters, it also achieves high enough precision, resulting in a more densely labeled target side and therefore better quality training data is expected. Additionally, previous works show that BERT contextualized representations are useful for monolingual Word Sense Disambiguation (WSD) tasks (Loureiro and Jorge, 2019; Huang et al., 2019) which lets us assume that we can rely on mBERT to find good word-level alignments across languages.

7.4.1 BERT Cosine Similarity

We start with our word tokenized parallel source $\mathbf{S} = (w_{s_0}, ..., w_{s_n})$ and target $\mathbf{T} = (w_{t_0}, ..., w_{t_m})$ sentences. Then, we use the mBERT tokenizer to obtain word-pieces and

[9] This is similar to what is done as a first step in BERTScore Zhang et al. (2020) towards computing a metric for (semantic) sentence similarity, but here we use the token-wise similarity as a guide for cross-lingual word alignments.

their corresponding vectors $\mathbf{S}' = (v_{s_0}, ..., v_{s_p})$ and $\mathbf{T}' = (v_{t_0}, ..., v_{t_q})$ respectively, where we have p source word-pieces and q target word-pieces. We compute the pairwise word-piece cosine similarity between $\mathbf{S}'$ and $\mathbf{T}'$. The cosine similarity between a source word-piece vector and a target word-piece vector is $\frac{v_s^T v_t}{||v_s||||v_t||}$ [10]. The result is a similarity matrix $\mathbf{SM}$ with p (columns) and q (rows) word-pieces (see Figure 7.4). In addition, we keep a mapping $\mathbf{S}' \rightarrow \mathbf{S}$ and $\mathbf{T}' \rightarrow \mathbf{T}$ from each of the word-piece vectors to their original respective word tokens to recover the full-word alignments when needed.

7.4.2 Word Alignments

For each column in SM, we choose the *k* most similar pairs (v_s, v_t) [11]. This is analogous to a $\mathcal{A}_{S' \rightarrow T'}$ alignment [12]. The alignment is done from full-word w_s to full-word w_t, meaning that for each v_s, instead of adding a $v_s \rightarrow v_t$ alignment, we retrieve the full-word w_s to which v_s belongs and the w_t to which v_t belongs and add a $w_s \rightarrow w_t$ alignment to the list of candidates for w_s. At this step, we still permit one-to-many mappings, which means that a w_s can be associated with more than one w_t candidates. We retain a dictionary $D = \{w_s : [(w_{t_1}, sim_{t_1})...(w_{t_x}, sim_{t_x})]|w_s \epsilon S\}$ with their associated similarity scores to keep track of the candidates. See the right hand side of Figure 7.4 for an example.

7.4.3 Alignment Modes

When projecting annotations to the translated training sections, we are confronted with the same *special cases* that we identified in the test set. In the absence of human validation, we have to define filters to eliminate noisy alignments. By only keeping the intersection of alignments $\mathcal{A}_{S \rightarrow T} \bigcap \mathcal{A}_{T \rightarrow S}$, we can get rid of a considerable amount of noisy alignments. This, however, comes at the cost of very low recall and a sparsely labeled dataset. Since we are using an accurate word-similarity measure instead of (noisier) word alignments, we can encourage higher recall by considering all $\mathcal{A}_{S \rightarrow T}$ alignments and include additional filters to get rid of noisy labels and thus preserve high precision. In (§7.5.1) we describe in detail the experiments that support this assumption.

7.4.4 Filtered Projection

First, we eliminate a considerable amount of potential noise by only looking at the w_s's that hold a predicate or argument label, while ignoring the rest. Next, for each labeled source

[10] We use the implementation of Zhang et al. (2020)

[11] k is a hyperparameter which we chose by hand. The best results were obtained with k=2.

[12] Conversely, we can simulate a $\mathcal{A}_{T' \rightarrow S'}$ alignment by defining a similar process for each row in the matrix.

Method	Lang	INTER			S2T		
		P	R	F1	P	R	F1
mBERT Only	EN-DE	86.6	49.6	63.0	69.0	76.1	72.4
	EN-ES	83.8	68.2	75.2	70.0	84.8	76.7
	EN-FR	82.7	61.8	70.7	67.7	79.5	73.1
mBERT+Filters	EN-DE	96.1	51.8	67.4	92.5	65.8	**76.9**
	EN-ES	94.0	68.8	79.4	91.9	80.7	**85.9**
	EN-FR	91.7	63.7	75.2	88.9	74.8	**81.2**

Table 7.3 Examining different projection methods on our *human-validated test set*: a) vanilla mBERT cosim (mBERT-Only) vs. adding filters (mBERT+Filters); b) INTER using intersective alignments vs. S2T using full source-to-target alignments. Using S2T alignments and applying filters yield highest F1 alignment score.

predicate, we retrieve from D the list of target candidates and keep only those that bear a verbal POS tag. If the list contains more than one target candidate we keep the one with the highest score, and if the list is empty we do not project the predicate, as it will most likely instantiate a translation shift or nominalization. Light verbs should be automatically filtered with this method, since the alignment links a verb to a noun and is therefore dropped. For the case of arguments, we also retrieve the candidates from D. In the ideal case, all candidates belong to the same w_t and we project the label to that word. Otherwise, we take the w_t with more *votes*, i.e. the w_t that was added most often to the list of candidates. In case of a tie, we turn to the similarity score and transfer the argument label to the w_t with the highest similarity[13].

7.5 Experiments and Evaluations

7.5.1 Label Projection

Intrinsic Evaluation. Since our test sets are human-validated, we can use them to measure the quality of the label projection methods we have at hand. First, we test the effectiveness of our full method (mBERT+Filters) by comparing it to vanilla cosine similarity (mBERT only) as a projection tool. We apply each method to the test sentences and evaluate the automatically assigned labels against the gold labels provided by annotators. We also show the performance differences when keeping all source to target alignments (S2T) vs. using

[13] Score aggregation would be a straightforward way of computing similarities. However, Zhang et al. (2020) mention that while cosine similarity is good to rank semantic similarity, the computed magnitude is not necessarily proportional, therefore it is not a strict metric. For this reason, we only rely on scores as a decision factor in case of ties.

	ZAP						**OURS**					
	PREDICATE			**ARGUMENT**			**PREDICATE**			**ARGUMENT**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
EN-DE	68.9	15.9	25.9	72.7	15.6	25.7	95.7	76.2	**84.9**	91.3	61.6	**73.6**
EN-ES	78.9	34.7	48.2	68.7	30.5	42.2	98.0	89.3	**93.4**	89.0	76.4	**82.2**
EN-FR	66.2	21.1	32.0	66.5	24.4	35.7	97.3	85.4	**91.0**	88.9	69.8	**78.2**

Table 7.4 We compare our best projection method with ZAP, a SOTA system for SRL label projection on our test sets. The recall of ZAP is extremely low, damaging their overall scores. In contrast, our method is very good at projecting verbal predicates and arguments.

the intersection of alignments (INTER) when projecting both predicates and arguments. In Table 7.3 the four combinations can be observed with their specific trade-offs. When using only mBERT with S2T alignments we have high recall but a very mediocre precision; when using INTER alignments we see big gains in precision at the expense of lower recall, as expected. On the other hand, mBERT+Filters obtains consistently better F1, with INTER showing similar behavior to what we observe with the vanilla method, yet with much better precision; however, using full S2T alignments *with filters* gives us the best trade-off: we still achieve around 90% precision and much better recall compared to INTER. This confirms that using S2T alignments (established using mBERT-based cosine similarity) combined with our filters are the best option for projecting labels.

Extrinsic Evaluation. Having settled our best method, we compare it with an SRL label projection software: ZAP Akbik and Vollgraf (2018) [14], which also works with the three target languages studied in this chapter. ZAP is a pipeline model that takes as input parallel $(\mathbf{S}, \mathbf{T})$ sentences, uses source syntactic and semantic parsers to obtain the annotations, and through a trained heuristic word alignment module that uses pre-computed word translation probabilities, it transfers the labels only when it considers the alignments to be valid, preferring to have fewer, but higher-quality annotations on the target side.

To compare our method to this baseline, we measure the density of the labels on the target training sets after applying both methods to project the labels[15]. Figure 7.5 shows the case of EN projected to DE where our method consistently recovers more labels from the source, resulting in a more densely annotated training set with comparable label distribution to the EN source. This trend is similar for Spanish and French (overall coverage relative to EN is: DE: 58.9%, ES: 67.3%, FR: 66.9%). To investigate more deeply why ZAP performs so poorly compared to our method, we use the test sets to measure performance. We first evaluate the capacity to transfer source predicates to the target side. Table 7.4 clearly shows that ZAP

[14] `www.github.com/zalandoresearch/zap`

[15] We consider the gold source labels for both methods, thus comparing only their projection performance.

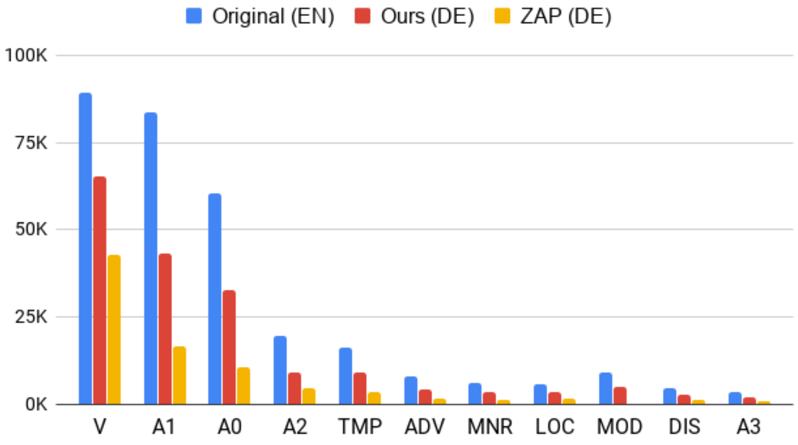


Fig. 7.5 Ten most frequent labels obtained with two label projection methods: OURS vs. ZAP - on the German train set, compared to English source annotations.

fails to transfer many predicates, perhaps because it has unreliable (or no) word-alignment probabilities for infrequent predicates and it is not fine-tuned for this domain (it was trained on Europarl). As a result, the argument scores are also very low, since for each predicate it misses, the system cannot recover any arguments. This highlights the main advantages of our method: by relying on a big multilingual language model i) we obtain high-quality word alignments featuring high precision *and* recall, and ii) we do not need to re-train for other language pairs nor different domains.

7.5.2 Training SRL Systems on X-SRL

At this point we have attested the quality of the automatic method for creating the training sets. Now, as an extrinsic evaluation, we will measure how well different models can learn from our data. Following the method described above, we achieve a large, annotated SRL dataset for three new languages, which is comparable in size, contains homogeneous annotations and is multi-way parallel. (cf. Table 7.1).

To train the models we follow Zhou and Xu (2015); He et al. (2017) in the sense that we feed the predicate in training and inference, and we process each sentence as many times as it has predicates, labeling one predicate-argument structure at a time.

mBERT fine-tuning. In all settings, we fine-tune mBERT[16]. We use batch size of 16, learning rate of $5e^{-5}$ and optimize using Adam with weight decay (Loshchilov and Hutter, 2019) and linear schedule with warmup. We train for 5 epochs on our data and pick the epoch that performs best on *dev*. Concretely, we explore three settings: The obvious baseline is i) to

[16] We use BertForTokenClassification from `https://huggingface.co/transformers/`

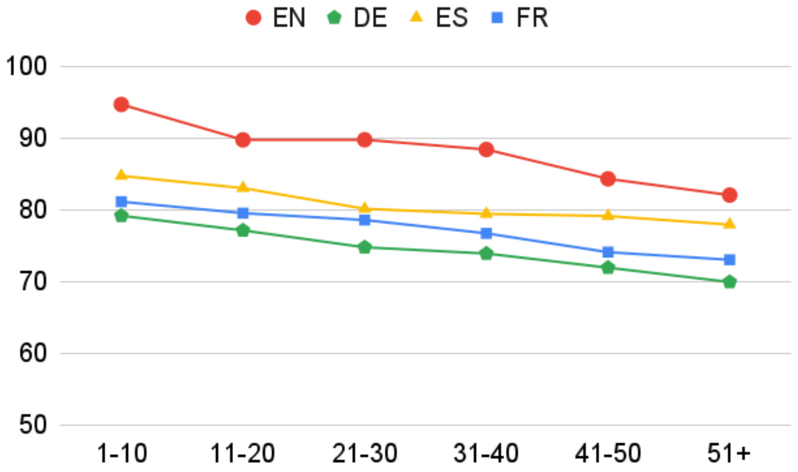


Fig. 7.6 Comparison of the F1 score that fine-tuned mBERT obtains for the four languages when broken by length of input sequences.

use only the available English high-quality labels for fine-tuning mBERT and apply zero-shot inference on the other three languages (we call this *EN-tuned*). The other two settings are ii) to fine-tune each language independently with its respective training set (*Mono*) and iii) using all the available data from the four languages to train a single model (*Multi*). Table 7.5 shows that, as expected, for the *EN-tuned* baseline, English reaches an F1 score of 91, and the other three languages can make good use of mBERT's knowledge in the zero-shot setting, reaching scores around 70. We also see that our training sets are more complete, obtaining, across the board, higher F1 scores than the training sets projected using ZAP. We observe that training on monolingual data results in improvements for all languages, and finally, the best setting is to use all data at once, improving the already robust mBERT results, and reaching scores of 77, 92, 81 and 78 for DE, EN, ES, FR respectively, about 8 points higher than the zero-shot baseline in the case of German.

Sequence Length Analysis. We also break the analysis of the transformer-based mBERT model to observe if there exists a difference in performance depending on the sequence length (similar to the two previous chapters). Although this is a completely separate dataset and a completely different model (Transformer vs LSTM), we can still observe interesting conclusions from looking at Figure 7.6: First of all, we see that the four languages behave similarly when we break the performance analysis by sequence length; more importantly, the transformer model shows a flatter descent correlated with the length of sequences. This shows support for the robustness of transformers for dealing with longer-range dependencies on sequences.

	EN		DE		ES		FR	
MODEL	ZAP	OURS	ZAP	OURS	ZAP	OURS	ZAP	OURS
mBERT EN-tuned	91.0	91.0	69.5	69.5	75.1	75.1	71.9	71.9
mBERT Mono (finetune)	91.0	91.0	58.6	76.1	64.5	80.5	59.5	77.4
mBERT Multi (finetune)	92.4	**92.9**	63.7	**77.0**	67.4	**81.1**	64.1	**78.3**

Table 7.5 F1 Score with Fine-tuning mBERT on our training data, created using ZAP vs. OUR projection method and evaluated on our test sets. We compare zero-shot (EN-tuned), mono- and multilingual settings.

MODEL	EN	DE	ES	FR
Ours (Ch. 6) [Mono]	90.9	67.6	56.2	58.1
Ours (Ch. 6) [Multi]	87.6	72.5	77.1	75.2
Cai et al. (2018) Mono	91.4	76.5	**82.6**	80.3
He et al. (2019) Mono	**92.4**	75.8	82.3	79.3
He et al. (2019) Multi	92.1	**77.3**	82.5	**80.4**

Table 7.6 F1 Score when training existing SRL models with our data and evaluating on our test. We compare monolingual (Mono) vs using all data available (Multi).

SOTA Models. Next, we choose three SRL systems that show SOTA results on CoNLL-09 and train them using our data instead. Note that our results are not comparable since our train and test sets are completely different for ES and DE; also the EN results are not comparable since we only label verbal predicates; finally, FR is not present in CoNLL-09. Table 7.6 summarizes the results. The model marked as "Ours" is the Encoder-Decoder model that was described in Chapter 6. Because it was conceived for multilingual SRL, it performs poorly when trained on monolingual data but improves significantly when trained with more data (multilingual setting). The model of Cai et al. (2018) adapts the biaffine attention scorer of Dozat and Manning (2017) to the SRL task; we note that this model is not designed for handling multilingual data, and therefore only show the monolingual results, which still achieve the best score (82.6) for ES on our test data. Finally, He et al. (2019) generalizes and enhances the biaffine attention scorer with language-specific rules that prune arguments to achieve SOTA on all languages in CoNLL-09. When training this model using our data it achieves the highest scores for EN in the *Mono* setting and for DE and FR when trained with multilingual data. In sum, using our new corpus to train multilingual SRL systems, with SOTA models and finetuning mBERT, we find evidence that the models can use the multilingual annotations for improved performance, especially for the weaker languages.

	German (*DE*)				**French (*FR*)**				**Spanish (*ES*)**			
Model [Filter]	**Full Seq**	**Word**	**Label**	**Kept %**	**Full Seq**	**Word**	**Label**	**Kept %**	**Full Seq**	**Word**	**Label**	**Kept %**
XL-BERT [All]	14.1	11.7	16.2	100	20.1	18.8	29.9	100	17.0	15.6	22.2	100
XL-BERT [$\geq$ 10]	23.3	23.2	50.1	55.9	26.3	26.8	56.4	71.1	24.6	24.7	49.0	63.6

Table 7.7 Cross-lingual (XL) system results using BLEU score on the *X-SRL Test* sets. We compute BLEU on labeled sequences (Full Seq), and separately for words and only labels. We also show scores when applying a filter on Full Seq of BLEU ≥ 10.

7.6 Cross-lingual SRL with X-SRL

Another advantage of our dataset is that it is straightforwardly suited for cross-lingual experiments. We use the same Enc-Dec model described in Section 6.3 initialized with pre-trained mBERT embeddings in the first layer to perform translation from source English sentences to SRL labeled sequences on a different target language. This time, we use our X-SRL dataset for training (c.f. Table 7.1).

We apply BLEU scores on the sequences for the three different target languages: German, French and Spanish. We emulate the evaluation setting from Section 6.4, and apply a BLEU ≥ 10 filter; however, this time we have a proper test set for which we know the reference sentences and labels on the target side, therefore we do not need to perform the back-translation step to obtain the filter, in fact, we directly perform the BLEU evaluation against the test gold sequences. By following this method, after applying the BLEU quality filter we see that we manage to retain 56%, 71%, and 63%, for DE, FR and ES respectively which presumably have a better sentence quality and, given the high label-only BLEU scores we can expect to retain most of the expected target labels.

It is important to notice that in the experiments of Chapter 6 we were dependent on a dataset created via a statistical annotation projection method followed by strict filtering for semantic correspondence, tying us to lower-density of labeled data for training. In contrast, in this Chapter we developed our own method for creating parallel training data, by transferring gold semantic labels to the non-English target size, therefore now we can cover the full pipeline for creating new SRL data on non-English languages.

More importantly, our Enc-Dec model is not a label projection method but a **labeled data generator**, which uses the multilingual lexical knowledge encoded in the contextualized representations to transfer source labels into target sentences. This means that it is not necessary to re-train from scratch an annotation projection model each time we need to generate new data or include a new language pair.

Chapter 8

Conclusions and Outlook

This thesis begins with the observation that the latest advances in SRL for English were achieved by means of deep neural networks which happen to be resource-intense architectures. Thus, in the search of emulating the gains achieved in English, we focus mainly on finding methods for improving the availability of annotated SRL data for non-English languages.

Concretely, we examine here the cases of three different languages: German, French and Spanish. As we mentioned already, although these languages are not normally considered as low-resource in NLP, the fact that they have not achieved the same improvements that we observe in the case of English, even when annotated resources are already attainable, calls for a search of strategies to augment the availability of high-quality training data. For this reason, by combining insights from neural MT, neural structured prediction models and joint multilingual training, we propose for the first time to treat SRL as a sequence-to-sequence task, with the aim of applying a flexible model that allows to consume data from multiple languages at the same time, hence automatically augmenting the availability of labeled data on the resource-poor languages. Moreover, the same Enc-Dec architecture that we use for multilingual labeling, is flexible enough be used as a data generator when trained with parallel cross-lingual SRL data, giving us an additional method for further augmenting the amount of available target language labeled data.

Our motivation for using a seq2seq model comes from the fact that this architecture contains a generative decoder which opens the possibility of creating unseen target sentences. Moreover, it already showed successful results for multilingual processing in MT tasks, as well as the possibility of generating new labeled data in cases such as syntactic dependency parsing and semantic parsing. These scenarios encouraged us to use similar methods for improving the SRL performance in lower-resource languages. Below, we summarize the contributions of this thesis as well as the insights gained with he different methods that were proposed here. We will then discuss the current limitations and potential future directions.

We will revisit our research questions, and summarize how our proposed model and created dataset addressed the questions raised in Chapter 1.

8.1 Summary

SRL as Sequence-to-Sequence. In Chapter 5, we investigate the basic formulation of PropBank SRL as a seq2seq task. We present and evaluate an Enc-Dec model with attention and copying mechanisms that encodes a source sentence and generates the same source words but this time with interleaved semantic role labels. We face specific challenges by formulating the problem in this way, such as: i) the decoding of labels and words within a single sequence; ii) generating balanced labeled brackets at the correct position; iii) avoiding repetition of tokens, and especially, iv) generating labeled sequences that perfectly match the source sentence in order to make the labeled sequence absolutely comparable. We test our outputs for these challenges and confirm that the Enc-Dec with copying mechanism is robust enough to generate sequences that, in most of the cases, avoid such problems. Next, by evaluating on the most popular English SRL span-based datasets, namely the CoNLL-05 and CoNLL-12 data, we successfully prove that the SRL task can be formulated in this manner and, although we didn't improve the SOTA at that time, we obtained robust results for English, which encouraged us to keep our research in this direction.

A Flexible Encoder-Decoder model for SRL. In Chapter 6 we present the first successful joint multilingual seq2seq model for identifying and labeling PropBank roles on different languages. This model is built on top of the basic Enc-Dec presented in Chapter 5 with architecture additions that allows it to benefit from multilingual data. Moreover, we demonstrate that our flexible architecture is capable of being trained as a data generator. Concretely, we propose a model that can be used in three different modes: monolingual, multilingual and cross-lingual. The first two modes are evaluated as SRL labeling task and the third mode is treated as a data generation task.

As expected, the neural Enc-Dec architecture is overparametrized to achieve satisfactory results in the non-English languages when trained monolingually (given the lack of a bigger annotated training corpus in such languages); however, because of the flexibility of our architecture, we experiment with different scenarios that allow us to directly augment the training data for the lower-resource languages and obtain improvements with the multilingual approach. We prove the efficacy of our enhanced monolingual and multilingual model by evaluating on the English and German CoNLL-09 datasets and on a publicly available French dataset (van der Plas et al., 2010, 2011) and demonstrate that our model improved the SOTA for English at the time of publication. We also show that training a multilingual neural Enc-Dec improves results over the monolingual baselines, as well as achieving better results when compared to the only available multilingual neural model for SRL at the time of publication (Mulcaire et al., 2018).

Evaluation Strategies for Cross-lingual SRL. When training our model with cross-lingual data we are not performing SRL labeling anymore, but SRL data generation. This calls for a separate and more detailed evaluation procedure in order to assure the validity of the data produced by our model. Thus, we propose three different stages of evaluation: by means of an automatic metric, BLEU score (*intrinsic evaluation*); by re-training semantic role labelers with our generated labeled data and observing improvements in labeling scores (*extrinsic evaluation*); and finally by performing a *human evaluation* on a stratified sample drawn from the generated labeled sentences in German. For this, we follow (He et al., 2015) to bypass the need of trained annotators with linguistic knowledge and apply a useful and fast evaluation method based on generating questions that later can be mapped to SRL labels.

The evaluation of our *cross-lingual* system shows that its filtered generated outputs can be used as additional SRL-labeled data for lower-resource languages; human evaluation also shows that the quality-filtered sentences are highly grammatical and natural, and that the generated PropBank labels can be more precise than a SOTA label projection system, namely ZAP (Akbik and Vollgraf, 2018). Moreover, we test our generated sequence as training data by re-training an SRL model using our own data vs using data generated by ZAP, and obtain the best results when using our method. Thus, our three evaluation settings confirm our hypothesis that an Enc-Dec can be used to successfully translate and label SRL sequences in a single step, without the need of a pipeline of statistical models as previous methods do.

A Method for Creating Parallel SRL Data. Our experiments of Chapter 6 use an available parallel dataset with labeled semantic roles from previous research, which is pre-filtered for preserving predicate correspondence on source and target sides (Akbik et al., 2015). However, we also note that such dataset is not fully-compatible with the monolingual labeled data that is normally used for evaluating SRL, which poses challenges for evaluating the same task when involving different languages. Therefore, to round up our proposal for cross-lingual data generation, in Chapter 7 we propose a method to obtain high-quality cross-lingual SRL data without the need of relying only on existing parallel datasets. Our approach also minimizes the need for trained human annotators and offers an alternative to the current label projection techniques that, because of the tight filtering that they need to apply, produce low-density annotations. We show how we can profit from the latest advances in neural MT and Multilingual Contextualized Language Models to propose a neural approach for transferring the English gold labels from English to other languages. This approach directly addresses the incompatibility problems that current SRL datasets face, namely the lack of label standardization across languages, as well as the imbalance in density of annotations. Even though we tested our method on German, French, and Spanish, our method is in principle extensible to other language pairs that have access to high-quality MT and mBERT.

By using DeepL to create parallel sentences to the existing English CoNLL-09 data; and multilingual BERT to simulate word alignments, we show that out-of-the-box MT and mBERT together can be used to obtain useful data for training SRL models in different languages. We evaluate our hypothesis by hiring experts in translation on the 3 different language pairs to annotate corresponding test sets. The experts see the translated test sets and validate the quality of the automatically produced sentences. Furthermore, they manually annotate the target words that semantically correspond to the original English predicates and roles (with an option to ignore the non-transferable roles). Our human annotators confirm that MT produces high-quality data that is lexically faithful to the English sources, avoiding translation shifts in the majority of cases. More importantly, by following this method we create a complete new dataset, the first multi-way parallel corpus with homogeneous and dense semantic role labels, which we name X-SRL and will be published soon as a resource in the Linguistic Data Consortium (LDC).

We test the quality of the newly created dataset by training different SOTA semantic role labelers and evaluating their performance using the human-validated tests sets. Furthermore, we demonstrate the usefulness of having more training data by comparing mBERT as a zero-shot labeler (i.e. fine-tuning it with English data and directly testing on the target languages) against a fine-tuning setting of mBERT with our data for each target language. We observe gains when using the latter setting, confirming that even such a robust architecture benefits from out new labeled dataset. In sum, we demonstrate that our dataset is an interesting resource for further exploring the capabilities of multilinguality in SRL.

8.2 Future Work

Our current Enc-Dec model can be improved by adding more automatically generated data in the data augmentation scenario, or by targeted selection in an active learning setting. Current limitations of the system may be alleviated by pre-training the model to acquire better translation knowledge from larger training data, and by developing more refined filtering methods. Currently, an advantage of our proposed model is that it does not need parallel data at inference time. Thus, promising work can also be done by aiming for augmenting the system flexibility, such as extending it to few-shot or zero-shot learning. This would alleviate the need for an initial big annotated set, and thus we would be able to use the knowledge from existing annotated languages and generate SRL data for truly resource-poor languages.

Further challenges for this novel architecture are to extend it to joint predicate and role labeling for more than one predicate at a time. By integrating predicate identification, we could perform end-to-end SRL with our Enc-Dec. This would give us more control over predicate senses, because for now we just work on argument labeling provided we already know the predicates. Another interesting approach to follow is upgrading our proposal to the latest Enc-Dec architectures that have pushed even further the SOTA of Machine Translation, such as Transformers and multi-task training.

Finally, our method for using mBERT as a means for word-alignment can be improved by using external supervision for a better lexical alignment of the pre-trained embeddings. Besides, because mBERT possesses more than 200 languages, our method is extendable to creating parallel labeled datasets for other languages, provided there is access to a good-quality MT system. Further tuning of the contextualized word representations could be performed in order to augment the semantic correspondence for languages that are typologically divergent and use it for transferring English labels into those languages. These new datasets could keep being compatible with the well-known English CoNLL-09 (as we did in this thesis) or use other kinds of high-quality source labelers to transfer annotations to other lower-resource languages. Having more languages with a compatible cross-lingual label-set may enhance the performance analysis of multilingual SRL systems.

As for our X-SRL dataset, we provided some analysis of the performance that existing architectures achieve with it, however, the published dataset is available for further exploration. X-SRL is optimal for testing cross-lingual SRL architectures and improving effectiveness in different languages with a single model, as opposed to using a pipeline of an annotation projection model followed by a sequence labeler, as it is currently done in most scenarios. Because of the label schema compatibility across languages, this resource can also be used for an deeper analysis of cross-linguality of semantic roles.

Appendix A

Cross-lingual Guidelines for Human Evaluation

A.1 Overall Sentence Annotation

Quality Mark on a scale from 1-5 how grammatical the sentence is:

1 = This sentence is completely ungrammatical.

5 = This sentence is perfectly grammatical.

Naturalness Mark on a scale from 1-5 how natural the sentence is:

1 = A native speaker would never produce such a sentence.

5 = This sentence could have been written by a native speaker.

- Mark the sentence as #NO-VERB if the marked predicate is NOT a verb.

A.2 Predicate-Argument Annotation

For each sentence:

1. The predicate of interest is given.
2. Argument Identification:
   (a) You should generate as many questions as possible using this predicate. The answers must be a sub-string [phrase] from the sentence.
   (b) Write a single Question-Answer pair per line.

(c) Once you cannot think of more Q-A pairs for this sentence, you should fill the column named "head" with the syntactic head of each answer phrase you entered in step 2c.

3. Argument Classification (Labeling):

   (a) Assign the closest possible labels to each Q-A pair according to the criteria of the Annotation Process B

A.3 Labeling Criteria

Kinds of Roles

1. **Core Roles (A0-A4)**: are supposed to be agents and patients of the sentence, and to be closely related to the action that the verb describes.

2. You can find the list of roles of interest in the Auxiliary Table at the end of this document.

3. **Modifier Roles (AM-XXX)**: are general and not tied to specific predicates. Example: `(# because AM-CAU)` will always be causal regardless of the predicate. Locations will always be places regardless of the action happening, and so on.

Annotation Process

1. Start by confirming that the given predicate is a Verb. If this is the case, write in the question field the token `#VERB`, the answer should be the predicate itself and the label should be marked as `V`.

2. As a General Rule, assign labels in this order: A0 » A1 » A2 ... » AM-XX

3. Always start asking *Who?/ What?* (A0). This label should always be assigned first (if it exists in the sentence). A0 is always the causation of a change of state (normally is an agent but in some exceptions, it could be the patient. Either way, it should be the main cause of the change of state that the verb represents).

4. Ask again the question *What?*. Now the label for this answer will be A1. A1 is normally the patient of that change of state. There are exceptions (normally with passive sentences) where you would have A1 without A0 in a sentence. Example: `The (# book A1) was given (# to A2) him`. Here the sentence does

not contain an answer to the question *Who gave the book?* therefore it has no A0 role. However, the first question you will ask is *What was given?* and the answer *The book* should be labeled as A1 (skipping A0) because it is the patient of the ’to give´ action. There are also cases where there are secondary animated participants in the patient role, for example, *John has been fooled by Mary*: *What?* Does not refer to John, but you need *Whom?*, and then John will be labeled as A1.

5. If there is no answer to *What?* in the sentence, and/or there are still questions to ask which have a patient as an answer, (someone or something affected by the action that is not only a general modifier of the situation), then use A2 - A4. The indices of core roles are assigned incrementally (for example, there can’t be an A3 if there is no A2). In the example shown in 4, the prepositional phrase to him is the answer to the question *To whom was the book given?*, therefore to is labeled with A2 because it is the second patient found in the sentence (and it is the head of the answer phrase).

6. Next, try to ask the questions *Where? Why? How?...* whose answers will be modifier AM-XXX roles.

7. Finally, consider that there some are roles that do not answer questions:

   (a) AM-NEG: this is used for the negations in the sentence. Write in the question field the token `#AM-NEG`, in the answer field the phrase that expresses negation, and the label should be marked as AM-NEG. Example: `(# You A0) will (# never AM-NEG) know.`

   (b) AM-DIS: this is used for the discourse markers in the sentence. Write in the question field the token `#AM-DIS`, in the answer field you should put the phrase that is acting as a discourse marker, and the label should be marked as AM-DIS. Also, you should apply this label for vocatives (For example, `Dear (# Mr. AM-DIS) President, ...`). In this case, write in the question field the token `#VOCATIVE`.

   (c) AM-MOD: this is used for the modal verbs in the sentence (when they are not the predicate of interest). For example, `I (# must AM-MOD) give this (# book A1) (#to A2) her`. Write in the question field the token AM-MOD, and in the answer field the answer should be the modal verb and the label will be marked as AM-MOD.

Examples

`(# I A0) tried to give (# her A2) (# a book A1), (# but she doesn't like to read AM-DIS).`

In this case, A0 is the giver (*Who?*), A1 is the thing given (*What?*) and A2 is the recipient (*To whom?*), and receives A2 because it is a secondary agent that is related to the act of giving. This tagging is following the rule of first asking who, then what, and then assigning core roles incrementally if there are more found.

A.4 Useful Remarks

Examples of most common heads

1. In the case of a Prepositional Phrase, the argument head is the preposition. E.g. `(# In AM-LOC) diesem Bereich`. Normally, this preposition (together with the rest of the phrase) would be the direct answer to the WH-Word used for the question.

2. For a Noun Phrase, the noun is the head `der Europäischen (# Kommission A1)`.

3. For verbs with a separable prefix, the head must be the main part (the stem). Example: `Ich (# rufe V) Sie an .`

Notes

1. After asking the common questions (*Who? What? To Whom?*) for agents and participants, try to find questions whose answers are Prepositional Phrases.

2. Try to state as many questions as possible, but always include the predicate in the question.

3. Always include in the answers all the relevant participants related to the given predicate.

WH-Phrase Correlation Table

Role	Question
A0 [Agent]	Who? What?
A1 [Patient]	What? Who? How much?
A2 [Patient 2]	What? How much? Where?
A3 [Patient 3]	What? Who?
A4 [Patient 4]	-
AM-DIR [Direction]	To where?
AM-LOC [Location]	Where?
AM-MNR [Manner - modify verb]	How?
AM-TMP [Temporal]	When?
AM-EXT [Extent]	How much? How?
AM-PNC [Purpose]	Why?
AM-CAU [Cause]	Why?
AM-ADV [Adverbial - modify entire sentence]	Why?
AM-DIS [Discourse Marker / Vocatives]	#AM-DIS / #VOCATIVE
AM-MOD [Modals]	#AM-MOD
AM-NEG [Negation]	#AM-NEG

Appendix B

X-SRL Dataset Annotation Guidelines

In this annotation task, you will be provided with pairs of sentences taken from the business section of the Wall Street Journal newspaper. Each pair consists of the original English source and its corresponding translation in a given target language (German, Spanish or French). Every target sentence was produced via automatic translation software. We are interested in three main aspects of the translations:

B.1 Validation of Automatically Translated Text - Overall Quality

We are interested in how much the automatic translation preserves the general meaning that the source intended to communicate. Considering the target sentence as a whole, mark on a scale from 1-5 (worst to best) how well the target translation captures the original meaning:

Quality Mark on a scale from 1-5 how grammatical the sentence is:

1 = `Ungrammatical or meaningless.`

2 = `Grammatical but does not preserve the source meaning.`

3 = `The translation has most of the source meaning but has key errors.`

4 = `This is a translation with small distortions.`

5 = `This translation could have been done by a (non-professional) human.`

B.2 Key Predicates

We are also interested in the preservation of source events (verbs) in the target sentence. We are aware that sometimes the best translation of a sentence does not contain a direct translation of each verb. For our specific purposes, however, we would like to have sentence pairs that contain them in both the source and the target. We will provide a list of the main verbs of the source and you should indicate if there is a one-to-one corresponding verb on the target side. We only aim to keep verbal predicates, therefore target corresponding nouns should be ignored even if they are a correct translation.

B.3 Key Arguments

We also want to measure how many keywords (arguments) from the source are still preserved in the target sentence. We will provide a list of such words taken from the source and you should indicate if there is a corresponding word on the target side. Note that in this case, rather than a one-to-one correspondence, we want to find semantically related sub-phrases inside the sentence (please see the "Difficult Examples" and "Important notes" sections for more details). We provide you with:

B.4 Annotation Elements

1. **Source sentence:** the original English sentence. Example:

   Heavy selling of Standard & Poor's 500-stock index futures in Chicago relentlessly beat stocks downward.

2. **Target sentence:** the automatic translation of the source (in English, German or French). Example:

   Starke Verkäufe von Standard & Poor's 500-Aktienindex-Futures in Chicago schlagen Aktien unerbittlich nach unten.

3. **Indexed Source:** the same English source but indicating, on the left side of each word, in which position it is located inside the source sentence. This will help you to identify the specific words we are interested in. Example:

   1_Heavy 2_selling 3_of 4_Standard 5_ 6_Poor 7_'s 8_500 9_-
   - 10_stock 11_index 12_futures 13_in 14_Chicago 15_relentlessly **16_beat** 17_stocks 18_downward 19_.

4. **Indexed Target:** the same translated sentence but indicating, on the left side of each word, in which position it is located inside the target sentence. Example:

   `1_Starke 2_Verkäufe 3_von 4_Standard 5_ 6_Poor 7_'s 8_500 9_- 10_Aktienindex-Futures 11_in 12_Chicago` **`13_schlagen`** `14_Aktien 15_unerbittlich 16_nach 17_unten 18_.`

5. **Source Predicates:** the relevant events (verbs) that occur in the source sentence. Example:

   `16_beat`

6. **Source Arguments (keywords):** these are words that bear most of the semantic meaning of the source, therefore we would like to find them as well in the target. Example:

   `2_selling`

   `15_relentlessly`

   `17_stocks`

   `18_downward`

7. **Propose an Alternative Translation:** if you think that the whole translation should be rephrased, please provide it in this space.

B.5 Difficult Examples

1. When an Argument (keyword) is a **preposition or a syntactic marker**, you should read the complete sub-phrase that is associated to it and try to find the translated sub-phrase on the target side: we are interested in matching the same semantic meaning, even if the keywords do not correspond. Example:

   `As the market plunged 90 points, it barely managed` ***`to`*** *`stay this side of chaos.`*

   `Als der Markt 90 Punkte einbrach, gelang es ihr kaum ,` ***`auf`*** *`dieser Seite des Chaos zu bleiben.`*

   `11_to ⟶ 13_auf`

2. When in doubt, it is preferable that you fill-in more than one word in the EQUIVALENT IN TARGET column, and mark it as a **special case**. The special cases that you find will be analyzed on a further steps by other annotators. For example:

`18_downward ⟶ 16_nach 17_unten`

`15_surrendered ⟶ 2_gaben 21_auf`

`20_unable ⟶ 27_nicht 28_in 29_der 30_Lage`

3. Sometimes the English side has a preposition but the target refers directly to a main noun, therefore you should pair it with the noun. Example:

   `Martha` **`gave`** `a book` *`to Jonas`* `.`

   `Martha` **`gab`** *`Jonas`* `ein Buch .`

   `5_to ⟶ 3_Jonas`

4. Sometimes the proper names in the text are not marked properly (even in the source). Please provide always the complete proper name (even when they are many words). Example, for "The New York Stock Exchange" you should answer like the following:

   `7_Exchange ⟶ 4_New 5_Yorker 6_Börse`

B.6 Important Notes

1. If the translation quality is 2 or 1 then you can ignore the predicates and arguments and you don't have to do the translation fix (such examples will be dropped).

2. If there is no equivalent predicate in the target, you should indicate this by putting the token `<NONE>` on the target predicate cell

3. Avoid spending too much time thinking about "the best possible translation", if you think the sentence should be improved but you can't think of it right away, please fill-in the token `<FIX-LATER>` and skip it (for now). Later go back and improve it if you find the time.

4. If you find concepts whose translation you don't know how to handle, add the Sentence-ID to your log so we can comment on it later.

5. If you found that a predicate or an argument could be present by fixing a mistaken word (just a one-word fix), please fill-in `<FIX-ARG>` and on the immediate cell to the right write the word that you consider most appropriate.

6. Please ALWAYS keep the original word indices, even after you provided different translations for the sentences or you fix word-segmentation issues (e.g. two words have a single index). Mark this cases on the log and they will be revisited later to be fixed.

Appendix C

Data Management

Resources for Chapters 5 and 6. we have a heiDATA repository available at https://doi.org/10.11588/data/TOI9NQ that contains the code for reproducing the monolingual, multilingual and cross-lingual experiments presented in Chapters 5 and 6 which correspond to the ACL workshop paper (Daza and Frank, 2018) and EMNLP conference paper (Daza and Frank, 2019) respectively. As for the datasets we used for the experiments:

- The span-based English SRL data is provided by the CoNLL-2005 Shared Task (Carreras and Màrquez, 2005) available at http://www.lsi.upc.edu/~srlconll/. However, the original words are from the Penn Treebank dataset which is not publicly available, but can be purchased at https://catalog.ldc.upenn.edu/LDC99T42.
- The CoNLL-12 shared task English dataset (Pradhan et al., 2012) is available at http://conll.cemantix.org/2012/data.html.
- The dependency-based English SRL data is provided by the CoNLL-2009 Shared Task (Hajič et al., 2009). The original task description is available at https://ufal.mff.cuni.cz/conll2009-st/task-description.html. However, the datasets were splitted in two parts and integrated into the Linguistic Data Consortium (LDC). The German, Czech, Spanish, Catalan and Japanese datasets are available at https://catalog.ldc.upenn.edu/LDC2012T03; whereas the English and Chinese datasets are available at https://catalog.ldc.upenn.edu/LDC2012T04.
- The dependency-based SRL parallel English-French corpus (van der Plas et al., 2010, 2011) was part of the CLaSSic project and is available at http://www.classic-project.org/.
- The parallel datasets we used for training the cross-lingual versions of the model were Europarl (Koehn et al., 2003), available at https://opus.nlpl.eu/Europarl.php

and the United Nations parallel corpus (Ziemski et al., 2016), can be taken from `https://conferences.unite.un.org/UNCorpus/`. We additionally used the training data from Akbik et al. (2015), which was requested to the authors of such publication.

- To train models you need to first pre-process the CoNLL data using the `CoNLL_to_JSON.py` script. For example, `python pre_processing/CoNLL_to_JSON.py -source_file datasets/raw/CoNLL2009-ST-English-trial.txt -output_file datasets/json/EN_conll09_trial.json -dataset_type mono -src_lang "<EN>" -token_type CoNLL09`. The results will be written in the `datasets` folder.
- Once you have a suitable data you train a model with the AllenNLP 0.8.2 framework (available at: `https://github.com/allenai/allennlp/tree/v0.8.2`). This requires you to have a configuration file that uses the models provided in our repository. For example, to train an English monolingual model one must run the following command:
  `allennlp train training_config/test/en_copynet-srl-conll09.json -s saved_models/example-srl-en/ -include-package src`.
- For a more thorough description of how to handle the code and run the specific models trained in this thesis you can also consult the repository at the Heidelberg-NLP GitHub available at `https://github.com/Heidelberg-NLP/SRL-S2S`.

Resources for Chapter 7 The heiDATA repository is available at `https://doi.org/10.11588/data/HVXXIJ` and contains the code for reproducing experiments presented in Chapter 7 which includes the BERT-based annotation projection code, as explained in the EMNLP paper (Daza and Frank, 2020). In particular,

- The English side of annotations were taken from the English CoNLL-09 corpus.
- The translations of the English text from that corpus to Spanish, German and French were obtained by using a DeepL Pro software subscription `https://www.deepl.com/pro#single`.
- The projections of annotations from English to any target language can be done by executing our `project_srl_annotations.py` script. A concrete example for projecting English labels to Spanish sentences would be: `python project_srl_annotations.py -src_file trial_data/X-SRL_Gold_EN.conll -tgt_file trial_data/ES_template_trial.syn.conll -tgt_lang ES -align_mode S2T`.

- For more detailed information of our code used for the experiments of Chapter 7 you can visit the Heidelberg-NLP Github repository at https://github.com/Heidelberg-NLP/xsrl_mbert_aligner.
- The final X-SRL Dataset (which follows the method and code published here) is going to be part of the Linguistic Data Consortium (LDC) at around Summer 2021.

List of Figures

List of Tables

List of Abbreviations

ANNs Artificial Neural Networks 18

BERT Bi-directional Encoder Representations from Transformers 34

Bi-LSTM Bidirectional LSTM 23, 24, 40–43, 67, 69, 82, 139

DL Deep Learning 2, 18

ELMo Embeddings from Language Models 34

Enc-Dec Encoder-Decoder 5, 6, 24–29, 31, 64, 65, 67, 68, 72, 73, 79–84, 86, 89, 93, 97, 101, 102, 117, 119–123, 139, 140

FFNN Feed-Forward Neural Network 19

GRU Gated Recurrent Unit 23

LM Language Model 20

LSTM Long Short-Term Memory 23

ML Machine Learning 1, 2, 5, 16, 18

MLP Multi-layer Perceptron 19, 24, 27, 28

MT Machine Translation v, vi, 1, 2, 24–26, 82, 90, 92, 93, 103, 119, 121, 122

NLP Natural Language Processing v, 1, 2, 5, 18, 19, 24, 33, 46, 119

NMT Neural Machine Translation 5, 26, 64

RNN Recurrent Neural Network 18, 21

RNNs Recurrent Neural Networks 21–23

seq2seq Sequence-to-Sequence 5, 6, 24, 29, 31–33, 63, 64, 67, 73, 74, 79, 85, 86, 88, 101, 102, 104, 119, 120

SOTA state-of-the-art. v, 5, 6, 17, 33, 35, 36, 38–40, 42, 43, 45, 55, 64, 65, 67, 80, 85–87, 89, 97, 98, 103, 120–122

SRL Semantic Role Labeling v, vi, 1, 5–7, 9, 13, 37–39, 45, 48, 61, 63, 67, 72, 79, 80, 83, 103, 119–121, 140

WSD Word Sense Disambiguation 53

References

Akbik, A., Blythe, D., and Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Akbik, A., Chiticariu, L., Danilevsky, M., Li, Y., Vaithyanathan, S., and Zhu, H. (2015). Generating high quality proposition banks for multilingual semantic role labeling. *ACL-IJCNLP 2015*, 1:397–418.

Akbik, A. and Vollgraf, R. (2018). ZAP: An open-source multilingual annotation projection framework. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Aminian, M., Rasooli, M. S., and Diab, M. (2017). Transferring semantic roles using translation and syntactic information. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 13–19, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Aminian, M., Rasooli, M. S., and Diab, M. (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*, pages 200–210, Gothenburg, Sweden. Association for Computational Linguistics.

Ammar, W., Mulcaire, G., Ballesteros, M., Dyer, C., and Smith, N. A. (2016). Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.

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

Ba, J., Kiros, J., and Hinton, G. E. (2016). Layer normalization. *ArXiv*, abs/1607.06450.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. 3rd International Conference on Learning Representations, ICLR 2015 ; Conference date: 07-05-2015 Through 09-05-2015.

Baker, C. F., Fillmore, C. J., and Cronin, B. (2003). The structure of the FrameNet database. 16(3):281–296.

Baker, C. F., Fillmore, C. J., Lowe, J. B., Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The Berkeley FrameNet Project. In *Proceedings of the 17th international conference on Computational linguistics*, volume 1, page 86, Morristown, NJ, USA. Association for Computational Linguistics.

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., and Schneider, N. (2013). Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Bender, E. (2013). *Linguistic Fundamentals for Natural Language Processing: 100 Essentials from Morphology and Syntax*. Morgan and Claypool Publishers.

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. 3(null):1137–1155.

Berant, J., Srikumar, V., Chen, P.-C., Vander Linden, A., Harding, B., Huang, B., Clark, P., and Manning, C. D. (2014). Modeling biological processes for reading comprehension. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1499–1510, Doha, Qatar. Association for Computational Linguistics.

Björkelund, A., Hafdell, L., and Nugues, P. (2009). Multilingual semantic role labeling. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 43–48, Boulder, Colorado. Association for Computational Linguistics.

Bonial, C., Hwang, J. D., Bonn, J., Conger, K., Babko-Malaya, O., and Palmer, M. (2015). :english PropBank annotation guidelines. In *Center for Computational Language and Education Research Institute of Cognitive Science University of Colorado at Boulder*.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Burchardt, A., Erk, K., Frank, A., Kowalski, A., Padó, S., and Pinkal, M. (2006). The SALSA corpus: a german corpus resource for lexical semantics. In *In Proceedings of LREC 2006*.

Burchardt, A., Erk, K., Frank, A., Kowalski, A., Padó, S., and Pinkal, M. (2009). *Using FrameNet for the semantic analysis of German: annotation, representation, and automation*.

Cai, J., He, S., Li, Z., and Zhao, H. (2018). A full end-to-end semantic role labeler, syntactic-agnostic over syntactic-aware? In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2753–2765, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Cai, R. and Lapata, M. (2019). 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)*, pages 1018–1027, Hong Kong, China. Association for Computational Linguistics.

Carreras, X. and Màrquez, L. (2004). Introduction to the CoNLL-2004 shared task: Semantic role labeling. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 89–97, Boston, Massachusetts, USA. Association for Computational Linguistics.

Carreras, X. and Màrquez, L. (2005). Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164, Ann Arbor, Michigan. Association for Computational Linguistics.

Che, W., Liu, Y., Wang, Y., Zheng, B., and Liu, T. (2018). Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium. Association for Computational Linguistics.

Chen, Y.-N., Wang, W., and Rudnicky, A. (2013). Unsupervised induction and filling of semantic slots for spoken dialogue systems using frame-semantic parsing. pages 120–125.

Chisholm, A., Radford, W., and Hachey, B. (2017). Learning to generate one-sentence biographies from Wikidata. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 633–642, Valencia, Spain. Association for Computational Linguistics.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Choi, J. D. and McCallum, A. (2013). Transition-based dependency parsing with selectional branching. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1052–1062, Sofia, Bulgaria. Association for Computational Linguistics.

Collins, M. (1997). Three generative, lexicalised models for statistical parsing. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 16–23, Madrid, Spain. Association for Computational Linguistics.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12(null):2493–2537.

Daza, A. and Frank, A. (2018). A sequence-to-sequence model for semantic role labeling. In *Proceedings of The Third Workshop on Representation Learning for NLP*, pages 207–216, Melbourne, Australia. Association for Computational Linguistics.

Daza, A. and Frank, A. (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)*, pages 603–615, Hong Kong, China. Association for Computational Linguistics.

Daza, A. and Frank, A. (2020). X-SRL: A parallel cross-lingual semantic role labeling dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3904–3914, Online. Association for Computational Linguistics.

DeNero, J. and Liang, P. (2007). The berkeley aligner. `https://code.google.com/archive/p/berkeleyaligner/`.

Denkowski, M. and Lavie, A. (2011). Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91, Edinburgh, Scotland. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dong, D., Wu, H., He, W., Yu, D., and Wang, H. (2015). Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

Dong, L. and Lapata, M. (2016). Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin, Germany. Association for Computational Linguistics.

Dowty, D. (1991). Thematic proto-roles and argument selection. *Language*, 67(3):547–619.

Dozat, T. and Manning, C. D. (2017). Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. (2015). Convolutional networks on graphs for learning molecular fingerprints. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28, pages 2224–2232. Curran Associates, Inc.

Eger, S., Daxenberger, J., Stab, C., and Gurevych, I. (2018). Cross-lingual argumentation mining: Machine translation (and a bit of projection) is all you need! In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 831–844, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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

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

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. MIT Press.

Fillmore, C. J. (1976). Frame semantics and the nature of language. *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, 280(1):20–32.

Firat, O., Cho, K., and Bengio, Y. (2016a). Multi-way, multilingual neural machine translation with a shared attention mechanism. In Knight, K., Nenkova, A., and Rambow, O., editors, *HLT-NAACL*, pages 866–875. The Association for Computational Linguistics.

Firat, O., Sankaran, B., Al-Onaizan, Y., Yarman Vural, F. T., and Cho, K. (2016b). Zero-resource translation with multi-lingual neural machine translation. pages 268–277.

FitzGerald, N., Täckström, O., Ganchev, K., and Das, D. (2015). Semantic role labeling with neural network factors. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 960–970, Lisbon, Portugal. Association for Computational Linguistics.

Fürstenau, H. and Lapata, M. (2009). Semi-supervised semantic role labeling. In *EACL*.

Gildea, D. and Jurafsky, D. (2000). Automatic labeling of semantic roles. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 512–520, Hong Kong. Association for Computational Linguistics.

Goldberg, Y. (2017). *Neural Network Methods for Natural Language Processing*. Morgan Claypool Publishers.

Graves, A. (2012). *Supervised sequence labelling with recurrent neural networks*. Pre-Print on webpage at: `https://www.cs.toronto.edu/~graves/preprint.pdf`.

Grenager, T. and Manning, C. D. (2006). Unsupervised discovery of a statistical verb lexicon. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 1–8, Sydney, Australia. Association for Computational Linguistics.

Gu, J., Lu, Z., Li, H., and Li, V. O. (2016). Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M. A., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Štěpánek, J., Straňák, P., Surdeanu, M., Xue, N., and Zhang, Y. (2009). The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, CoNLL '09, pages 1–18, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hajič, J., Panevová, J., Hajičová, E., Sgall, P., Pajas, P., Štepánek, J., Havelka, J., Mikulová, M., and Zabokrtsky, Z. (2006). Prague treebank 2.0.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

He, L., Lee, K., Lewis, M., and Zettlemoyer, L. (2017). Deep semantic role labeling: What works and what's next. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

He, L., Lewis, M., and Zettlemoyer, L. (2015). Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653, Lisbon, Portugal. Association for Computational Linguistics.

He, S., Li, Z., and Zhao, H. (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)*, pages 5350–5359, Hong Kong, China. Association for Computational Linguistics.

Hermann, K. M., Das, D., Weston, J., and Ganchev, K. (2014). Semantic frame identification with distributed word representations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1448–1458, Baltimore, Maryland. Association for Computational Linguistics.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9:1735–80.

Hofmann, T. and Puzicha, J. (1998). Statistical models for co-occurrence data. Technical report, USA.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Huang, L., Sun, C., Qiu, X., and Huang, X. (2019). GlossBERT: BERT for word sense disambiguation with gloss knowledge. 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)*, pages 3509–3514, Hong Kong, China. Association for Computational Linguistics.

Hwa, R., Resnik, P., Weinberg, A., Cabezas, C., and Kolak, O. (2005). Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11:311–325.

Johansson, R. and Nugues, P. (2008). The effect of syntactic representation on semantic role labeling. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 393–400, Manchester, UK. Coling 2008 Organizing Committee.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., Hughes, M., and Dean, J. (2017). Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Joulin, A., Bojanowski, P., Mikolov, T., Jégou, H., and Grave, E. (2018). Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2984, Brussels, Belgium. Association for Computational Linguistics.

Jurafsky, D. and Martin, J. H. (2019). *Speech and Language Processing (3er ed. Draft)*.

Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Karpathy, A. and Fei-Fei, L. (2017). Deep visual-semantic alignments for generating image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:664–676.

Kawahara, D., Kurohashi, S., and Hasida, K. (2002). Construction of a japanese relevance-tagged corpus.

Khan, A., Salim, N., and Jaya Kumar, Y. (2015). A framework for multi-document abstractive summarization based on semantic role labelling. *Appl. Soft Comput.*, 30(C):737–747.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Kipf, T. N. and Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR '17.

Kipper-Schuler, K. (2006). Verbnet: A broad-coverage, comprehensive verb lexicon.

Klein, G., Kim, Y., Deng, Y., Senellart, J., and Rush, A. (2017). OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.

Konstas, I., Iyer, S., Yatskar, M., Choi, Y., and Zettlemoyer, L. (2017). Neural AMR: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157, Vancouver, Canada. Association for Computational Linguistics.

Koomen, P., Punyakanok, V., Roth, D., and Yih, W.-t. (2005). Generalized inference with multiple semantic role labeling systems. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 181–184, Ann Arbor, Michigan. Association for Computational Linguistics.

Kozhevnikov, M. and Titov, I. (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)*, pages 1190–1200, Sofia, Bulgaria. Association for Computational Linguistics.

Krippendorff, K. (1980). *Content Analysis: An Introduction to Its Methodology*. Sage.

Lample, G., Conneau, A., Denoyer, L., and Ranzato, M. (2017). Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Lang, J. and Lapata, M. (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*, pages 939–947, Los Angeles, California. Association for Computational Linguistics.

Levin, B. (1993). *English Verb Classes and Alternations: A Preliminary Investigation*.

Levin, B. (2019). On Dowty's 'Thematic Proto-roles and Argument Selection'. `https://web.stanford.edu/~bclevin/dowty19fin.pdf`.

Lewis, M., He, L., and Zettlemoyer, L. (2015). Joint A* CCG parsing and semantic role labelling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1444–1454, Lisbon, Portugal. Association for Computational Linguistics.

Liu, D. and Gildea, D. (2010). Semantic role features for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 716–724, Stroudsburg, PA, USA. Association for Computational Linguistics.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2020). Ro{bert}a: A robustly optimized {bert} pretraining approach.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Loureiro, D. and Jorge, A. (2019). Language modelling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5682–5691, Florence, Italy. Association for Computational Linguistics.

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Magerman, D. M. (1994). *Natural Language Parsing as Statistical Pattern Recognition*. PhD thesis, Stanford, CA, USA.

Marcheggiani, D., Bastings, J., and Titov, I. (2018). Exploiting semantics in neural machine translation with graph convolutional networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 486–492, New Orleans, Louisiana. Association for Computational Linguistics.

Marcheggiani, D., Frolov, A., and Titov, I. (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)*, pages 411–420, Vancouver, Canada. Association for Computational Linguistics.

Marcheggiani, D. and Titov, I. (2017). Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark. Association for Computational Linguistics.

Marcu, D. and Wong, D. (2002). A phrase-based,joint probability model for statistical machine translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 133–139. Association for Computational Linguistics.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313–330.

Merlo, P. and Van Der Plas, L. (2009). Abstraction and generalisation in semantic role labels: PropBank, VerbNet or both? In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 288–296, Suntec, Singapore. Association for Computational Linguistics.

Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B., and Grishman, R. (2004). The NomBank project: An interim report. In *Proceedings of the Workshop Frontiers in Corpus Annotation at HLT-NAACL 2004*, pages 24–31, Boston, Massachusetts, USA. Association for Computational Linguistics.

Mihaylov, T. and Frank, A. (2019). Discourse-Aware Semantic Self-Attention For Narrative Reading Comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR*, 2013.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013b). Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Mulcaire, P., Swayamdipta, S., and Smith, N. A. (2018). Polyglot semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 667–672, Melbourne, Australia. Association for Computational Linguistics.

Nivre, J., Agić, Ž., Ahrenberg, L., Aranzabe, M. J., Asahara, M., Atutxa, A., Ballesteros, M., Bauer, J., Bengoetxea, K., Berzak, Y., Bhat, R. A., Bick, E., Börstell, C., Bosco, C., Bouma, G., Bowman, S., Cebiroğlu Eryiğit, G., Celano, G. G. A., Chalub, F., Çöltekin, Ç., Connor, M., Davidson, E., de Marneffe, M.-C., Diaz de Ilarraza, A., Dobrovoljc, K., Dozat, T., Droganova, K., Dwivedi, P., Eli, M., Erjavec, T., Farkas, R., Foster, J., Freitas, C., Gajdošová, K., Galbraith, D., Garcia, M., Gärdenfors, M., Garza, S., Ginter, F., Goenaga, I., Gojenola, K., Gökırmak, M., Goldberg, Y., Gómez Guinovart, X., Gonzáles Saavedra, B., Grioni, M., Grūzītis, N., Guillaume, B., Hajič, J., Hà Mỹ, L., Haug, D., Hladká, B., Ion, R., Irimia, E., Johannsen, A., Jørgensen, F., Kaşıkara, H., Kanayama, H., Kanerva, J., Katz, B., Kenney, J., Kotsyba, N., Krek, S., Laippala, V., Lam, L., Lê Hồng, P., Lenci, A., Ljubešić, N., Lyashevskaya, O., Lynn, T., Makazhanov, A., Manning, C., Mărănduc, C., Mareček, D., Martínez Alonso, H., Martins, A., Mašek, J., Matsumoto, Y., McDonald, R., Missilä, A., Mititelu, V., Miyao, Y., Montemagni, S., Mori, K. S., Mori, S., Moskalevskyi, B., Muischnek, K., Mustafina, N., Müürisep, K., Nguyễn Thị, L., Nguyễn Thị Minh, H., Nikolaev, V., Nurmi, H., Osenova, P., Östling, R., Øvrelid, L., Paiva, V., Pascual, E., Passarotti, M., Perez, C.-A., Petrov, S., Piitulainen, J., Plank, B., Popel, M., Pretkalniņa, L., Prokopidis, P., Puolakainen, T., Pyysalo, S., Rademaker, A., Ramasamy, L., Real, L., Rituma, L., Rosa, R., Saleh, S., Saulīte, B., Schuster, S., Seeker, W., Seraji, M., Shakurova, L., Shen, M., Silveira, N., Simi, M., Simionescu, R., Simkó, K., Šimková, M., Simov, K., Smith, A., Spadine, C., Suhr, A., Sulubacak, U., Szántó, Z., Tanaka, T., Tsarfaty, R., Tyers, F., Uematsu, S., Uria, L., van Noord, G., Varga, V., Vincze, V., Wallin, L., Wang, J. X., Washington, J. N., Wirén, M., Žabokrtský, Z., Zeldes, A., Zeman, D., and Zhu, H. (2016). Universal dependencies 1.4. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Ouchi, H., Shindo, H., and Matsumoto, Y. (2018). A span selection model for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1630–1642.

Pado, S. (2007). *Cross-lingual Annotation Projection Models for Semantic Role Labeling*. PhD thesis.

Padó, S. and Lapata, M. (2009). Cross-lingual annotation projection of semantic roles. *J. Artif. Int. Res.*, 36(1):307–340.

Palmer, M., Kingsbury, P., and Gildea, D. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Paul, D. and Frank, A. (2020). Social commonsense reasoning with multi-head knowledge attention. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2969–2980, Online. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (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)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Pradhan, S., Hacioglu, K., Ward, W., Martin, J. H., and Jurafsky, D. (2005). Semantic role chunking combining complementary syntactic views. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 217–220, Ann Arbor, Michigan. Association for Computational Linguistics.

Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., and Zhang, Y. (2012). Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, CoNLL '12, pages 1–40, Stroudsburg, PA, USA. Association for Computational Linguistics.

Punyakanok, V., Roth, D., and Yih, W.-t. (2008). The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287.

Radford, A., Narasimhan, K., Tim, S., and Sutskever, I. (2018). Improving language understanding with unsupervised learning. In *Technical report, OpenAI*.

Rajpurkar, P., Jia, R., and Liang, P. (2018). Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Roth, M. and Lapata, M. (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)*, pages 1192–1202, Berlin, Germany. Association for Computational Linguistics.

Ruder, S., Søgaard, A., and Vulić, I. (2019). Unsupervised cross-lingual representation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 31–38, Florence, Italy. Association for Computational Linguistics.

Ruppenhofer, J., Sporleder, C., Morante, R., Baker, C., and Palmer, M. (2009). SemEval-2010 task 10: Linking events and their participants in discourse. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pages 106–111, Boulder, Colorado. Association for Computational Linguistics.

Saito, H., Kuboya, S., Sone, T., Tagami, H., and Ohara, K. (2008). The japanese framenet software tools.

Schwenk, H., Dchelotte, D., and Gauvain, J.-L. (2006). Continuous space language models for statistical machine translation. In *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, COLING-ACL '06, page 723–730, USA. Association for Computational Linguistics.

See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Shareghi, E., Gerz, D., Vulić, I., and Korhonen, A. (2019). Show some love to your n-grams: A bit of progress and stronger n-gram language modeling baselines. 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)*, pages 4113–4118, Minneapolis, Minnesota. Association for Computational Linguistics.

Shi, P. and Lin, J. (2019). Simple BERT models for relation extraction and semantic role labeling. *CoRR*, abs/1904.05255.

Smith, N. A. (2019). Contextual word representations: A contextual introduction. *CoRR*, abs/1902.06006.

Snover, M., Dorr, B. J., Schwartz, R., and Micciulla, L. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Stowe, K., Moeller, S., Michaelis, L., and Palmer, M. (2019). Linguistic analysis improves neural metaphor detection. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 362–371, Hong Kong, China. Association for Computational Linguistics.

Strubell, E., Verga, P., Andor, D., Weiss, D., and McCallum, A. (2018). Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium. Association for Computational Linguistics.

Subirats, C. and Sato, H. (2003). Surprise! spanish framenet. In *In Proceedings of the Workshop on Frame Semantics at the XVII. International Congress of Linguists*.

Surdeanu, M., Johansson, R., Meyers, A., Màrquez, L., and Nivre, J. (2008). The CoNLL 2008 shared task on joint parsing of syntactic and semantic dependencies. In *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 159–177, Manchester, England. Coling 2008 Organizing Committee.

Surdeanu, M., Màrquez, L., Carreras, X., and Comas, P. R. (2007). Combination strategies for semantic role labeling. *J. Artif. Int. Res.*, 29(1):105–151.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Täckström, O., Ganchev, K., and Das, D. (2015). Efficient inference and structured learning for semantic role labeling. *Transactions of the Association for Computational Linguistics*, 3:29–41.

Tan, Z., Wang, M., Xie, J., Chen, Y., and Shi, X. (2018). Deep semantic role labeling with self-attention. In *AAAI Conference on Artificial Intelligence*.

Taulé, M., Martí, M. A., Recasens, M., and Computació, C. D. L. I. (2008). Ancora: Multi level annotated corpora for catalan and. In *Spanish. 6th International Conference on Language Resources and Evaluation, Marrakesh*.

Tiedemann, J. and Agic, Z. (2016). Synthetic treebanking for cross-lingual dependency parsing. *J. Artif. Intell. Res.*, 55:209–248.

Titov, I. and Klementiev, A. (2012). A Bayesian approach to unsupervised semantic role induction. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 12–22, Avignon, France. Association for Computational Linguistics.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Toutanova, K., Haghighi, A., and Manning, C. D. (2008). A global joint model for semantic role labeling. *Computational Linguistics*, 34(2):161–191.

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *CoRR*, abs/1003.1141.

Tyers, F., Sheyanova, M., Martynova, A., Stepachev, P., and Vinogorodskiy, K. (2018). Multi-source synthetic treebank creation for improved cross-lingual dependency parsing. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 144–150, Brussels, Belgium. Association for Computational Linguistics.

van der Plas, L., Apidianaki, M., and Chen, C. (2014). Global methods for cross-lingual semantic role and predicate labelling. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1279–1290, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

van der Plas, L., Merlo, P., and Henderson, J. (2011). Scaling up automatic cross-lingual semantic role annotation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 299–304, Portland, Oregon, USA. Association for Computational Linguistics.

van der Plas, L., Samardzic, T., and Merlo, P. (2010). Cross-lingual validity of propbank in the manual annotation of french. In *Linguistic Annotation Workshop*, pages 113–117. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Vickrey, D. and Koller, D. (2008). Sentence simplification for semantic role labeling. In *Proceedings of ACL-08: HLT*, pages 344–352, Columbus, Ohio. Association for Computational Linguistics.

Vinyals, O., Kaiser, L., Koo, T., Petrov, S., Sutskever, I., and Hinton, G. (2015). Grammar as a foreign language. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 2773–2781, Cambridge, MA, USA. MIT Press.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Wang, C., Xue, N., and Pradhan, S. (2015a). A Transition-based Algorithm for AMR Parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 366–375. Association for Computational Linguistics.

Wang, H., Bansal, M., Gimpel, K., and Mcallester, D. (2015b). Machine comprehension with syntax, frames, and semantics. In *In Proceedings of ACL*.

Woodsend, K. and Lapata, M. (2017). Text rewriting improves semantic role labeling (extended abstract). In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 5095–5099.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G. S., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144.

Xue, N. and Palmer, M. (2004). Calibrating features for semantic role labeling. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 88–94, Barcelona, Spain. Association for Computational Linguistics.

Xue, N. and Palmer, M. (2009). Adding semantic roles to the chinese treebank. *Natural Language Engineering*, 15:143–172.

Yarowsky, D., Ngai, G., and Wicentowski, R. (2001). Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the First International Conference on Human Language Technology Research*.

Zettlemoyer, L. S. and Collins, M. (2005). Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, UAI'05, page 658–666, Arlington, Virginia, USA. AUAI Press.

Zhang, S., Duh, K., and Van Durme, B. (2017). MT/IE: Cross-lingual open information extraction with neural sequence-to-sequence models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 64–70, Valencia, Spain. Association for Computational Linguistics.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020). Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Zhou, J. and Xu, W. (2015). End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1127–1137, Beijing, China. Association for Computational Linguistics.

Ziemski, M., Junczys-Dowmunt, M., and Pouliquen, B. (2016). The united nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Zoph, B. and Knight, K. (2016). Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 30–34, San Diego, California. Association for Computational Linguistics.