Cross-Lingual Transfer of Semantic Roles: From Raw Text to Semantic Roles

Maryam Aminian[1], Mohammad Sadegh Rasooli[2], Mona Diab[1,3]
[1]The George Washington University, Washington
[2]Facebook AI, Menlo Park, CA
[3]AWS, Amazon AI
{aminian,mtdiab}@gwu.edu, rasooli@fb.com

Abstract

We describe a transfer method based on annotation projection to develop a dependency-based semantic role labeling system for languages for which no supervised linguistic information other than parallel data is available. Unlike previous work that presumes the availability of supervised features such as lemmas, part-of-speech tags, and dependency parse trees, we only make use of word and character features. Our deep model considers using character-based representations as well as unsupervised stem embeddings to alleviate the need for supervised features. Our experiments outperform a state-of-the-art method that uses supervised lexico-syntactic features on 6 out of 7 languages in the Universal Proposition Bank.

1 Introduction

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

This paper considers the problem of annotation projection of *dependency-based* SRL in a scenario for which *only* parallel data is available for the target language. Recent state-of-the-art SRL systems have shown a significant reliance on the predicate lemma information while in a low-resource language, a lemmatizer might not be available. We first demonstrate that unsupervised stems can be used as an alternative to supervised lemma features. We further show that we can obtain a robust and simple SRL model for the target language without relying on *any* explicit linguistic feature (including lemmas), either supervised or unsupervised. We achieve this goal by changing the structure of a state-of-the-art deep SRL system (Marcheggiani et al., 2017) to make it independent of supervised features. Our model solely rely on word and character level features in the target language.

The main contribution of this work is on applying annotation projection without relying on supervised features in the target language of interest. To the best of our knowledge, this is the first study that builds a cross-lingual SRL transfer model in the absence of any explicit linguistic information in the target language. We make use of the recently released Universal Proposition Banks (Akbik et al., 2016)[1],

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

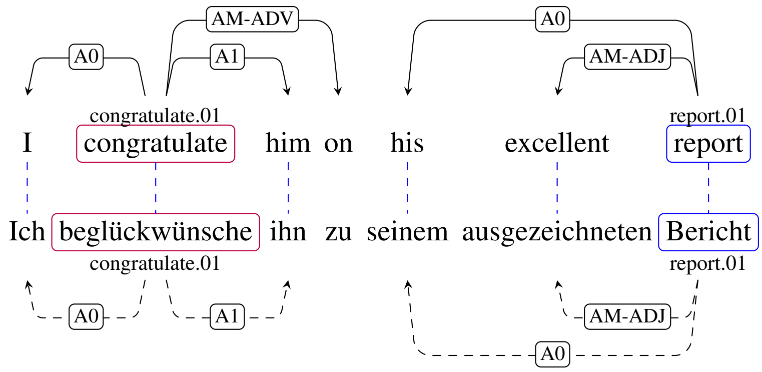

Figure 1: An example of annotation projection for an English-German sentence pair from the Europarl corpus (Koehn, 2005). Supervised predicate-argument structure of the English sentence (edges on top) is generated using our supervised SRL system trained on PropBank 3 (Palmer et al., 2005). Dashed lines in the middle show intersected word alignments from Giza++ (Och and Ney, 2003). Dashed edges at the bottom show the projected predicate-arguments.

a semi-automatically annotated data that unifies the annotation scheme for all languages. We show the effectiveness of our method on a range of languages, namely German, Spanish, Finnish, French, Italian, Portuguese, and Chinese. We compare our model to a state-of-the-art baseline that uses a rich set of supervised features and show that our model outperforms on six out of seven languages in the Universal Proposition Banks. Furthermore, for Finnish, a morphologically rich language, our model with unsupervised features improves over the model that relies on a supervised lemmatizer.

This paper is structured as the following: §2 briefly overviews the dependency-based SRL task and annotation projection, §3 describes our approach, §4 shows the experimental results and analysis, §5 gives overviews about the related work, and §6 concludes the paper and proposes suggestions for future work.

2 Background

In this section, we provide a brief overview of dependency-based SRL and annotation projection.

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

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

where r is the role of the jth word as an argument for the predicate word x_{p_i}. In case that a word is not an argument, r is `NULL`. Evaluation of the system output is conducted on semantic dependencies $(p_i \xrightarrow{r} j|\psi_i)$; thus the SRL system should find predicate senses as well as argument roles. During training, these dependencies are used as training instances for a machine learning algorithm. Previous work (Björkelund et al., 2009; Roth and Lapata, 2016; Marcheggiani et al., 2017) factorized this task into predicate sense disambiguation, argument identification, and argument classification.

Annotation Projection In annotation projection, we assume that we have a parallel data $\mathcal{P} = [(s^{(1)}, t^{(1)}), \cdots, (s^{(k)}, t^{(k)})]$ such that each sentence $s^{(i)}$ is a translation of sentence $t^{(i)}$. Here, we assume that $s^{(i)}$ belongs to a rich-resource language in which annotated resources are available. In contrast, $t^{(i)}$ belongs to a low-resource target language where annotated data and tools such as semantic roles, dependency trees, part-of-speech tags, word senses, and lemmas might not be available.

For every sentence $s^{(i)}$, we run a supervised SRL system to obtain its supervised argument structure $\mathbb{L}_{s^{(i)}}$. Assuming that $s^{(i)} = [s_1^{(i)}, \cdots, s_{l_i}^{(i)}]$ and $t^{(i)} = [t_1^{(i)}, \cdots, t_{l'_i}^{(i)}]$, we use an automatic word alignment system to obtain *one-to-one* word alignments. We define $0 \leq a_j^{(i)} \leq l_i$ as the index of the source word that is aligned to the jth word in the ith target sentence, where $a_j^{(i)} = 0$ indicates a missing alignment. We use the following conditions to project a semantic dependency from a source sentence to a target sentence:

$$(a_p^{(i)} \xrightarrow{r} a_m^{(i)}|y) \in \mathbb{L}_{s^{(i)}} \Rightarrow \text{add } (p \xrightarrow{r} m|y) \text{ to } \mathbb{L}_{t^{(i)}}$$

where $\mathbb{L}_{s^{(i)}}$ is the supervised argument structure and $\mathbb{L}_{t^{(i)}}$ is the projected argument structure for the ith sentence. We assume that there is a universal predicate sense that is common across languages (this is the case in the Universal Propositon Banks). Figure 1 shows an example for an English-German translation pair. We use the projected data as training data in a supervised learning system to train a SRL system in the target language. In practice, many words do not receive any projected label mainly due to missing alignments. Thus, $\mathbb{L}_{t^{(i)}}$ usually contains sentences with partially projected semantic dependencies.

3 Our Model

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

3.1 Joint Argument Identification and Classification

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

Input Representation For each encoder $[\mathbb{E}_j]_{j=1}^m$, we represent each token s_i as the concatenation of its word embedding (x_i^{re} and x_i^{pe}), character embedding (x_i^{char}) and predicate lemma embedding ($x_{i,j}^{lem}$):[2]

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

where:

- $x_i^{re} \in \mathbb{R}^{d_w}$ is a randomly initialized word embedding vector;
- $x_i^{pe} \in \mathbb{R}^{d_w}$ is an external pre-trained word embedding that is fixed during training;
- $x_i^{char} \in \mathbb{R}^{d_{ch}}$ is character representation of each token s_i. For every token, we obtain x_i^{char} by running a deep bidirectional LSTM (Hochreiter and Schmidhuber, 1997) on top of each word. We use the concatenation of the final backward representation of the first character, and final forward representation of the last character to represent each token:

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

[2]We use [;] notation to show vector concatenation.

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

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

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

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

  – Represent each lemma by a deep character BiLSTM. This BiLSTM is different from the character BiLSTM in x^{char}.

  – Use an unsupervised morphological analyzer to give the surface-form stem of each word. This way, we can use a lemma embedding dictionary without requiring a lemmatizer.

Predicate-Specific Encoder A deep BiLSTM is used to get the final representation for each token in a sentence. In the following notation, $h_{i,j}$ is the final hidden state from the deep BiLSTM model for the ith token with respect to the jth predicate:

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

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

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

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

$$W_{j,r} = \texttt{RELU}(U[u_j^l, v_r])$$

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

A graphical depiction of the network in a case for which lemmas are represented by character BiLSTMs is shown in figure 2. As shown in the figure, we use two different character BiLSTMs in order to represent lemmas: one for the input representation and the other for the decoder representation.

4 Experiments

Datasets and Tools We use English as the source language and project SRL annotations to the following languages: German, Spanish, Finnish, French, Italian, Portuguese, and Chinese. We use the Europarl parallel corpus (Koehn, 2005) for the European languages and a random sample of 2 million sentence pairs from the MultiUN corpus (Eisele and Chen, 2010) for Chinese. We use the Giza++ tool (Och and Ney, 2003) with its default setting for word alignment. We run Giza++ in source-to-target and the reverse

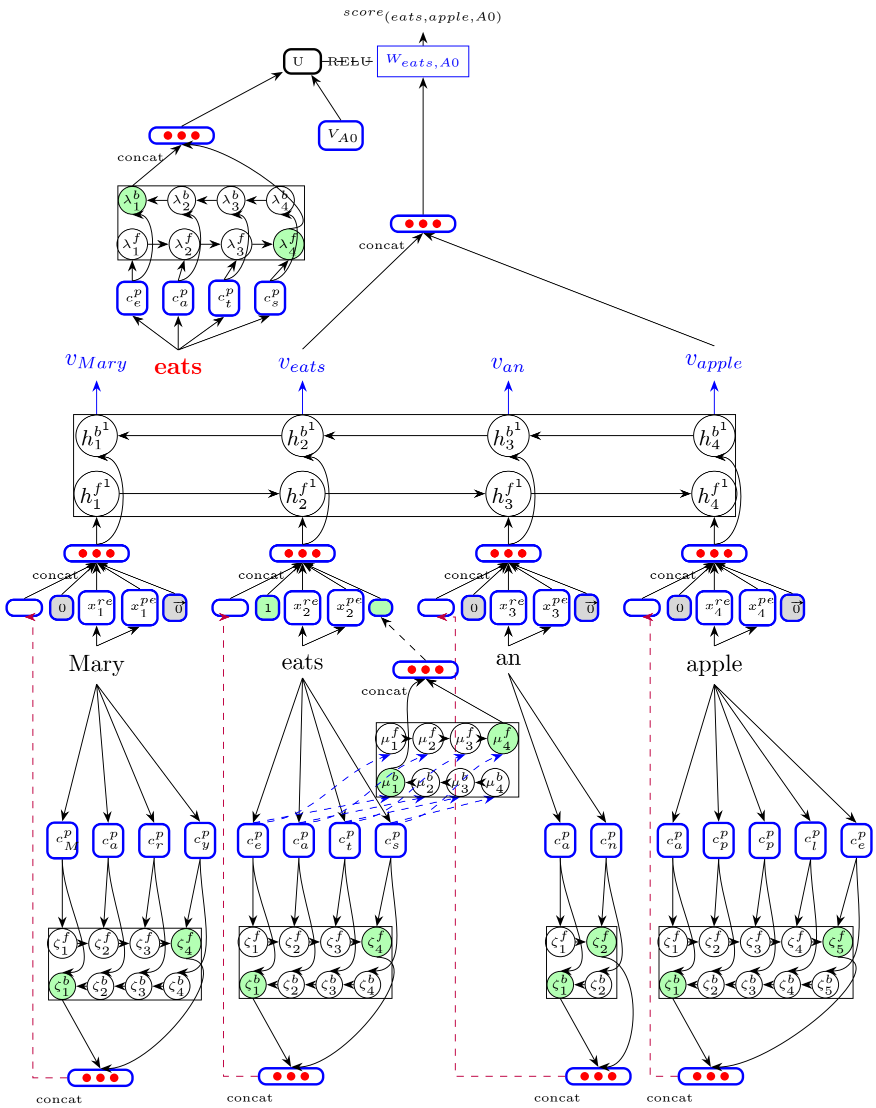

Figure 2: A graphical depiction of our joint argument identification and classification model without using part-of-speech tags, lemmas, and syntax. In this example, the predicate-specific encoder considers word *eats* as the sentence predicate and the goal is to score the assignment of argument *apple* with label A_0. Our model contains three *different* character BiLSTMs; at the bottom, a character BiLSTM is run to acquire a character-based representation for all the words in the sentence in the absence of POS tags. There are two character BiLSTMs for predicate lemma: one in the encoder level (next to the second word) to model predicate lemma in the input layer and the other in the decoder level (top left). In this example, we just show one layer of BiLSTM but we use a deep BiLSTM in our experiments.

direction and get the intersection of alignment links. For English, we use the pre-trained embedding vectors generated using the structured skip-gram model of Ling et al. (2015). For the target languages, we train Word2vec (Mikolov et al., 2013) on Wikipedia data to generate embedding vectors.

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

Predicate Disambiguation Our model is agnostic to predicate senses but since our automatic evaluation relies on automatic predicate senses, we need a disambiguation module. Predicate disambiguation systems typically contains separate classifiers for each predicate lemma (Björkelund et al., 2009). Since we do not have a reliable lemmatizer in the target language, we train a single classifier for all predicates. We encode a sentence with a three-layer deep BiLSTM and run a softmax layer on top of each predicate to disambiguate the predicate sense of each predicate.

Predicate identification on the source side For projection experiments, first of all we need to identify predicates in the source language. Input to our predicate identifier is the concatenation of word embed-

Lang.	#Sent.	#Tokens	#Types	#Pred.
de	332K	6M	90K	867K
es	903K	25M	120K	3M
fi	558K	8M	243K	1M
fr	924K	26M	93K	3M
it	617K	17M	88K	2M
pt	632K	17M	98K	2M
zh	821K	21M	183K	1M

Table 1: Sizes of the projected data.

ding, pre-trained fixed word embedding, POS embedding[3], and character representation (obtained from a character BiLSTM) for every token in the sentence. We use a deep BiLSTM to get the final representation for each token. The ultimate predictions are made by performing an affine transform on the BiLSTM hidden output.

4.1 Projection Experiments

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

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

$$\texttt{density}^{(i)} = \frac{\sum_{j=1}^{l'_j} \mathbb{I}(a_j^{(i)} > 0)}{l'_i}$$

where l'_i is the length of the ith target sentence in parallel data, $a_j^{(i)}$ is the alignment index for the jth word in the target sentence, and $\mathbb{I}(a_j^{(i)} > 0)$ is an indicator for a non-$\texttt{NULL}$ alignment. We prune the target sentence pairs with density less than 80% for all European languages. We set this threshold to 60% for Chinese in order to obtain a comparable number of sentences to the European languages. Table 1 summarizes the sizes of projected datasets after applying the density filter. We set the number of training epochs to 2 for all languages based on development results obtained from the English to German projections.

Since the original model of Marcheggiani et al. (2017) heavily relies on the predicate lemma information for making robust prediction, we further assess the influence of using explicit linguistic features in our model by using a) supervised lemma from the UDPipe pre-trained models (Straka and Straková, 2017), and b) unsupervised stems obtained from unsupervised morphological analyzer. We use the unsupervised morphological analyzer of Virpioja et al. (2013), and obtain morpheme classes by running Morfessor FlatCat (Grönroos et al., 2014) on the output of the analyzer. We run the *fixed-affix* finite-state machine of (Rasooli et al., 2014) to obtain a single stem for all words including the out-of-vocabularies.

Results We compare our character-based approach (*CModel*) with three different models: 1) The cross-lingual model of Aminian et al. (2017) (*Bootstrap*) that uses a rich set of supervised features including supervised lemmas, POS tags, and dependency parse information, 2) a variant of our model that uses supervised lemmas (*SLem*) generated by a lemmatizer to represent predicate lemmas in the input and the decode layers, and 3) a model similar to the second model but using unsupervised stems (*UStem*) generated by an unsupervised morphological analyzer to represent predicate lemmas. Here, we aim to asses

[3] Since this is only used for a supervised setting, we are able to use POS features.

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

Table 2: Results of projection experiments using our character based model (*CModel*) on the Universal PropBank test sets compared to different baselines: the SRL system of *Aminian et al. (2017)* (*Bootstrap*), *SLem* that shows the results of our model when supervised lemma is used and *UStem* that show the results of our model with unsupervised stem. Numbers in parenthesis show results with automatic predicate senses.

the effects of using different levels of explicit linguistic features ranging from fully specified supervised features to unsupervised features in our model. The *Bootstrap* model uses an iterative bootstrapping approach by utilizing a special cost function and benefiting from a rich set of supervised lexico-syntactic features, thereby, it is considered a hard baseline. Since *Bootstrap* has a large number of features, the model is not memory-wise scalable to our projection data sizes. Therefore we train the *Bootstrap* model on a random sample of 20K sentences. This number is similar to the number of sentences used in the original experiments (Aminian et al., 2017).

Table 2 shows labeled F-scores using both gold and automatic predicate senses on the test portion of the Universal Proposition Banks. The last raw in the table shows results from the supervised SRL systems trained on the training portion of the Universal Proposition Banks for each language, thereby can servr as an upper bound for our model. As shown in Table 2, our model (*CModel*) outperforms the *Bootstrap* model for all languages except French. Additionally, our model performs on par to the supervised lemma and unsupervised stem models. This demonstrates the power of our approach even though our model has access to fewer linguistic features in the target language. Using unsupervised stems outperforms supervised lemma on all languages except Portuguese and Italian. This further highlights the reliance of the model on the accuracy of lemmatizer.

Analysis As shown in Table 2, using automatic predicate senses leads to a significant reduction in accuracy. This degradation is caused by two reasons. First, training a single classifier for all predicates in the absence of explicit predicate lemma information, and second, using unified predicate senses for all languages leads to lower precision for out-of-vocabulary words. This happens due to the fact that we cannot make use of the default sense of predicate (`lemma.01`). Among all the languages in our experiments, French is the only language that our model underperforms the *Bootstrap* model. Our analysis on French shows that our model has not been able to correctly predict A0 and A1 arguments in 20% and 30% of cases, and labeled them as `NULL`.

5 Related Work

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

One obstacle for developing transfer models is the absence of a unified annotation scheme for all languages. There has been a great deal of work in developing universal annotation schemes for a variety of tasks such as POS tagging (Petrov et al., 2011), dependency parsing (Nivre et al., 2017), morphology (Kirov et al., 2018), and SRL (Kozhevnikov and Titov, 2013a; Wang et al., 2017). Our work makes use of the recently released Universal Proposition Bank (Akbik et al., 2016). This dataset maps every predicate lemma in every language to its corresponding English lemma following the frame and role label schemes of the English Proposition Bank 3.0 (Palmer et al., 2005)

In the realm of *supervised* SRL methods, however, there have been several efforts to build SRL models that do not need a wide range of linguistic features (specifically syntactic features) (Marcheggiani et al., 2017; Zhou and Xu, 2015; He et al., 2017, 2018; Cai et al., 2018; Mulcaire et al., 2018). In a more recent study, Mulcaire et al. (2018) proposed a polyglot SRL system that benefits from the similarities between the semantic structures of different languages to improve monolingual SRL. All those studies, however, assume the availability of semantically annotated datasets for the target language, thus making them non-applicable to low-resource languages.

6 Conclusion

We have described a method for cross-lingual transfer of dependency-based SRL systems via annotation projection. Our model is agnostic to linguistic features leading to a robust model that can be trained on projected text on a target language without annotated data. We have shown that our model achieves comparable performance in annotation projection and also supervised SRL. In addition to improving the performance of our model with the current setting, future work should study more effective ways to apply the transfer methods; e.g. combining with the direct transfer method in the absence of large parallel corpora.

Acknowledgments

The first and third authors have been partly funded by the DARPA LORELEI grant and generous support by Leidos Corp.. We would like to acknowledge the useful comments by three anonymous reviewers who helped in making this publication more concise and better presented.

References

Akbik, A., l. chiticariu, M. Danilevsky, Y. Li, S. Vaithyanathan, and H. Zhu (2015). Generating high quality proposition banks for multilingual semantic role labeling. 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)*, pp. 397–407. Association for Computational Linguistics.

Akbik, A., v. kumar, and Y. Li (2016). Towards semi-automatic generation of proposition banks for low-resource languages. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 993–998. Association for Computational Linguistics.

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

Björkelund, A., L. Hafdell, and P. Nugues (2009). *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, Chapter Multilingual Semantic Role Labeling, pp. 43–48. Association for Computational Linguistics.

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

Collins, M. (2002, July). Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pp. 1–8. Association for Computational Linguistics.

Eisele, A. and Y. Chen (2010, may). Multiun: A multilingual corpus from united nation documents. In N. C. C. Chair), K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias (Eds.), *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Erk, K., A. Kowalski, S. Padó, and M. Pinkal (2003, July). Towards a resource for lexical semantics: A large german corpus with extensive semantic annotation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, pp. 537–544. Association for Computational Linguistics.

Grönroos, S.-A., S. Virpioja, P. Smit, and M. Kurimo (2014, August). Morfessor flatcat: An hmm-based method for unsupervised and semi-supervised learning of morphology. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, Dublin, Ireland, pp. 1177–1185. Dublin City University and Association for Computational Linguistics.

He, L., K. Lee, M. Lewis, and L. Zettlemoyer (2017). Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 473–483. Association for Computational Linguistics.

He, S., Z. Li, H. Zhao, and H. Bai (2018). Syntax for semantic role labeling, to be, or not to be. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2061–2071. Association for Computational Linguistics.

Hochreiter, S. and J. Schmidhuber (1997). Long short-term memory. *Neural computation 9*(8), 1735–1780.

Hwa, R., P. Resnik, A. Weinberg, and O. Kolak (2002). Evaluating translational correspondence using annotation projection. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 392–399. Association for Computational Linguistics.

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

Kirov, C., R. Cotterell, J. Sylak-Glassman, G. Walther, E. Vylomova, P. Xia, M. Faruqui, S. Mielke, A. D. McCarthy, S. Kübler, et al. (2018). Unimorph 2.0: Universal morphology. *arXiv preprint arXiv:1810.11101*.

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

Kozhevnikov, M. and I. Titov (2013a). Bootstrapping semantic role labelers from parallel data. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pp. 317–327. Association for Computational Linguistics.

Kozhevnikov, M. and I. Titov (2013b). 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)*, pp. 1190–1200. Association for Computational Linguistics.

Ling, W., C. Dyer, A. W. Black, and I. Trancoso (2015, May–June). Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, Colorado, pp. 1299–1304. Association for Computational Linguistics.

Marcheggiani, D., A. Frolov, and I. Titov (2017, August). A simple and accurate syntax-agnostic neural model for dependency-based semantic role labeling. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, Vancouver, Canada, pp. 411–420. Association for Computational Linguistics.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119.

Mulcaire, P., S. Swayamdipta, and N. A. Smith (2018). Polyglot semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 667–672. Association for Computational Linguistics.

Nair, V. and G. E. Hinton (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814.

Neubig, G., C. Dyer, Y. Goldberg, A. Matthews, W. Ammar, A. Anastasopoulos, M. Ballesteros, D. Chiang, D. Clothiaux, T. Cohn, et al. (2017). Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

Nivre, J., Ž. Agić, L. Ahrenberg, M. J. Aranzabe, M. Asahara, et al. (2017). Universal Dependencies 2. LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague.

Och, F. J. and H. Ney (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics 29*(1), 19–51.

Padó, S. and M. Lapata (2005). Cross-linguistic projection of role-semantic information. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.

Padó, S. and M. Lapata (2009). Cross-lingual annotation projection for semantic roles. *Journal of Artificial Intelligence Research 36*(1), 307–340.

Palmer, M., D. Gildea, and P. Kingsbury (2005). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics, Volume 31, Number 1, March 2005*.

Petrov, S., D. Das, and R. McDonald (2011). A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086*.

Rasooli, M. S., T. Lippincott, N. Habash, and O. Rambow (2014, June). Unsupervised morphology-based vocabulary expansion. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Baltimore, Maryland, pp. 1349–1359. Association for Computational Linguistics.

Roth, M. and M. Lapata (2016). Neural semantic role labeling with dependency path embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1192–1202. Association for Computational Linguistics.

Straka, M. and J. Straková (2017, August). Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Vancouver, Canada, pp. 88–99. Association for Computational Linguistics.

van der Plas, L., M. Apidianaki, and C. Chen (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*, pp. 1279–1290. Dublin City University and Association for Computational Linguistics.

van der Plas, L., P. Merlo, and J. Henderson (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*, pp. 299–304. Association for Computational Linguistics.

Virpioja, S., P. Smit, S.-A. Grönroos, M. Kurimo, et al. (2013). Morfessor 2.0: Python implementation and extensions for morfessor baseline. Technical report, Aalto University.

Wang, C., A. Akbik, l. chiticariu, Y. Li, F. Xia, and A. Xu (2017, September). Crowd-in-the-loop: A hybrid approach for annotating semantic roles. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, pp. 1913–1922. Association for Computational Linguistics.

Zaghouani, W., M. Diab, A. Mansouri, S. Pradhan, and M. Palmer (2010, July). The revised arabic propbank. In *Proceedings of the Fourth Linguistic Annotation Workshop*, Uppsala, Sweden, pp. 222–226. Association for Computational Linguistics.

Zhou, J. and W. Xu (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)*, pp. 1127–1137. Association for Computational Linguistics.