Cross-Lingual SRL Based upon Universal Dependencies

Ondřej Pražák and **Miloslav Konopík**
NTIS – New Technologies for the Information Society,
Department of Computer Science and Engineering,
Faculty of Applied Sciences, University of West Bohemia, Technická 8, 306 14 Plzeň
Czech Republic
ondfa@ntis.zcu.cz
konopik@kiv.zcu.cz

Abstract

In this paper, we introduce a cross-lingual Semantic Role Labeling (SRL) system with language independent features based upon Universal Dependencies. We propose two methods to convert SRL annotations from monolingual dependency trees into universal dependency trees. Our SRL system is based upon cross-lingual features derived from universal dependency trees and supervised learning that utilizes a maximum entropy classifier. We design experiments to verify whether the Universal Dependencies are suitable for the cross-lingual SRL. The results are very promising and they open new interesting research paths for the future.

1 Introduction

The Semantic Role Labeling (SRL) task (Gildea and Jurafsky, 2002) belongs among shallow semantic parsing techniques. The SRL goal is to identify and categorise semantic relationships or *semantic roles* of given *predicates*. Verbs, such as "believe" or "cook", are natural predicates but certain nouns are accepted as predicates as well (see the third example in Figure 1). The simplified definition of semantic roles is that semantic roles are abstractions of predicate arguments. For example, the semantic roles for "believe" can be *Agent* (a believer) and *Theme* (a statement) and for "cook" *Agent* (a chef), *Patient* (a food), *Instrument* (a device for cooking) – see examples in Figure 1. The theory of predicates and their roles is very well established several linguistic resourcesin such as PropBank (Palmer et al., 2005), NomBank (for nouns) (Meyers et al., 2004) , VerbNet (Kipper et al., 2006) or FrameNet (Baker et al., 1998).

(1) [He]$_{\text{AGENT|A0}}$ <u>believes</u> [in what he plays] $_{\text{THEME|A1}}$.

(2) Can [you] $_{\text{AGENT|A0}}$ <u>cook</u> [the dinner] $_{\text{PATIENT|A1}}$?

(3) [The nation's] $_{\text{AGENT|AM-LOC}}$ largest [pension]$_{\text{THEME|A1}}$ <u>fund</u>,

Figure 1: Three examples of shallow semantic annotations: 1) and 2) are examples of verb predicates (labels are from VerbNet v3.2|SRL) and 3) of a noun predicate (labels NomBank|SRL).

In the SRL task, the semantic roles are considered at a higher abstraction level. E.g. for English, there are several core roles denoted by *A0* (usually Agent), *A1* (usually Patient) and *A2*, modifier arguments (*AM-**), restriction arguments (*R-**) and others.

Cross-lingual SRL shares the same goal with monolingual SRL, however, it attempts to provide semantic annotations for languages that lack for training data. The training data are taken from a source language (usually English) and they are transferred to a target language or a language independent model is trained. Cross-lingual SRL contains added value when compared to the monolingual SRL. In cross-lingual SRL, it is ensured that the tagset and the annotation procedure is coherent for all supported languages because the annotation is transferred from one common source (one language) to other languages. This does not apply to the monolingual scenario where we often deal with incompatible tagsets and different annotation guidelines.

In our paper, Universal Dependencies (UD) (Nivre et al., 2016) are the primary vehicle to enable transferring the learned rules from one language to another. UD annotations successfully describe syntax of a high number of languages, however, their annotation structure is somewhat simplified when compared to specific dependencies

Proceedings of Recent Advances in Natural Language Processing, pages 592–600,
Varna, Bulgaria, Sep 4–6 2017.
https://doi.org/10.26615/978-954-452-049-6_077

(SD)[1]. In this paper, we attempt to find out to what extent the UD annotations are suitable for cross-lingual SRL task.

2 Related Work

Approaches to SRL on languages that lack proper training data can be divided into three main categories: **1)** *Annotation projection* methods attempt to transfer the annotation from one language to the other one and then train a SRL system on the transferred annotation. **2)** *Model transfer* approaches are designed to use language-independent features to train a universal model that can be applied to languages that support the designed features. **3)** Methods based upon *unsupervised training* require no annotated data, however, the models have difficulties to assign meaningful labels to predicate arguments.

Annotation projection Padó and Lapata (2009) transfer annotation to a target language via word alignments obtained from parallel corpora. Annesi and Basili (2010) use a similar approach and extend it with an HMM model to increase the transfer accuracy.

Model transfer Kozhevnikov and Titov (2013) use cross-lingual word mappings and cross-lingual semantic clusters obtained from parallel corpora, and cross-lingual features extracted from unlabelled syntactic dependencies to create a cross-lingual SRL system. In (Kozhevnikov and Titov, 2014), they try to automatically find a mapping between language specific models using parallel data.

Unsupervised SRL Grenager and Manning (2006) deploy unsupervised learning using the EM algorithm based upon a structured probabilistic model of the domain. Lang and Lapata (2011) discover arguments of verb predicates with high accuracy using a small set of rules. A split-merge clustering is consequently applied to assign (nameless) roles to the discovered arguments. Titov and Klementiev (2012) propose a superior argument clustering by using the Chinese restaurant process.

Our approach belongs among the model transfer approaches. We introduce a first attempt to use Universal Dependencies as cross-language features in SRL. Most of the state-of-the-art approaches to SRL rely on lexical features (e.g. word lemmas). In the cross-language scenario, such features require bilingual models (e.g. word mapping via machine translation or bilingual clusters). In this paper, we attempt to create a multi-language model that can be potentially trained on many languages and therefore we omit all bilingual features including the lexical features.

3 Universal Dependencies

The main topic of our paper is to utilize the power of Universal Dependencies to analyse syntax in a cross-lingual manner. The UD annotation scheme evolved as a compilation of three universal annotation principles: universal dependency relations from Stanford (de Marneffe and Manning, 2008), part-of-speech tags from Google (Petrov et al., 2012), and morphological features (Zeman, 2008) from UFAL, Charles University.

The design of UD annotations is in principle similar to SD annotations but it differs in some crucial aspects. From the SRL point of view, the most critical difference is making content words the heads. In SD annotation for many languages, the auxiliary verbs, prepositions and conjunctions are often the heads.

Although the specification is quite new, there are several frameworks that enable end-to-end parsing into UDs: UDpipe (Straka et al., 2016), Stanford CoreNLP (Manning et al., 2014), Malt parser (Nivre and Hall, 2005), and others.

4 Annotation Conversion

4.1 Datasets

In our experiment, we use the CoNLL 2009 datasets (Hajič et al., 2009) for the following languages: Czech (CZ), English (EN), German (DE), Spanish (ES) and Chinese (ZH). Only English and Chinese datasets use compatible[2] tagsets of SRL roles. All other languages use different tagsets.

Next, we use a freely available script from Mikhail Kozhevnikov[3] to convert the Czech-English Dependency Treebank (PCEDT) (Hajič et al., 2012) into SRL annotations. The resulting annotations use the same labels for both languages.

[1] SD stands for task-Specific or language-Specific Dependencies, e.g. the “old” Stanford dependencies.

[2] The datasets can be mapped to each other by few simple rules.

[3] http://www.ml4nlp.de/code-and-data/treex2conll

The CoNLL data are divided into training and evaluation parts. We use the CoNLL 2009 Czech evaluation part also for the PCEDT dataset since it uses the same tagset.

4.2 Existing SD Annotations

SRL copora defined in Section 4.1 use SD trees as the basis of SRL annotations. The annotations label tree heads as predicate arguments (see Figure 2a). The meaning of such an annotation is that the predicate arguments are composed of all words that belong under heads including the head words. Figure 2b illustrates that the *A1* argument is composed of the phrase "in what he plays".

4.3 Conversion Methods

We need SRL annotations built upon UD trees for our experiments. Therefore, we propose two methods to obtain such annotations by converting the annotations built on SD trees into the annotations build on UD trees. Given the UD tree, we label such tree heads so that the predicate arguments cover the same words as in the SD trees – see Figure 2c. In this example, we label the word "what" as the *A1* argument in the UD tree instead of the word "in" which was annotated in the SD tree.

In the first proposed method, we look for such annotations of heads in the given UD trees so that the word coverage of arguments is as close to the original SD trees as possible. For each predicate argument, we look for the optimal head that covers as many of the original words as possible but at the same time as few extra words as possible. In this method, we annotate only one head for each predicate argument. The same constraint is present in some original SRL annotations where only one head per predicate argument is allowed. To formalize the process, we look for an optimal head obtained by the following formula:

$$N^{\star}_{UD} = \operatorname*{arg\,max}_{N_{UD} \in T_{UD}} |\,\texttt{C}(N_{UD}) \cap \texttt{C}(N_{SD})| - \\ |\,\texttt{C}(N_{UD}) \setminus \texttt{C}(N_{SD})| : \forall N_{SD} \in \texttt{arg}(T_{SD}),$$

where N_{SD} and N_{UD} are nodes of an SD tree T_{SD} and an UD tree T_{UD}, respectively; $\texttt{C}$ is coverage operator which returns all nodes that belong under the specified node including the node itself; $\texttt{arg}$ returns all predicate arguments of a given tree.

In the second method, we abandon the constraint of one head per predicate argument (the same holds true e.g. for the original Czech SRL annotations in CoNLL 2009 (Hajič et al., 2009)). We proceed similarly to the first method and we try to optimize the word coverage of the new argument labels in the UD annotations. However, in this approach, we label as many heads as necessary while ensuring that none of the heads is in a subtree of another one. We start by annotating all words that are located under a head of an argument in the original SD tree. Then we iteratively look for common heads (parents) of all the words in the corresponding UD tree. The algorithm stops, when no more common heads are found. We keep the annotation of the heads and discard the annotation of all dependent words (children of the heads). In this way, the SRL labels may be assigned to more than one head for some predicate arguments. The optimization formula is as follows:

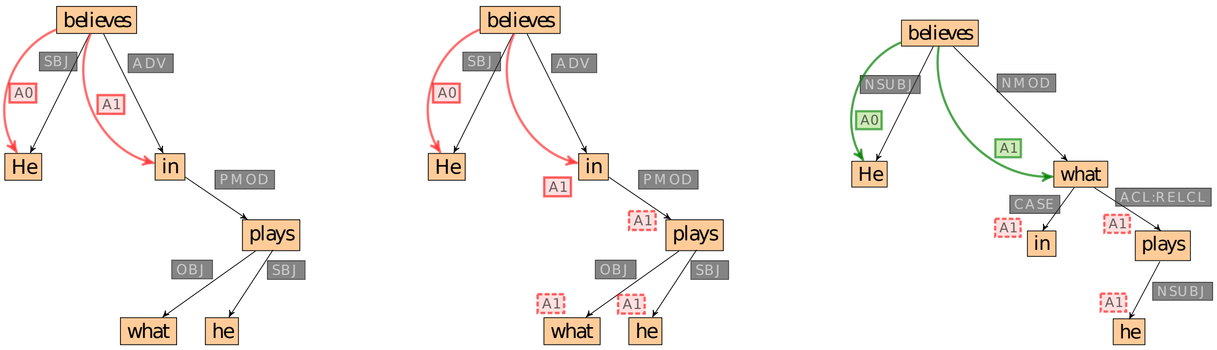
$$\{N^{\star}_{UD}\} = \operatorname*{arg\,max}_{\{N_{UD}\} \in T_{UD}} |\,\texttt{C}(\{N_{UD}\}) \cap \texttt{C}(N_{SD})| - \\ |\,\texttt{C}(\{N_{UD}\}) \setminus \texttt{C}(N_{SD})| : \forall N_{SD} \in \texttt{arg}(T_{SD}),$$

where $\{N_{UD}\}$ is a set from T_{UD} and $\{N_{UD}\} \in T_{UD}$ returns all possible sets of distinct subtrees from T_{UD}.

Figure 3b shows different outcomes of the first and the second conversion methods. The first method is constrained to label only one head per predicate. In this case, the optimal head that covers most of the words is "number". The second method is free to label as many heads as necessary and thus it labels both "plants" and "number" heads. This can be confusing for downstream applications, however, in some cases it is the only way to correctly label all argument words.

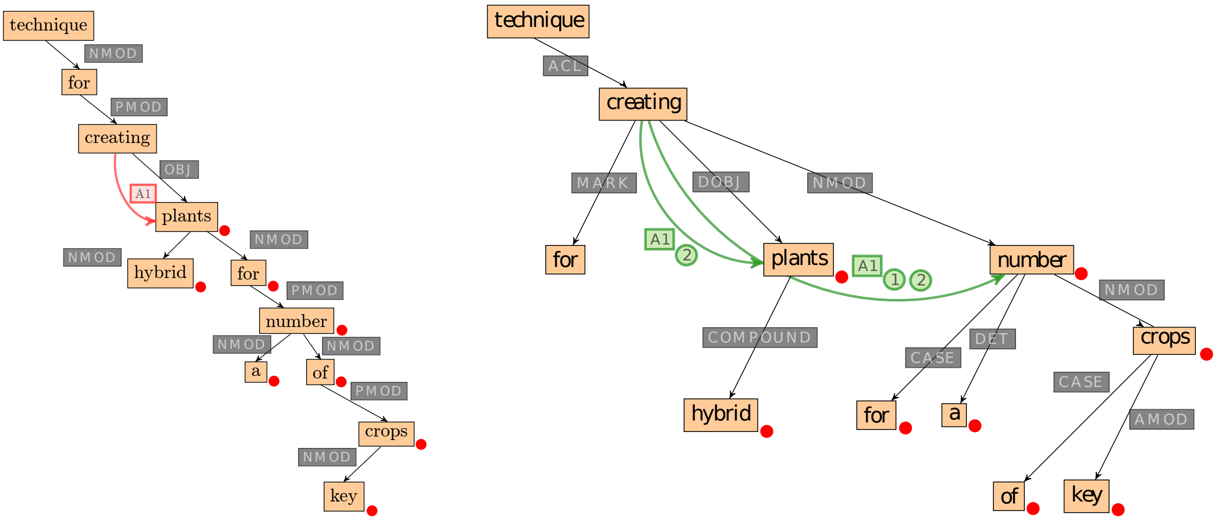
4.4 Obtaining the UD Trees

So far, we have dealt with the annotation conversion to UD trees but we have not mentioned how to obtain the UD trees in the first place. We have considered two options: **1)** to convert the SD trees to UD trees and **2)** to use syntactic parsers capable of producing UD trees. The first option looks more attractive at first glance since it would produce trees almost without errors. Indeed, the existing UD treebanks were created mostly by conversion from other treebanks. However, it is almost impossible to put this approach into practice. The conversion is not trivial and the result needs to be corrected by hand. The scripts are not publicly available and moreover they are designed to work with specific types of the syntactic annotations. Since we perform experiments on



(a) (b) (c)

Figure 2: Example of SRL annotations: (a) shows semantic role annotation for an SD tree, (b) shows the coverage of role annotations and (c) shows the result of the conversion to the UD tree. The annotation in sub-figures (a) and (b) are real examples from the Conll 2008 dataset (sentence no. 57 from train.closed.conll08 – all examples in our paper are based upon the CoNLL 2008 and 2009 training data) and sub-figure (c) shows the output of our conversion algorithm.

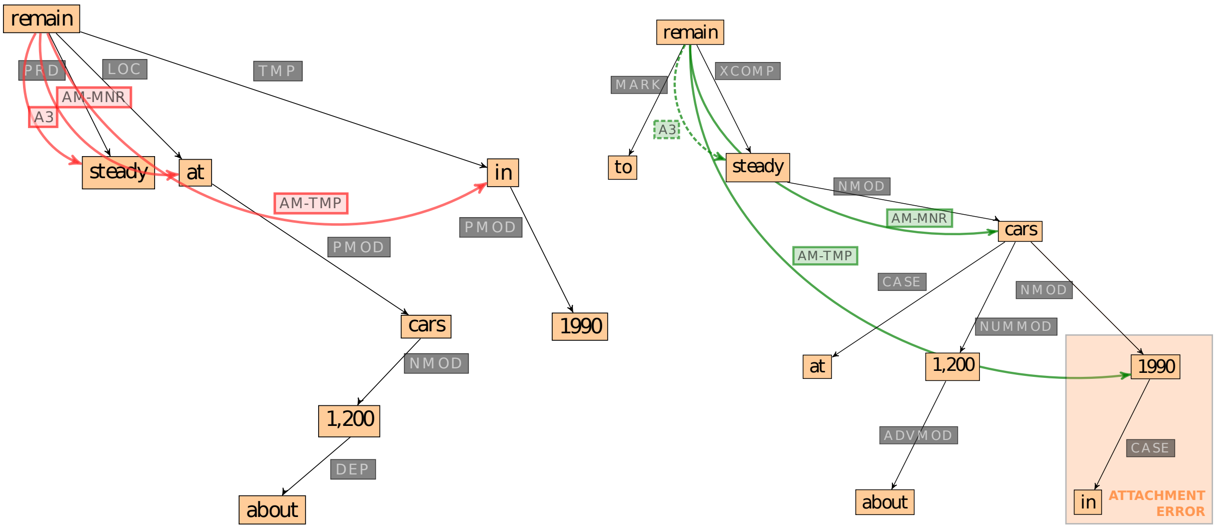


(a) (b)

Figure 3: An illustration of a difference of the first and the second conversion method. The dots indicate the coverage of the argument *A1*: (a) is the orginal annotation (sentence no. 117 from train.closed.conll08) and (b) shows the result of the conversion to an UD tree – ① marks the annotation for the first method and ② marks the second method.

five languages, we would have to create or obtain five scripts. These problems lead us to use the second option. We create the UD trees by parsing the sentences in datasets with an UD syntactic parser. This produces trees with errors and complicates the SRL conversion. However, this approach is available for all languages that are supported by an UD parser instantly. In our experiments, we use UDPipe (Straka et al., 2016) for Czech, English, German and Spanish, and Stanford CoreNLP (Manning et al., 2014) for English and Chinese for the cross-lingual experiment (EN-ZH) because UDPipe does not provide a model for Chinese and Stanford CoreNLP does not provide a model for Czech. We use models provided with parsers based on UD v1.2.

The conversion computed on an UD tree with parsing errors is shown in Figure 4. In the UD tree (Figure 4b), the “1990” node was incorrectly attached to “cars” node instead of the correct node “remain”. The results of the SRL annotation conversion are the same for both conversion methods in this case. Both methods labelled the “cars” node with the *AM-MNR* argument because any other annotation would induce a bigger error in the coverage of the argument. The *A3* argument was



(a)

(b)

Figure 4: shows the SRL annotation conversion from (a) the SD tree (sentence no. 3 from train.closed.conll08), (b) to an UD tree that contains parsing errors.

discarded completely because labelling any node would cause a bigger error than not labelling any node.

5 Experiments

As in all approaches to SRL, we perform the task in two steps: **1)** *argument identification* and **2)** *argument cassification*.

Argument Identification The goal of the argument identification step is to identify (find) the tree nodes that are the heads of argument phrases. We classify each node into two classes: *argument / not argument*.

Argument Classification The task of the argument classification step is to assign correct labels to the nodes identified as arguments in the previous step. We classify each node identified as an argument into classes that correspond to all possible role labels. We use exactly the same classifiers and features for both steps.

5.1 Evaluation Metrics

For comparison with the state-of-the-art techniques, we compute labelled and unlabelled F1 measure with CoNLL 2009 official evaluation script (marked *us* for unlabelled, *ls* for labelled F1 measures). However, these metrics are computed on heads only which is not suitable for experiments based upon UD trees because the heads of UD trees and SD trees differ significantly. Therefore, we compute F1 score of precision and recall for argument identification (u) and accuracy (l) for role labelling. Both metrics are evaluated on the whole subtrees. This means that we compute the evaluation score for all words from subtrees, not just for the heads. The labelling accuracy is computed only for correctly identified argument nodes (the same approach is taken in (Kozhevnikov and Titov, 2014)).

The accuracy of argument labelling can be computed only for datasets that use the same labels for arguments. For the cross-lingual experiment, the tagsets of some language pairs differ. In such cases, we compute the F1 score of collocation and purity as defined in (Lang and Lapata, 2010) and denote it F_1^c as in (Kozhevnikov and Titov, 2013).

5.2 Classifier & Features

We train a supervised system based upon the Maximum Entropy classifier using the Brainy tool (Konkol, 2014). We use separate models for verb and non-verb predicates.

All features employed in our system are syntactic:

- *Predicate-argument distance* – distance between locations of a predicate and an argument in a sentence.
- *POS* – part-of-speech of the predicate, the argument and their parent nodes.
- *Dependency relation* – dependency tree relation of the predicate, the argument and their

parent nodes.

- *Directed path* – dependency tree path from the predicate to the argument including the indication of dependency directions.
- *Undirected path* – list of relations from the predicate to the argument.
- *Verb voice* – indication of active/passive voice.
- *Other syntactic features* – `feats` column in CoNLL 2009 format with additional information about the words.
- *Bigram features* – predicate-argument bigrams of the part-of-speech and the dependency relations.

The dependency path features are encoded as a probability of a word being a predicate argument (or having a specific role) given the path. These features are more general and the resulting vectors have smaller dimension. Also, the cost function is smoother and thus the model is easier to train.

5.3 Experiment Descriptions

Evaluation of the conversion methods In the first experiment, we evaluate both the conversion methods in order to find out their abilities to transfer the SRL annotations from SD trees to UD trees. We measure the coverage of arguments in the converted annotations and compare it to the original annotations. This experiment should give us a theoretical upper boundary for the experiments based upon UD trees.

Monolingual experiments on SD trees Next, we perform a monolingual SRL experiment on gold SD trees. The experiment uses the same data and conditions as there were at the CoNLL 2009 conference. The results should evaluate the deployed classifier and features in relation to existing monolingual SRL systems. We use the official evaluation metrics as well as our own metric (see Section 5.1) in order to see the relation between the metrics.

Monolingual experiments on UD trees We provide an additional monolingual experiment on converted data in the UD trees. Models for all five languages are trained on the converted data and tested on evaluation parts of the datasets. The test should reveal how well can the features based upon Universal Dependencies support the model decisions about argument identification and labelling.

Cross-lingual experiments Cross lingual experiments are the most important experiments described in this paper. They show the ability of the proposed method based upon Universal Dependencies to transfer a model learned on one language (English) to other languages Czech (CZ), German (DE), Spanish (ES) and Chinese (ZH).

6 Results

We state the results separately for the verb predicates (*V* in the tables), non-verb (other) predicates (*O*) and all predicates (*A*). When there are no non-verb predicates in the dataset, the corresponding *O* cell is marked with —.

Annotation Conversion Table 1 shows the performance of our first and second conversion method – see Section 4. In all following experiments we use the second conversion method.

			CZ	EN	DE	ES	ZH
1	u	V	87.10	84.19	81.16	83.79	85.63
		O	85.29	81.52	—	—	-
		A	86.32	83.32	81.16	83.79	85.63
	l	V	98.78	92.75	97.28	96.40	97.81
		O	99.01	89.75	—	—	—
		A	98.88	91.79	97.28	96.40	97.81
2	u	V	89.10	95.06	93.43	91.08	89.87
		O	87.80	89.94	—	—	—
		A	88.55	93.38	93.43	91.08	89.87
	l	V	100.00	100.00	100.00	100.00	100.00
		O	100.00	100.00	—	—	—
		A	100.00	100.00	100.00	100.00	100.00

Table 1: Quality of annotation conversion.

Monolingual Experiments Table 2 gives the results of our system on the SD hand-annotated data (exactly the same task that was at ConLL 2009). The results are measured with the official CoNLL 2009 evaluation script. *ls Zhao* row shows the results of the best system at the CoNLL 2009 shared task. Table 3 shows the results with automatically converted UD annotations.

Cross-lingual Table 4 compares results of the cross-lingual experiments on the UD annotated data. Labelled accuracy (*l*) is measured only on datasets with the same set of semantic labels (PCEDT for Czech and CoNLL 2009 data for Chinese).

		CZ	EN	DE	ES	ZH
us	A	96.12	91.75	90.42	100.00	92.08
ls	A	87.02	80.52	72.48	75.62	81.73
ls Zhao	A	85.19	85.44	75.99	80.46	78.15
u	V	94.06	95.82	86.52	100.00	87.81
	O	91.41	78.52	—	—	—
	A	92.97	91.07	86.52	100.00	87.81
l	V	74.16	86.02	65.95	69.02	82.99
	O	76.51	79.11	—	—	—
	A	75.11	83.28	65.95	69.02	82.99
F_1^c	V	93.05	93.64	95.09	93.00	92.98
	O	95.38	88.56	—	—	—
	A	94.83	91.10	95.09	93.00	92.99

Table 2: Results on the gold SD trees.

		CZ	EN	DE	ES	ZH
u	V	81.11	83.29	75.19	82.58	67.44
	O	72.12	59.01	—	—	—
	A	77.38	76.51	75.19	82.58	67.44
l	V	63.13	68.70	53.79	53.07	77.71
	O	69.51	64.05	—	—	—
	A	65.60	67.69	53.97	53.07	77.71
F_1^c	V	86.76	83.33	85.75	83.87	89.18
	O	93.22	85.33	—	—	—
	A	91.62	83.72	85.75	83.87	89.18

Table 3: Results with the automatically generated UD trees.

7 Discussion

From the results of the conversion methods (Table 1) we can see that our second conversion method provides better results than the first one. This is a natural outcome since the second method has a higher freedom in selection of the heads. The second method selects only correct heads and therefore the labels can be converted without errors (100% scores in the last section of Table 1). However, some SRL annotations do not permit labelling of more than one head per predicate argument and some SRL systems rely on this constraint. We have designed our systems without the requirement of one head per predicate argument and thus we use the second conversion method in all the following experiments.

		EN-CZ	EN-ES	EN-ZH	EN-DE	PCEDT
u	UD	76.28	75.67	64.08	71.66	78.97
	K	—	—	51.70	—	63.90
l	UD	—	—	75.57	—	55.09
	K	—	—	71.70	—	59.00
F_1^c	UD	84.90	81.78	87.94	82.37	83.03
	K	—	—	84.50	—	74.10

Table 4: Cross-lingual results with the UD trees. The *UD* row shows the results of our system based on UD trees and the K row shows the results from (Kozhevnikov and Titov, 2013).

The results of the monolingual experiment on the Gold standard trees in Table 2 show that our SRL system performs at a level comparable to the best systems at the CoNLL 2009 shared task (Hajič et al., 2009). Taking into account our reduced feature set (without lexical features), we can safely conclude that the selected features are sufficient for the consequent experiments.

The monolingual experiment performed on the UD trees (Table 3) shows a significant drop of the performance when compared to the Gold standard trees. However, the decrease of the performance is expected since we operate on system generated trees and the annotation conversion introduces additional errors. An additional analysis of the results revealed that most of the errors are caused by incorrect syntactic parses. The CoNLL sentences use a rather complicated structure and the parsing errors are expected. In some cases, the sentences are ambiguous – e.g. see Figure 3a. The node “for” could be easily attached under the node “creating” instead of the node “plant”. Such differences directly influence SRL labelling (see the dots in the figure). Taking into account the fact that the obtained SRL annotations are based solely upon syntactic information (we do not use lexical features), we conclude that the UD support the SRL identification and labelling well.

The most interesting results are the outcomes of the cross-lingual experiment in the Table 4. We can see that the predicate argument identification phase (the *u* row, *V* subrow) was very successful except for Chinese. However, we significantly outperformed the results presented in (Kozhevnikov and Titov, 2013) (the *u* row, *K* subrow) for both Czech and Chinese SRL annotations. We believe we can make such a statement even though our results are computed on all argument words and in (Kozhevnikov and Titov, 2013) the results are computed on heads only. From Table 2 we can see that these numbers are highly correlated. The argument classification phase (the *l* row) is similar (diffrence of about 4 points) between our system and the results presented in (Kozhevnikov and Titov, 2013). Both system are able to deal well with the Chinese dataset but the Czech dataset created from PCEDT proved to provide worse scores.

The results of the cross-lingual experiment measured by the unsupervised metrics (Table 4

– F_1^c row) indicate that the transfer of semantic roles from English to all other languages is possible. The high values of F_1^c scores of collocation and purity show that it should be possible to find a mapping of role labels with high agreement between the produced results of the system and the annotations in the evaluation data. Again, the results are better than the ones presented in (Kozhevnikov and Titov, 2013).

8 Future Work

We believe that our system can be improved in several areas. Currently, we work on employing bilingual clusters based upon the Word2Vec model (Mikolov et al., 2013). Next, we plan to train a SRL system on combined annotations from several languages and also to try transferring knowledge from other languages than English. The goal of this experiment would be to evaluate how the differences between languages affect the performance of our approach. Our approach is also capable of training model on more languages and we are currently working on that.

The biggest goal for the future is to propose language-independent semantic roles for the UD trees. Our system introduced in this paper can help significantly in this effort. It can be used for bootstrapping by pre-annotating the UD trees for humans to correct them afterwards.

By studying the UD annotations, we believe that it might be possible to construct language-independent rules for the predicate argument identification. An appropriate consequent clustering phase could be then used to create a language independent unsupervised SRL system.

9 Conclusion

In our paper, we have introduced a model for the cross-lingual SRL based upon the UD annotations. We have outperformed the similar approach presented in (Kozhevnikov and Titov, 2013). However, our primary goal was to evaluate whether the UD trees are suitable for the SRL task. We conclude that UD annotations are a very promising vehicle for creating a SRL system for a broad range of languages. We provide our methods for generating SRL annotations for UD trees freely for download.[4]

[4] Available on: goo.gl/rjjhp8.

Acknowledgments

This publication was supported by the project LO1506 of the Czech Ministry of Education, Youth and Sports and by Grant No. SGS-2016-018 Data and Software Engineering for Advanced Applications. Computational resources were provided by the CESNET LM2015042 and the CERIT Scientific Cloud LM2015085, provided under the programme ”Projects of Large Research, Development, and Innovations Infrastructures”.

References

Paolo Annesi and Roberto Basili. 2010. Cross-lingual alignment of FrameNet annotations through hidden markov models. In *Proceedings of the 11th International Conference on Computational Linguistics and Intelligent Text Processing*. Springer-Verlag, Berlin, Heidelberg, CICLing'10, pages 12–25. https://doi.org/10.1007/978-3-642-12116-6_2.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the 17th International Conference on Computational Linguistics - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, COLING '98, pages 86–90. https://doi.org/10.3115/980451.980860.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford Typed Dependencies representation. In *Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*. Association for Computational Linguistics, Stroudsburg, PA, USA, CrossParser '08, pages 1–8. http://dl.acm.org/citation.cfm?id=1608858.1608859.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational linguistics* 28(3):245–288.

Trond Grenager and Christopher D. Manning. 2006. Unsupervised discovery of a statistical verb lexicon. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP '06, pages 1–8. http://dl.acm.org/citation.cfm?id=1610075.1610077.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, et al. 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*. Association for Computational Linguistics, pages 1–18.

Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Silvie Cinková, Eva Fučíková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiří Semecký, Jana Šindlerová, Jan Štěpánek, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. 2012. Prague czech-english dependency treebank 2.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University in Prague.

Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2006. A large-scale extension of VerbNet with novel verb classes. In Cristina Onesti Elisa Corino, Carla Marello, editor, *Proceedings of the 12th EURALEX International Congress*. Edizioni dell'Orso, Torino, Italy, pages 173–184.

Michal Konkol. 2014. Brainy: A machine learning library. In Leszek Rutkowski, Marcin Korytkowski, Rafa Scherer, Ryszard Tadeusiewicz, Lotfi A. Zadeh, and Jacek M. Zurada, editors, *Artificial Intelligence and Soft Computing*. Springer International Publishing, volume 8468 of *Lecture Notes in Computer Science*, pages 490–499. https://doi.org/10.1007/978-3-319-07176-3_43.

Mikhail Kozhevnikov and Ivan Titov. 2013. Cross-lingual transfer of semantic role labeling models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*. pages 1190–1200. http://aclweb.org/anthology/P/P13/P13-1117.pdf.

Mikhail Kozhevnikov and Ivan Titov. 2014. Cross-lingual model transfer using feature representation projection. In *ACL (2)*. pages 579–585.

Joel Lang and Mirella Lapata. 2010. Unsupervised induction of semantic roles. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '10, pages 939–947. http://dl.acm.org/citation.cfm?id=1857999.1858135.

Joel Lang and Mirella Lapata. 2011. Unsupervised semantic role induction via split-merge clustering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, pages 1117–1126. http://www.aclweb.org/anthology/P11-1112.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*. pages 55–60. http://www.aclweb.org/anthology/P/P14/P14-5010.

A. Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. 2004. The NomBank project: An interim report. In A. Meyers, editor, *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*. Association for Computational Linguistics, Boston, Massachusetts, USA, pages 24–31.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781. http://arxiv.org/abs/1301.3781.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan T McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *LREC*.

Joakim Nivre and Johan Hall. 2005. Maltparser: A language-independent system for data-driven dependency parsing. In *In Proc. of the Fourth Workshop on Treebanks and Linguistic Theories*. pages 13–95.

Sebastian Padó and Mirella Lapata. 2009. Cross-lingual annotation projection for semantic roles. *Journal of Artificial Intelligence Research* 36:307–340.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics* 31(1):71–106. https://doi.org/10.1162/0891201053630264.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. European Language Resources Association (ELRA), Istanbul, Turkey.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. European Language Resources Association (ELRA), Paris, France.

Ivan Titov and Alexandre Klementiev. 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*. Association for Computational Linguistics, Stroudsburg, PA, USA, EACL '12, pages 12–22. http://dl.acm.org/citation.cfm?id=2380816.2380821.

Daniel Zeman. 2008. Reusable tagset conversion using tagset drivers. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. European Language Resources Association (ELRA), Marrakech, Morocco. Http://www.lrec-conf.org/proceedings/lrec2008/.