

Multilingual Dependency Parsing for Low-Resource Languages: Case Studies on North Saami and Komi-Zyrian

KyungTae Lim, Niko Partanen, Thierry Poibeau

LATTICE (CNRS & École normale supérieure / PSL & Université Sorbonne nouvelle Paris 3 / USPC)

1, rue Maurice Arnoux, 92120 Montrouge, France

kyungtae.lim@ens.fr, nikotapiopartanen@gmail.com, thierry.poibeau@ens.fr

Abstract

The paper presents a method for parsing low-resource languages with very small training corpora using multilingual word embeddings and annotated corpora of larger languages. The study demonstrates that specific language combinations enable improved dependency parsing when compared to previous work, allowing for wider reuse of pre-existing resources when parsing low-resource languages. The study also explores the question of whether contemporary contact languages or genetically related languages would be the most fruitful starting point for multilingual parsing scenarios.

Keywords: dependency parsing, word embeddings, Uralic languages

1. Introduction

Developing systems for low-resource languages is a crucial issue for Natural Language Processing (NLP). Most NLP systems are built using supervised learning techniques (Weiss et al., 2015; Straka et al., 2016; Ballesteros et al., 2016). These systems require a large amount of annotated data and are thus targeted toward specific languages for which this kind of data exists. Unfortunately, producing enough annotated data is known to be time- and resource-consuming, which means that annotated data, especially of the type required for parsing, is lacking for most languages. To take a recent example, the 2017 CoNLL Shared Task concerned around 50 languages, roughly all of the languages for which enough syntactically annotated data is available in the Universal Dependency format. This was probably (by far) the most ambitious parsing challenge ever undertaken with regard to language diversity, but the figure of 50 languages should be viewed relative to the 6,000 languages in the world: even if one includes only those languages for which written data is available, the 50 languages targeted at CoNLL 2017 cover only a fraction of all the world's languages.

When it comes to parsing, the supervised, monolingual approach based on syntactically annotated corpora has long been the most common one. However, thanks to recent developments involving feature-representation methods (a.k.a. word embeddings) and neural network models, it is now possible to develop accurate multilingual models, too. The multilingual approach has yielded encouraging results for both low- (Guo et al., 2015) and high-resource languages (Ammar et al., 2016a). Generally speaking, the multilingual approach can be implemented in two ways. The first involves projecting annotations available for a high-resource language onto a low-resource language using a parallel corpus, while the second aims at producing a cross-lingual transfer model that can work for several languages.

Guo et al. (2016) and Ammar et al. (2016a) have conducted multilingual parsing studies for Indo-European languages using the model transfer approach. They demonstrated that

a multilingual model can yield better results than monolingual models for different European languages. However, their approach relied on the existence of a massive parallel corpus, as their experiment was based on Europarl.¹ Thus the problem of low-resource languages remains unaddressed, especially in cases when no parallel corpus is available. In this paper, we propose a simple but powerful method for creating a dependency parsing model when no annotated corpus or parallel corpus is available for training. Our approach requires only a small bilingual dictionary and the manual annotation of a handful of sentences. We assume that the performance one can obtain with this approach depends largely on the set of languages used to train the model. This is why we have developed several models using genetically related and non-related languages, so as to gain a better understanding of the limitations or possibilities of model transfer across different language families. In this study, we are working with languages that, until very recently, did not or still do not have a Universal Dependency corpus: North Saami² and Komi-Zyrian, henceforth Komi.³ (In this paper, we use ISO 639-3 codes to refer to the languages in tables.) This alone does not make them low-resource languages, but they are still poorly equipped with regard to NLP tools. Saami language technology has been in active development for a longer time in the Gielatekno project of the University of Tromsø,⁴ but in the case of Komi, resources have only very recently begun to emerge. Work has also been done on these languages within the framework of language documentation; however, with the exception of a few individual projects (Blokland et al., 2015; Gerstenberger et al., 2016), this field has generally not paid much attention to the use of NLP tools. The same scenario applies to many other low-resource languages. On the other hand, the amount of written data in smaller languages may have grown rapidly in last years thanks to an increased online presence and various digitalization projects.

¹www.statmt.org/europarl

²glottolog.org/resource/languoid/id/nort2671

³glottolog.org/resource/languoid/id/komi1268

⁴giellatekno.uit.no

This makes it possible to build word embeddings, as the only resource needed for these is a sufficiently large amount of text.

A major contribution of this work also lies in the new resources that have been developed as a result of it. We have created a new UD-type corpus for Komi⁵ (version 0.1 used in this study), as well as bilingual dictionaries, multilingual word embeddings for Komi and Saami, and a multilingual parser, all of which are freely available in public repositories.^{6 7}

The structure of the paper is as follows. We first provide an overview of our approach (Section 2) before detailing the multilingual resource representation used (Section 3) and our parsing model (Section 4). We then describe a series of experiments aiming at validating the approach (Section 5) before presenting an analysis of our results along with some thoughts for future work (Section 6).

2. Approach

The languages used for training the models have been selected with the assumption that genetically related languages or contemporary contact languages, at least in certain scenarios, share structural similarities with the low-resource languages in question. Our aim is to test whether it is possible to use the annotated data from larger languages as part of the training data, exploiting this structural similarity.

Our work with Northern Saami was initially motivated by it being one of the surprise languages in the CoNLL 2017 Shared Task. After the Shared Task ended, the larger training data became available in the dev-branch of the UD_North_Sami GitHub repository in the Universal Dependencies project⁸, which enabled us to carry out broader testing. With Komi-Zyrian, the situation is essentially the same as it was with the North Saami before the recent release of the new data: we have 85 manually annotated examples we can use to train and test the parsing result. The training and testing set was created manually for Komi for the purpose of this study, but we are currently expanding it. This Komi-Zyrian treebank will be included in Universal Dependencies project during 2018.

Currently, only English has been used as a control language with no direct contact or genetic relation with Komi or Saami. Our expectation is that the match rate between Russian and Komi-Zyrian should be exceptionally high, as there has been such a long history of contact between these languages, leading to a variety of morphosyntactic changes in Komi (Leinonen, 2006, p. 241). North Saami has a complex contact relationship with Finnish, but in addition to this, it is also a closely genetically related language (Aikio, 2012, p. 67–69). From this point of view, one could expect Finnish to perform well in parsing both Komi and North Saami, although the similarities between these languages have not been studied in great detail from the perspective of syntax and dependency structures. Other types of experiments have also been conducted using this approach,

for example, by using a Komi-Zyrian–Russian multilingual model to parse data that contains both languages in the form of code-switching: in these tests, the parser has been shown to be able to analyse language-specific constructions when they occur within same utterance (Partanen et al., 2018).

3. Multilingual Lexical representation

3.1. Preparation of Language Resources

The bilingual lexicons were taken from the Giellatekno infrastructure (Giellatekno, 2017), as these are rather large and have been released with a GNU GPLv3 license. As some portions of this data are derived from printed dictionaries, not all entries are directly usable for our purposes. Multi-word translations and entries containing question marks have been removed. Table 1 shows the sizes of the dictionaries used.

Bilingual pairs	Bi-dictionary	Bi-embedding
Finnish–Komi	12,879	2.3GB
Finnish–North Saami	12,398	2.4GB
Komi–English	8,746	7.5GB
North Saami–Finnish	10,541	2.4GB
Russian–Komi	12,354	5.7GB

Table 1: Dictionary sizes and size of bilingual word embeddings generated by each dictionary.

We have used the pretrained Finnish and Russian FastText word embeddings published by Facebook in May 2017 (Bojanowski et al., 2016). Since the Komi and Saami Wikipedias are relatively small, we have also trained larger word embeddings using FastText. For Komi, we have used Public Domain books digitalized in the Fenno-Ugrica collection (<https://fennougrica.kansalliskirjasto.fi/>) and proofread by The Finno-Ugric Laboratory for Support of the Electronic Representation of Regional Languages in Syktyvkar (<http://komikyv.org/>). For North Saami, we have used the SIKOR North Saami free corpus (<http://hdl.handle.net/11509/100>), which has been published with a CC-BY 3.0 license.

3.2. Projection of Multiple Word Embeddings onto a Single Space

In the previous section, we described how we obtained and trained monolingual embeddings for each language, but each of those embeddings is trained in its own vector space. In order to transform the different embeddings into one single bilingual word embedding (encoded through a single vector space model), we apply the linear transformation method proposed by Artetxe et al. (2016). According to comparisons presented in Artetxe et al. (2017, p. 457), the size of dictionaries we used is well above what is needed to carry out the mapping task using this method.

The method is as follows. Let target language X and source language Y be the word embedding matrix trained by two different languages. And let $D = \{(x_i, y_i)\}_{i=1}^m$ (where $x_i \in X$, $y_i \in Y$) be a bilingual dictionary consisting of word-embedding vector pairs. Our goal is to find a transformation matrix W such that xW approximates y . This is done by

⁵ github.com/langdoc/UD_Komi-Zyrian

⁶ github.com/jujbob/multilingual-bist-parser

⁷ github.com/jujbob/multilingual-models

⁸ github.com/UniversalDependencies/UD_North_Sami

minimizing the sum of squared errors, following Mikolov et al. (2013):

$$\arg \min_W \sum_{i=1}^m \|x_i W - y_i\|^2 \quad (1)$$

However, the application of a linear transformation to an embedding without constraints may cause a degradation of performance since the computation of mapping may break monolingual invariance. Artetxe et al. (2016) proposed an orthogonal mapping method that lets W be an orthogonal matrix and makes it possible to avoid a degradation of performance.

We followed this method in order to map two different embeddings and produce bilingual word embeddings with language-specific prefixes preceding each surface forms. For example, in case of a English-French pair, the projected embedding can have “eng:dog” and “fra:chien”. To train word embeddings related to more than two languages, we first selected a standard source language S and then mapped it with each target language $T=(t_1, t_2, t_i \dots)$. Based on the trained parameter $W_T = (W_{t_1}, W_{t_2}, W_{t_i} \dots)$, we can thus build a multilingual word embedding by multiplying the parameter W_{t_i} and all the surface forms included in t_i .

4. Cross-Lingual Dependency Parsing Model

Traditionally, many parsers have applied linear supervised learning models with hand-crafted feature functions. The feature function takes features for classifying head-modifier and relations among tokens (i.e. “word forms and POS tags from first and second tokens on top of the stack”) (Kiperwasser and Goldberg (2016)). Parsers also require many templates in order to make a decision about the relations between tokens. However, extracting the proper features and templates manually is a difficult and time-consuming job. In order to address the limitation of manual work, Chen and Manning (2014) proposed using non-linear classifiers with a neural network model. This method encodes lexical (words) and non-lexical (POS tags) features as vectors and then concatenates the features of each token to feed the non-linear classifiers. This has two advantages: On the one hand, non-linear classifiers show better performance than linear models in identifying relations between tokens, and on the other hand, the use of a neural network with concatenated features alleviates the need for manual work because the neural model, especially in Recurrent Neural Networks (RNNs), has access to tokens and features computed previously for a given sentence.

Our basic feature representation approach is based on Chen and Manning (2014), with the exception of the method used for pretrained multilingual embeddings. In order to take into account lexical resources during parsing, we have extended the multilingual graph-based parser based on the bidirectional LSTM feature representations proposed by Lim and Poibeau (2017).

4.1. Bidirectional LSTM Feature Representations

Recent advances in NLP have been possible largely due to innovative feature representations that provide an accurate

overview of word relations inside the sentence (Cho, 2015; Huang et al., 2015). A good example is the BIST-parser proposed by Kiperwasser and Goldberg (2016), which is based on bidirectional LSTM learning. BiLSTM is a powerful learning model for sequential data because it consists of two LSTM layers, a *Forward* layer that reads the sentence from left to right, and another that reads it from right to left. For example, given a sentence $t = (t_1, t_2, \dots, t_n)$, in which the symbol \circ denotes a concatenation operation, the BiLSTM function can be represented as: $\text{BiLSTM}(t_{1:n}, i) = \text{LSTM}_{\text{Forward}}(t_{1:i}) \circ \text{LSTM}_{\text{Backward}}(t_{i:n})$.

4.2. Cross-Lingual Feature Representations

In their paper, Lim and Poibeau (2017) proposed a system for multilingual parsing based on bidirectional LSTM feature representations. They followed most of the approach used for the BIST-parser (see Kiperwasser and Goldberg (2016) and the previous section) and transformed it into a multi-source trainable model with multilingual word embeddings and language hot-encoding. This system obtained an LAS F1 score of 70.93% in the 2017 CoNLL Shared Task (rank 5/33). For this paper, we have extended the parser using the multilingual word embeddings proposed in Section 3, and have adapted the same token representation methods and dimensions proposed by Lim and Poibeau (2017).

Token Representation. Given an input sentence $t = (t_1, t_2, \dots, t_n)$, a word form w , a corresponding POS tag p , pretrained word embedding xw and language hot-encoding l , the i^{th} token is defined as: $t_i = e(w_i) \circ e(p_i) \circ e(xw_i) \circ e(l_i)$, where e denotes the embedding vector of each feature and $e(xw_i)$ is the pretrained word embedding introduced in Section 3. Additionally, we added a language hot-encoding vector composed of 0 and 1 for each language as proposed earlier (Naseem et al., 2012; Ammar et al., 2016a). Compared with monolingual parsers, most use $e(w_i)$ and $e(p_i)$, with additional features such as distance between head node and language-specific lexical features included in the UD corpus. Note that t_i feeds into $\text{BiLSTM}(t_{1:n}, i)$ in order to store the *Forward* and *Backward* contexts from the LSTM.

4.3. Parsing Model

There are two mainstream approaches to parsing, one being the transition-based model (Nivre, 2004) and the other the graph-based model (McDonald et al., 2005b). For this study, we chose the graph-based approach based on the BIST-parser since graph-based approaches seem to show better performance for parsing UD-type corpora (Dozat et al., 2017). From the features and tokens stored in the BiLSTM layer, the BIST-parser computes a candidate tree for each head word and modifier, after which scores attached to the different candidate trees are computed using the multi-layer perceptron (MLP), which is a basic neural network model that can be used as a scoring function. Finally, the system finds the best dependency parsing trees based on the sum of the subtrees. For further information on the graph-based and arc-factored model used in the BIST-parser, please see Taskar et al. (2005) and McDonald et al. (2005a).

Case	Training corpus	LAS	UAS
1	sme (20)	32.96	46.85
2	eng (12,217)	32.72	50.44
3	fin (12,543)	40.74	54.24
4	sme (20) + eng (12,217)	46.54	61.61
5	sme (20) + fin (12,543)	51.54	63.06

Table 2: Labeled attachment scores (LAS) and unlabeled attachment scores (UAS) for Northern Saami (sme)

5. Experiment

We conducted a series of experiments on Saami and Komi. For Saami, we tested different language combinations for the cross-lingual model. All the experiments were carried out using 20 training sentences in Saami, as was the case for the 2017 CoNLL Shared Task, which means these results can be compared to the ones in the official CoNLL evaluation. For Komi, no annotated corpus was available, but we designed ten different experiments, again exploring different language combinations for the cross-lingual model. The experiments for Komi are representative of an extremely low-resource scenario, which is quite common, meaning the approach can be reused for a wide variety of other languages.

Training Corpus. We used corpora available in the Universal Dependency 2.0 format (Nivre et al., 2017) to train and test all models except for Komi. Since there is no UD 2.0 Komi corpus, we used 10 sentences for training and 75 sentences for the testing set (the corpus was designed specifically for this study). Following previous works by Guo et al. (2015) and Zhang and Barzilay (2015), we used gold POS sets for training and testing for Komi. For Saami, however, the teams in the CoNLL Shared Task used pre-processed POS tagging sets processed by UDpipe. In order to maintain the same conditions as in the Shared Task, we also used the pre-processed POS tagging sets to compare with others in Table 3.

Training Conditions. Since we wanted to explore low-resource scenarios (even in the case of Saami, for which larger data now exists), we assumed that there had been no development data for parameter tuning, and we restricted all training experiments to run just one epoch with the training corpus without early stopping (i.e. the 5_{th} -low of column titled “Case” in Table 2 runs only 12,563 iterations). Similar restrictions have also been suggested by Ammar et al. (2016b), who proposed running one epoch, and by Guo et al. (2016), who proposed restricting iterations to 20,000 for low-resource scenarios. However, when training a model with multi-source training data, the size of training corpora for low-resource languages is comparably smaller than for high-resource languages. Following the previous work of Guo et al. (2016), we iterated 20 times more for low-resource training data than for high-resource. In Table 2 and Table 4, the sizes of the training sets used are provided in brackets.

Comparison with the CoNLL Shared Task. We used the same training environments in these experiments as for the CoNLL Shared Task (the same training sets and no development set). Moreover, in order to guarantee similar experimental conditions, we allowed the training models to

Team	LAS	UAS
C2L2 (Ithaca)	48.96	58.85
IMS (Stuttgart)	40.67	51.56
HIT-SCIR (Harbin)	38.91	52.51
This work	42.50	54.94

Table 3: The highest results of this experiment compared with top 3 results for Saami from the CoNLL 2017 Shared Task.

Case	Training corpus	LAS	UAS
1	kpv (10)	22.33	51.78
2	eng (12,217)	44.47	59.29
3	rus (3,850)	53.85	71.29
4	fin (12,543)	48.22	66.98
5	kpv (10) + eng (12,217)	50.47	66.23
6	kpv (10) + rus (3,850)	53.1	69.98
7	kpv (10) + fin (3,850)	53.66	71.29
8	kpv (10) + fin (12,543)	55.16	73.73
9	kpv (10) + eng (12,217) + fin (12,543)	52.5	68.57
10	kpv (10) + rus (3,850) + fin (12,543)	56.66	71.86

Table 4: Labeled attachment scores (LAS) and unlabeled attachment score (UAS) for Komi (kpv). We doesn’t conduct training for “kpv + eng + rus” language combination because of unrealistic training scenario (It takes more than 40GB memory for training)

run just one epoch, following Guo et al. (2016). Table 3 reports the official 2017 CoNLL results. C2L2 (Cornell Univ.) obtained the best performance for Saami with a delexicalized transfer approach (using a Finnish training corpus and a corpus of 20 Saami sentences as a development set for parameter tuning without lexicalized features). IMS (Stuttgart) used a delexicalized transfer approach with a very large training corpus based on 40 different training corpora in UD, obtaining the second-best result. Compared with the result of the Shared Task, it seems that our approach (lexicalized cross-lingual transfer parsing with resources from relevant languages) can be effective for parsing low-resource languages. However, additional language features and the application of the ensemble mechanism also seem to be very important. This is due to the fact that the team C2L2 outperform others with more than 6.4 percent of LAS score based on character embeddings as additional features and an ensemble mechanism composed of three different dependency parsers.

6. Analysis

All the experiments we conducted using Finnish for training obtained better results than other language combinations (i.e. English for Saami and Russian and English for Komi). This means that transferring knowledge from genetically related languages is, at least in our case, a very efficient method for parsing. This is true in the case of Saami and Finnish, but also in the case of Finnish and Komi, which are arguably more distantly related to one another than Finnish and Saami. A contact language can also be of significant help in improving the results with a low-resource language, as can be seen in the case of Russian and Komi.

Even in cases where the LAS scores (i.e. labels of the syntactic dependencies) were not significantly different, there was often greater variation in the UAS scores (unlabeled dependencies). This means that although the actual labels may not have been assigned correctly, the basic relations were found and the root was correctly recognized. It is also worth noting that the highest UAS scores were obtained with the Komi–Finnish pair, although the Komi–Russian–Finnish multilingual model gave the best overall result. Our results show that adding even a small number of target-language example sentences into a parser that uses bilingual word embeddings can improve the result significantly. The size of the available training corpus is very important, and the quality of the bilingual dictionaries used to align word embeddings is also crucial.

7. Conclusion

In this paper, we have presented a multilingual approach to parsing that is effective for languages with few resources and no syntactically annotated corpora available for training. We have shown that our multilingual models provide better results than monolingual ones. Adding training material from other languages usually did not decrease the parsing result. It should, however, be noted that the relative size of the different corpora used for training seems to be relevant, since using corpora that are too imbalanced may weaken the result. More detailed analysis of the results, beyond the LAS and UAS scores, is most likely needed in order to determine the exact influences of different language pairs or combinations. It remains a question for further study whether the improvements observed here are actually attributable to the genetic relationship between the language, or if the same result could be obtained by simply selecting languages that are otherwise typologically similar.

Acknowledgments

We want to thank the LREC reviewers for their useful comments. This work has been developed in the framework of the LAKME project funded by a grant from Paris Sciences et Lettres (IDEX PSL reference ANR-10-IDEX-0001-02). Thierry Poibeau is partially supported by a RGNF-CNRS (grant between the LATTICE-CNRS Laboratory and the Russian State University for the Humanities in Moscow). Kyungtae Lim is partially supported by the ERA-NET Atlantis project. Niko Partanen’s work has been carried out at the LATTICE laboratory within the project ‘Language Documentation meets Language Technology: the Next Step in the Description of Komi’, funded by the Kone Foundation. Thanks to Alexandra Kellner for proofreading the paper.

8. Bibliographical References

- Aikio, A. (2012). An essay on Saami ethnolinguistic prehistory. In Riho Grünthal et al., editors, *A Linguistic Map of Prehistoric Northern Europe*, pages 63–117. Société Finno-Ougrienne.
- Ammar, W., Mulcaire, G., Ballesteros, M., Dyer, C., and Smith, N. A. (2016a). One parser, many languages. *CoRR*, abs/1602.01595.
- Ammar, W., Mulcaire, G., Tsvetkov, Y., Lample, G., Dyer, C., and Smith, N. A. (2016b). Massively multilingual word embeddings. *arXiv preprint arXiv:1602.01925*.
- Artetxe, M., Labaka, G., and Agirre, E. (2016). Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *EMNLP*, pages 2289–2294.
- 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.
- Ballesteros, M., Goldberg, Y., Dyer, C., and Smith, N. A. (2016). Training with exploration improves a greedy stack lstm parser. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2005–2010, Austin, Texas, November. Association for Computational Linguistics.
- Blokland, R., Fedina, M., Gerstenberger, C., Partanen, N., Rießler, M., and Wilbur, J. (2015). Language documentation meets language technology. In *Septentrio Conference Series*, number 2, pages 8–18.
- Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.
- Chen, D. and Manning, C. D. (2014). A fast and accurate dependency parser using neural networks. In *EMNLP*, pages 740–750.
- Cho, K. (2015). Natural language understanding with distributed representation. *arXiv preprint arXiv:1511.07916*.
- Dozat, T., Qi, P., and Manning, C. D. (2017). Stanford’s graph-based neural dependency parser at the conll 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, Vancouver, Canada, August. Association for Computational Linguistics.
- Gerstenberger, C., Partanen, N., Rießler, M., and Wilbur, J. (2016). Utilizing language technology in the documentation of endangered Uralic languages. *Northern European Journal of Language Technology*, 4:29–47.
- Guo, J., Che, W., Yarowsky, D., Wang, H., and Liu, T. (2015). Cross-lingual dependency parsing based on distributed representations. In *Proceedings of ACL*.
- Guo, J., Che, W., Yarowsky, D., Wang, H., and Liu, T. (2016). A representation learning framework for multi-source transfer parsing. In *AAAI*, pages 2734–2740.
- Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.
- Kiperwasser, E. and Goldberg, Y. (2016). Simple and accurate dependency parsing using bidirectional LSTM feature representations. *TACL*, 4:313–327.
- Leinonen, M. (2006). The russification of Komi. Number 27 in *Slavica Helsingiensia*, pages 234–245. Helsinki University Press.
- Lim, K. and Poibeau, T. (2017). A system for multilingual dependency parsing based on bidirectional lstm feature representations. In *Proceedings of the CoNLL 2017*

Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pages 63–70, Vancouver, Canada, August. Association for Computational Linguistics.

McDonald, R., Crammer, K., and Pereira, F. (2005a). Online large-margin training of dependency parsers. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 91–98. Association for Computational Linguistics.

McDonald, R., Pereira, F., Ribarov, K., and Hajič, J. (2005b). Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT '05*, pages 523–530, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013). Exploiting similarities among languages for machine translation.

Naseem, T., Barzilay, R., and Globerson, A. (2012). Selective sharing for multilingual dependency parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 629–637. Association for Computational Linguistics.

Nivre, J., Agić, Ž., Ahrenberg, L., et al. (2017). Universal dependencies 2.0, CoNLL 2017 shared task development and test data. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University.

Nivre, J. (2004). Incrementality in deterministic dependency parsing. In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, pages 50–57. Association for Computational Linguistics.

Partanen, N., Lim, K., Rießler, M., and Poibeau, T. (2018). Dependency parsing of code-switching data with cross-lingual feature representations. In Tommi A. Pirinen, et al., editors, *Proceedings of the 4th International Workshop on Computational Linguistics for Uralic languages*, ACL Anthology, pages 1–17. Association for Computational Linguistics.

Straka, M., Hajič, J., and Straková, J. (2016). UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia. European Language Resources Association.

Taskar, B., Chatalbashev, V., Koller, D., and Guestrin, C. (2005). Learning structured prediction models: A large margin approach. In *Proceedings of the 22nd international conference on Machine learning*, pages 896–903. ACM.

Weiss, D., Alberti, C., Collins, M., and Petrov, S. (2015). Structured training for neural network transition-based parsing. *CoRR*, abs/1506.06158.

Zhang, Y. and Barzilay, R. (2015). Hierarchical low-rank tensors for multilingual transfer parsing. Association for Computational Linguistics.

9. Language Resource References

Giellatekno. (2017). *The Northern Sami-Finnish-Northern Sami Dictionaries of Neahttagisánnit*. Available online, SVN revision r157659, ISLRN 452-785-459-365-4.