# Neural Scoring Function for MST Parser

**Jindřich Libovický**
Charles University in Prague
Malostranské náměstí 25, Praha, Czech Republic
libovicky@ufal.mff.cuni.cz

**Abstract**

Continuous word representations appeared to be a useful feature in many natural language processing tasks. Using fixed-dimension pre-trained word embeddings allows avoiding sparse bag-of-words representation and to train models with fewer parameters. In this paper, we use fixed pre-trained word embeddings as additional features for a neural scoring function in the MST parser. With the multi-layer architecture of the scoring function we can avoid handcrafting feature conjunctions. The continuous word representations on the input also allow us to reduce the number of lexical features, make the parser more robust to out-of-vocabulary words, and reduce the total number of parameters of the model. Although its accuracy stays below the state of the art, the model size is substantially smaller than with the standard features set. Moreover, it performs well for languages where only a smaller treebank is available and the results promise to be useful in cross-lingual parsing.

**Keywords:** dependency parsing, MST Parser, word embeddings, Matrix-tree theorem

## 1. Introduction

Syntactic parsing is one of the oldest problems studied by Computational Linguistics. Recently, dependency parsing gained big popularity with graph-based (McDonald et al., 2005) and transition-based parsing (Nivre, 2006) being two major paradigms.

In this paper, we work with MST parser – the most often used graph-based approach. We present several experiments showing how pre-trained word embeddings could be used instead of explicit categorial (one-hot) lexical features. Training of the word embeddings is based on language modeling, so it can be trained on plain text with any further annotation. They can be computed on much bigger data than the treebanks used for training the parser. By having the lexical information in form of embeddings, the model can learn how to deal with words that were not seen in the training part of the treebank (out-of-vocabulary words).

The simple neural scoring function that we propose can be used to avoid handcrafting feature templates for feature conjunctions. We use the Matrix-Tree theorem (Chaiken and Kleitman, 1978) to sum all possible sentence parses to be able to use a gradient learning procedure as proposed by Koo et al. (2007). Using such a scoring function also allows to control the number of parameters of the model.

In the next section, we summarize related work. Section 3. describes the model we use for training. In Section 4. we evaluate the model. Section 5. then concludes the work.

## 2. Related Work

In the MSTParser by McDonald et al. (2005), parsing is modeled as finding the maximum spanning tree in a weighted oriented complete graph with vertices representing words of the sentence. The purpose of the learning procedure is therefore to find a function that favors the edges which are actual dependencies in the sentence. The advantage of using such a model is that it can naturally produce non-projective parses.

Bansal et al. (2014) tried to use word embeddings to improve the MSTParser. First, they used the word embeddings dimensions as features for a linear model, which did not bring any success. This can be explained by findings of Szegedy et al. (2013) who claimed that in neural models, the information is encoded by properties of the whole vector space induced by the neural network rather than the individual dimensions.

What actually appeared to be a helpful feature for Bansal et al. (2014) was word clustering based on the cosine distance of word embeddings and adding it to the feature vector the same way Brown cluster can be used (Koo et al., 2008). Because the training procedure of the continuous-bag-of-words embeddings they used (Mikolov et al., 2013) could be seen as an implicit factorization of word co-occurrence matrix (Levy and Goldberg, 2014), we can expect the resulting clustering to be very similar to Brown clusters (Brown et al., 1992).

Chen and Manning (2014) used pre-trained word embeddings and Part-Of-Speech (POS) tag embeddings in a shift-reduce parser and managed to achieve a state-of-the-art performance on English without explicitly using lexical features. This approach has been recently even improved by Weiss et al. (2015) and Dyer et al. (2015).

Using the Matrix-Tree theorem for MST parsing is not novel. Apart from the previously mentioned use (Koo et al., 2007), it was independently introduced by McDonald and Satta (2007) and Smith and Smith (2007).

## 3. Model

We use the *MSTperl* implementation (Rosa et al., 2012) of the first-order MST parser, originally introduced by McDonald et al. (2005). Edges in a sentence graph are scored using a neural scoring function which we describe in Section 3.1.. Training such a function requires a differentiable loss function. The loss function we use for training is discussed in Sections 3.2. (negative log-likelihood) and 3.3. (Hinge loss). The function parameters are estimated using the stochastic gradient descent algorithm with the er-

ror back-propagated to the scoring function (LeCun et al., 1998).

### 3.1. Scoring Function

We score edges using a function implemented by a neural network with one hidden layer with the 'tanh' activation function. The output of the function is computed as a linear combination of the hidden layer outputs without non-linearity. For an edge $e$ and parametrization $\theta$, we denote the scoring function $f(e|\theta)$. There are four groups of features used in our experiments:

- positional information – bucketed edge length, sentence length, relative order in the sentence, indication of being first and last in the sentence;
- POS tags of edge words and their linear neighbors;
- word embeddings of edge words and their linear neighbors (fixed; not updateid during the training);
- lemmas of edge words and their neighbors if among 200 most frequent lemmas.

The technical root is assigned embedding of the end-of-sentence token.

### 3.2. Cross-Entropy Loss

Assuming all possible parses of a sentence are independent on each other, we can model their probability with a multinomial distribution over the set of all possible parses – all oriented trees rooted in the technical root of the sentence. Probability of a tree $T$ in a graph $G$ can be expressed as a soft-max function of a tree score defined as a sum of its edge scores:

$$\mathsf{P}\left(T|G,\theta\right) = \frac{\exp\left\{\sum_{e\in T} f(e|\theta)\right\}}{Z(G)}. \tag{1}$$

Computing the partition function $Z(G)$ requires summation over all spanning trees which is intractable in the explicit form. Similarly to Koo et al. (2007), we solve this by using the Matrix-Tree theorem (Chaiken and Kleitman, 1978). We define the weighted Laplacian of the graph $G$ as:

$$L_{ij} = \begin{cases} -\exp\{f(e_{ij}|\theta)\} & i \neq j, \\ \sum_{k\neq i} \exp\{f(e_{ik}|\theta)\} & i = j. \end{cases} \tag{2}$$

According to the Matrix-Tree theorem, we omit the row and the column that correspond to the parse root. The determinant of such a matrix is equal to the products of the edge scores, summed over all correctly rooted spanning trees in the graph. Because we use exponential of the edge scores in the definition of the Laplacian, we add products of these exponential, which are actually exponentials of the edge score sum as in Equation 1:

$$\begin{aligned} Z(G) &= \det(-L') = (-1)^n \det(L') \\ &= \sum_{T'} \exp \sum_{e\in T'} f(e|\theta) \end{aligned} \tag{3}$$

where $L'$ is the weighted Laplacian of the graph $G$ modified by omitting the column and row that correspond to the root of the spanning tree and $n$ is the length of the sentence. That negative modified Laplacian is used because for a graph with even number of vertices, the determinant of the modified Laplacian is the negative sum of the spanning tree scores.

We use the negative log likelihood of the correct parse tree $T^*$ as the loss function. For data set $\mathcal{D} = \{(G_i, T_i^*)\}_{i=1}^N$, the loss is:

$$\begin{aligned} \mathcal{L}(\mathcal{D}|\theta) &= -\sum_{(G,T^*)\in\mathcal{D}} \log \mathsf{P}\left(T_i^*|G_i,\theta\right) \\ &= -\sum_{(G,T^*)\in\mathcal{D}} \sum_{e\in T^*} f(e|\theta) - \log Z(G). \end{aligned} \tag{4}$$

Direct computation of the logarithm of a determinant is numerically unstable, so we use the sum of logarithms of the diagonal matrix of the QR matrix decomposition.

By differentiating the loss function with respect to $\theta$ we get:

$$-\sum_{(G,T^*)\in\mathcal{D}} \left( \sum_{e\in T^*} \frac{\partial f(e|\theta)}{\partial\theta} - \operatorname{tr}\left[ L'^{-1} \cdot \frac{\partial L'}{\partial\theta} \right] \right). \tag{5}$$

Matrix $L'$ is often close from being singular, so the computation of the gradient also suffers with numeric instability. We therefore clip the values of adjacency matrix such that the minimum values in the matrix are at most $\exp(20)$ times smaller than the maximum.

### 3.3. Hinge Loss

One drawback of the previously described loss is the assumption of all possible parses being independent on each other because we can expect that spanning trees which share many edges are more correlated than those which do not share edges.

MST parsers are usually trained using variants of the Structured Perceptron algorithm (Collins, 2002). Using the Hinge loss is thus analogous to doing the Perceptron update in a linear model. Moreover, because we are able to efficiently do the loss augmented inference with respect to the Hamming loss, we can easily express the loss in the maximum-margin form as in the case of the structured SVM (Joachims et al., 2009).

Formally, we want the scoring function to separate the correct spanning tree $T^*$ from each possible spanning tree $T$ by at least their Hamming distance (number of edges in which the two trees differ):

$$\Phi(T^*|\theta) \geq \Delta(T^*,T) + \Phi(T|\theta) \tag{6}$$

where $\Phi(T|\theta) = \sum_{e\in T} f(e|\theta)$. If the conditions hold, they must be satisfied also for the highest possible value of the right side in the inequality (6); it thus suffices to check whether:

$$\Phi(T^*|\theta) \geq \max_T \left(\Delta(T^*,T) + \Phi(T|\theta)\right). \tag{7}$$

If this is not satisfied, we do a weight update proportional to the error. From that, we get the Hinge loss for a single data point $(G, T^*)$ as:

$$\max\left(0, \max_T \left\{\Delta(T^*,T) + \Phi(T|\theta)\right\} - \Phi(T^*|\theta)\right), \tag{8}$$

which is differentiable almost everywhere and computation of the gradient is straightforward.

| input vector | input size | # parameters | Cross Entropy | Hinge Loss |
|---|---|---|---|---|
| distance only | 19 | 190 | .297 ± .017 | .133 ± .010 |
| distance + POS | 163 | 71k | .697 ± .018 | .620 ± .021 |
| distance + vec. | 619 | 191k | .657 ± .020 | .622 ± .020 |
| distance + POS + vec. | 763 | 477k | .728 ± .020 | .626 ± .019 |
| distance + POS + 200 lemmas | 1819 | 1.3M | .700 ± .020 | .618 ± .021 |
| distance + POS + vec. + 200 lemmas | 2419 | 2.3M | .727 ± .021 | .622 ± .020 |
| baseline – left branching | 0 | | .275 ± .013 | |
| MSTParser (McDonald et al., 2005) | | | .844 | |
| MSTPerl (Rosa et al., 2012) | 13.6M | | .809 | |
| MSTPerl, delexcalized (Rosa et al., 2012) | 1.0M | | .709 | |
| state-of-the-art (Bohnet et al., 2013) | | | .890 | |

Table 1: The unlabeled attachment score for various configurations of the input vector on the Prague Dependency Treebank. Note that state-of-the art result was achieved using a transition parser.

The loss-augmented inference can be done efficiently by increasing the weight by one for all edges which are not part of the correct solution. In practice, computing the loss-augmented inference is much slower than computing the complex loss function and derivative in the case of the cross-entropy loss.

## 4. Experiments

### 4.1. Experimental settings

We used the Czech-language Prague Dependency Treebank (Böhmová et al., 2003) for initial experiments with various feature sets. The training part consists of 68,562 sentences, the test part of 9,270 sentences. We split the test data into 50 parts and report the mean and standard deviation of the unlabeled attachment score (UAS). We experimented with different settings of the feature groups to investigate the role of particular features.

For the multilingual experiments, we used the harmonized multilingual treebank collection HamleDT (Zeman et al., 2014) containing treebanks for 30 languages out of which we selected four with various sizes. We use the Prague-style annotations for the dependencies with the Universal POS tagset (Petrov et al., 2012).

The word embeddings were trained using the continuous bag-of-words model (Mikolov et al., 2013) on corpora created from Wikipedia for all the languages. All numerical tokens were replaced with zero and tokens appearing less than 10 times were replaced with a special symbol. We used a context window of size 4 and embeddings of dimension 100.

For the scoring function, we use a neural network with a single hidden layer of the size equal to the half of the input vector dimension. We also experimented with different numbers of hidden layers and different sizes. While varying these parameters in a reasonable range, the results were unaffected, although the training time increases noticeably while increasing the number of parameters of the model.

### 4.2. Results

The results for various inputs are tabulated in Table 1. The big difference between the scores while using only the positional features and the concatenation of positional features with word embeddings shows that the embeddings indeed carry a lot of information about the words. On the other hand, significantly better result are achieved by using the POS tags. Another improvement is brought by combining both the word vectors and the POS tags. Adding the lexical features in one-hot representation does not further improve the performance. This confirms that the word embeddings are more informative than categorial representation of words of similar dimension. In delexicalized parsing, we achieved the same UAS score as MST Parser with linear scoring function using a model with much fewer parameters.

When comparing the loss functions, we can see that the cross-entropy loss performs consistently ten percent better. Probably, the property of having information about the whole set of possible solutions in one continuous variable outweighs the non-realistic assumption of the mutual independence of parse trees. According to the results of Koo et al. (2007), we can expect that using a different optimization algorithm, such as exponentiated gradient (Bartlett et al., 2004), may lead to better result, although it assumes the independence as in the case of the cross-entropy loss.

The evaluation of experiments with other languages is tabulated in Table 2. It shows that the word embeddings trained on bigger data than the treebank are more useful for languages for which only a small treebank is available.

| Language | # sent. | MSTPerl | | NMST |
|---|---|---|---|---|
| | | Lex | Delex | |
| Czech | 25k | .809 | .709 | .727 |
| English | 40k | .847 | .765 | .769 |
| Estonian | 1k | .851 | .810 | .850 |
| German | 38k | .846 | .775 | .768 |
| Hungarian | 6k | .776 | .725 | .734 |
| Spanish | 16k | .854 | .797 | .786 |

Table 2: The unlabeled attachment score for selected treenbanks from HamleDT (Zeman et al., 2014) comparing lexicalized and delexicalized MSTPerl parser (Rosa, 2015) with our neural scoring function with positions, embeddings and POS tags on input.

## 5. Conclusions

We introduced a MST parser with a neural scoring function using pre-trained word embeddings as its inputs. We successfully employed word embeddings as a feature for the parser which improved the performance of the delex-

icalized parser. This improvement was higher than while using categorial representation of 200 most frequent lemmas. This result appears to be consistent among various languages.

Our original intuition was that word embeddings should be able to fully replace (and probably even improve upon) categorial lexical features. The fact that it did not appear to be true could suggest that a different vector representation of words could have been used – either more informative pretrained embedding (Turian et al., 2010; Pennington et al., 2014) or to learn the embeddings during the parser training or using character-based embeddings being able capture morphological features of out-of-vocabulary words (Ling et al., 2015).

Despite not outperforming the lexicalized parser with handcrafted features, it works with the substantially smaller model (not counting the table with embeddings).

The models whose only source of lexical information are the word embeddings could be used for cross-lingual parsing. The results give a hope that word embeddings could be used to introduce additional lexical information. Automatic word alignment can be used to train a translation function for word embeddings and these translated embeddings then used as lexical information on the input or we can use directly cross-lingual word embeddings (Klementiev et al., 2012).

## Acknowledgements

I would like to thank Rudolf Rosa for providing a tutorial for his Perl implementation of MST parser and Terry Koo for sharing with us numerical tricks they have used in their previous work (Koo et al., 2007). I also thank Pavel Pecina and the reviewers for valuable comments.

This research has been supported by the Charles University Grant Agency project No. 5235 and by the SVV project No. 260 224. The work has been using language resources developed, stored and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project No. LM2015071).

## References

Bansal, M., Gimpel, K., and Livescu, K. (2014). Tailoring continuous word representations for dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 809–815, Baltimore, Maryland, June. Association for Computational Linguistics.

Bartlett, P. L., Collins, M., Taskar, B., and McAllester, D. A. (2004). Exponentiated gradient algorithms for large-margin structured classification. In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada]*, pages 113–120.

Bohnet, B., Nivre, J., Boguslavsky, I., Farkas, R., Ginter, F., and Hajič, J. (2013). Joint morphological and syntactic analysis for richly inflected languages. *Transactions of the Association for Computational Linguistics*, 1:415–428.

Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Böhmová, A., Hajič, J., Hajičová, E., and Hladká, B. (2003). The prague dependency treebank. In Anne Abeillé, editor, *Treebanks*, volume 20 of *Text, Speech and Language Technology*, pages 103–127. Springer Netherlands.

Chaiken, S. and Kleitman, D. (1978). Matrix tree theorems. *Journal of Combinatorial Theory, Series A*, 24(3):377–381.

Chen, D. and Manning, C. (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar, October. Association for Computational Linguistics.

Collins, M. (2002). 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*, pages 1–8. Association for Computational Linguistics, July.

Dyer, C., Ballesteros, M., Ling, W., Matthews, A., and Smith, N. A. (2015). Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China, July. Association for Computational Linguistics.

Joachims, T., Finley, T., and Yu, C.-N. (2009). Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59.

Klementiev, A., Titov, I., and Bhattarai, B. (2012). Inducing crosslingual distributed representations of words. In *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India*, pages 1459–1474.

Koo, T., Globerson, A., Carreras, X., and Collins, M. (2007). Structured prediction models via the matrix-tree theorem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 141–150, Prague, Czech Republic, June. Association for Computational Linguistics.

Koo, T., Carreras, X., and Collins, M. (2008). Simple semi-supervised dependency parsing. In *Proceedings of ACL-08: HLT*, pages 595–603, Columbus, Ohio, June. Association for Computational Linguistics.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Levy, O. and Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In Z. Ghahramani, et al., editors, *Advances in Neural Information Processing Systems 27*, pages 2177–2185. Curran Associates, Inc.

Ling, W., Dyer, C., Black, A. W., Trancoso, I., Fermandez, R., Amir, S., Marujo, L., and Luis, T. (2015). Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1520–1530, Lisbon, Portugal, September. Association for Computational Linguistics.

McDonald, R. and Satta, G. (2007). On the complexity of non-projective data-driven dependency parsing. In *Proceedings of the Tenth International Conference on Parsing Technologies*, pages 121–132, Prague, Czech Republic, June. Association for Computational Linguistics.

McDonald, R., Pereira, F., Ribarov, K., and Hajič, J. (2005). Non-projective dependency parsing using spanning tree algorithms. In *In Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In C.J.C. Burges, et al., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Nivre, J. (2006). *Inductive Dependency Parsing (Text, Speech and Language Technology)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

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

Rosa, R., Dušek, O., Mareček, D., and Popel, M. (2012). Using parallel features in parsing of machine-translated sentences for correction of grammatical errors. In *Proceedings of Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-6), ACL*, pages 39–48, Jeju, Korea. Association for Computational Linguistics.

Rosa, R. (2015). Multi-source cross-lingual delexicalized parser transfer: Prague or stanford? In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 281–290, Uppsala, Sweden. Uppsala University.

Smith, D. A. and Smith, N. A. (2007). Probabilistic models of nonprojective dependency trees. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 132–140, Prague, Czech Republic, June. Association for Computational Linguistics.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. (2013). Intriguing properties of neural networks. *CoRR*, abs/1312.6199.

Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394, Stroudsburg, PA, USA. Association for Computational Linguistics.

Weiss, D., Alberti, C., Collins, M., and Petrov, S. (2015). Structured training for neural network transition-based parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 323–333, Beijing, China, July. Association for Computational Linguistics.

Zeman, D., Dušek, O., Mareček, D., Popel, M., Ramasamy, L., Štěpánek, J., Žabokrtský, Z., and Hajič, J. (2014). HamleDT: Harmonized multi-language dependency treebank. *Language Resources and Evaluation*, 48(4):601–637.