MSTParser Model Interpolation for Multi-source Delexicalized Transfer

Rudolf Rosa and **Zdeněk Žabokrtský**
Charles University in Prague, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, Prague, Czech Republic
{rosa, zabokrtsky}@ufal.mff.cuni.cz

Abstract

We introduce interpolation of trained MSTParser models as a resource combination method for multi-source delexicalized parser transfer. We present both an unweighted method, as well as a variant in which each source model is weighted by the similarity of the source language to the target language. Evaluation on the HamleDT treebank collection shows that the weighted model interpolation performs comparably to weighted parse tree combination method, while being computationally much less demanding.

1 Introduction

The task of delexicalized dependency parser transfer (or delex transfer for short) is to train a parser on a treebank for a source language (src), using only non-lexical features, most notably part-of-speech (POS) tags, and to apply that parser to POS-tagged sentences of a target language (tgt) to obtain dependency parse trees. Delex transfer yields worse results than a supervised lexicalized parser trained on the tgt language treebank. However, for languages with no treebanks available, it may be useful to obtain at least a lower-quality parse tree for tasks such as information retrieval.

Usually, multiple src treebanks are available, and it is non-trivial to select the best one for a given tgt language. Therefore, information from some or all src treebanks is usually combined together. The standard ways are to train a parser on the concatenation of all src treebanks, or to train a separate parser on each src treebank and to combine the parse trees produced by the parsers using a maximum spanning tree algorithm. The tree combination method typically performs better; it can also be easily extended by weighting the src parser predictions by similarity of the src language to the tgt language, which can further improve its results.

In this work, we present a novel method for src information combination, based on interpolation of trained parser models. Our approach was motivated by an intuition that the more fine-grained information provided by the src edge scores could be of benefit, probably serving as src parser confidence. Moreover, model interpolation is significantly less computationally demanding at inference than the parse tree combination method, as instead of running a set of separate src parsers, only one parser is run.

2 Related Work

Delex transfer was conceived by Zeman and Resnik (2008), who also introduced two important preprocessing steps – mapping treebank-specific POS tagsets to a common set using Interset (Zeman, 2008), and harmonizing treebank annotation styles into a common style, which later led to the HamleDT harmonized treebank collection (Zeman et al., 2012).

McDonald et al. (2011) applied delex transfer in a setting with multiple src treebanks available, finding that the problem of selecting the best src treebank without access to a tgt language treebank for evaluation is non-trivial, and proposed the treebank concatenation method as a solution. Søgaard and Wulff (2012) introduced weighting into the method, using a POS n-gram model trained on a tgt POS-tagged corpus to weight src sentences in a weighted perceptron learning scenario (Cavallanti et al., 2010); due to its large computational complexity, we only compare to the unweighted variant in our paper.

The parse tree combination method was introduced by Sagae and Lavie (2006) for a supervised monolingual setting, optionally weighting each src parser with a weight based on its accuracy. In (Rosa and Žabokrtský, 2015), we ported

Proceedings of the 14th International Conference on Parsing Technologies, pages 71–75,
Bilbao, Spain; July 22–24, 2015. ©2015 Association for Computational Linguistics

the method to a crosslingual setting by combining delex parsers for different languages, weighted by src-tgt language similarity; we largely build upon that work in this paper.

Other possibilities of estimating src-tgt language similarity for delex transfer include employment of WALS (Dryer and Haspelmath, 2013), focusing e.g. on genealogy distance and word-order features, as done by Naseem et al. (2012) and Täckström et al. (2013), among others.

We are not aware of any prior work on interpolating dependency parser models. However, there is work on interpolating trained phrase-structure parsers, both in a monolingual setting for domain adaptation by McClosky et al. (2010), as well as in a multilingual setting by Cohen et al. (2011).

3 Method

In this section, we present our suggested approach of combining information from multiple src treebanks for parsing tgt language sentences in a crosslingual delex transfer scenario. The method proceeds as follows:

1. Train a delex parser model on each src treebank (Section 3.1).
2. Normalize the parser models (Section 3.2).
3. Interpolate the parser models (Section 3.3).
4. Parse the tgt text with a delex parser using the interpolated model.

3.1 Delexicalized MSTParser

Throughout this work, we use MSTperl (Rosa, 2015b), an unlabelled first-order non-projective single-best implementation of the MSTParser of McDonald et al. (2005b), trained using 3 iterations of MIRA (Crammer and Singer, 2003).

The MSTParser model uses a set of binary features F that are assigned weights w_f by training on a treebank. When parsing a sentence, the parser constructs a complete weighted directed graph over the tokens of the input sentence, and assigns each edge e a score s_e which is the sum of weights of features that are active for that edge:

$$s_e = \sum_{\forall f \in F} f(e) \cdot w_f \,. \tag{1}$$

The sentence parse tree is the maximum spanning tree over that graph, found using the algorithm of Chu and Liu (1965) and Edmonds (1967).

The delex feature set we use is based on the set of McDonald et al. (2005a) with lexical features removed. It consists of combinations of signed edge length (distance of head and parent, bucketed for values above 4 and for values above 10) with POS tag of the head, dependent, their neighbours, and all nodes between them. We use the Universal POS Tagset (UPOS) of Petrov et al. (2012). The parser configuration files containing the full feature set, together with the scripts we used for our experiments, are available in (Rosa, 2015a).

3.2 Model Normalization

An important preliminary step to model interpolation is to normalize each of the trained models, as the feature weights in models trained over different treebanks are often not on the same scale (we do not perform any regularization during the parser training). We use a simplified version of normalization by standard deviation. First, we compute the uncorrected sample standard deviation of the weights of the features in the model as

$$s_M = \sqrt{\frac{1}{|M|} \sum_{\forall f \in M} (w_f - \bar{w})^2} \,, \tag{2}$$

where $\bar{w}$ is the average feature weight, and $|M|$ is the number of feature weights in model M; only features that were assigned a weight by the training algorithm are taken into account.

We then divide each feature weight by the standard deviation:[1]

$$\forall f \in M : w_f := \frac{w_f}{s_M} \,. \tag{3}$$

The choice of normalization by standard deviation is based on its high and stable performance on our development set, and Occam's razor.[2]

3.3 Model Interpolation

The interpolated model is a linear combination of the normalized models trained over the src treebanks. The result is a model that can be used in the same way as a standard MSTParser model.

[1] We have not found any further gains in performance when subtracting the sample mean from the weight before the division; the MSTParser models seem to be typically centered very similarly.

[2] We tried 12 normalization schemes, nearly all of which achieved an improvement of 2.5% to 5% UAS absolute over an interpolation of unnormalized models on average, but often with large differences for individual languages. Another well-performing method was to divide each feature weight by the sum of absolute values of all feature weights in the model; or a similar method, applied during inference individually for each sentence, using only the feature weights that fired for the sentence to compute the divisor.

In unweighted model interpolation, the weight of each feature (w_f) is computed as the sum of the weights of that feature in the src models ($w_{f,src}$):

$$\forall f \in F : w_f = \sum_{\forall src} w_{f,src} \,. \quad (4)$$

In the weighted variant of model interpolation, we extend (4) with multiplication by the $KL^{-4}_{cpos^3}$ weight of Rosa and Žabokrtský (2015):

$$\forall f \in F : w_f = \sum_{\forall src} w_{f,src} \cdot KL^{-4}_{cpos^3}(tgt, src) \,. \quad (5)$$

The $KL^{-4}_{cpos^3}$ weight corresponds to the similarity of the src language to the tgt language, and is defined as the negative fourth power of the KL divergence (Kullback and Leibler, 1951) of coarse POS tag trigram distributions in tgt and src corpora:

$$KL^{-4}_{cpos^3}(tgt, src) = \left(\sum_{\substack{\forall cpos^3 \\ \in tgt}} f_{tgt}(cpos^3) \cdot \log \frac{f_{tgt}(cpos^3)}{f_{src}(cpos^3)} \right)^{-4} , \quad (6)$$

where $cpos^3$ is a UPOS trigram, and $f(cpos^3)$ is its relative frequency in a src or tgt corpus.[3]

4 Baseline Methods

In this section, we describe the two baseline resource combination methods against which we compare our model interpolation method.

4.1 Treebank Concatenation

The treebank concatenation method of McDonald et al. (2011) proceeds as follows:

1. Concatenate all src treebanks.
2. Train a delex parser on the resulting treebank.
3. Apply the parser to the tgt text.

4.2 Parse Tree Combination

The parse tree combination method is defined by Rosa and Žabokrtský (2015) in the following way:

1. Train a delex parser on each src treebank.
2. Apply each of the parsers to the tgt sentence, obtaining a set of parse trees.
3. Construct a weighted directed graph over tgt sentence tokens, with each edge assigned a score equal to the number of parse trees that contain this edge. (i.e., each parse tree contributes by 0 or 1 to the edge score). In the weighted variant, the contribution of each src parse tree is multiplied by $KL^{-4}_{cpos^3}(tgt, src)$.
4. Find the maximum spanning tree over the graph with the Chu-Liu-Edmonds algorithm.

Note that if each src parse tree contributed with a (normalized) score of the edge as assigned by its model rather than with a 0 or 1, this method would be equivalent to the model interpolation method.

5 Dataset

We carry out all experiments using HamleDT 2.0 (Rosa et al., 2014), a collection of 30 treebanks converted into Universal Stanford Dependencies (de Marneffe et al., 2014). We use gold-standard UPOS tags in all experiments; while this is not fully realistic in the setting of under-resourced languages, there exist high-performance semi-supervised taggers that could be used instead of gold tags (Das and Petrov, 2011; Agić et al., 2015), which we plan to evaluate in future. We use the treebank training sections for parser training and $KL^{-4}_{cpos^3}$ computation, and the test sections for evaluation. We used 12 of the treebanks as a development set to select the model normalization method to avoid overfitting it to the dataset.[4]

6 Evaluation

Table 1 contains the results of our model interpolation methods, as well as the baseline methods. For each tgt language, all remaining 29 src treebanks were used for parser training. We base our evaluation on comparing absolute differences in UAS on the whole set of 30 languages as targets.[5]

The performance of the weighted model interpolation is comparable to the weighted tree combination – the difference in average UAS of the methods is lower than 0.1%, with model interpolation achieving a higher UAS than the tree combination for 16 of the 30 tgt languages. This shows

[3] $f_{src}(cpos^3) := \frac{1}{N}$ if the src corpus does not contain the given trigram (N is the number of tokens in the corpus).

[4] The development set was chosen to contain multiple members of several language families (Uralic, Romance), as well as a very solitary language (Japanese), etc.; also, we cared that both smaller and larger treebanks are represented.

[5] The results of our method are generally better on the test set than on the development set, suggesting that no overfitting happened.

Target language	Unweighted			Weighted	
	Conc	Tree	Inter	Tree	Inter
Bengali	61.0	63.2	**67.1**	66.7	**66.9**
Czech	**60.5**	60.4	57.5	**65.8**	65.2
Danish	**56.2**	54.4	48.9	**50.3**	49.5
German	12.6	**27.6**	18.2	56.8	**61.6**
English	12.3	**21.1**	16.2	42.6	**48.6**
Basque	**41.2**	40.8	39.5	30.6	**34.9**
Anc. Greek	43.2	**44.7**	41.4	42.6	**44.0**
Latin	38.1	**40.3**	39.7	**39.7**	39.5
Dutch	55.0	**56.2**	54.2	58.7	**59.4**
Portuguese	62.8	**67.2**	62.8	62.7	**63.7**
Romanian	44.2	**51.2**	48.6	50.0	**50.3**
Russian	55.5	**57.8**	53.3	**57.2**	56.3
Slovak	52.2	**59.6**	55.7	58.4	**60.6**
Slovenian	45.9	**47.1**	42.8	**53.9**	49.6
Swedish	45.4	**52.3**	49.4	**50.8**	50.4
Tamil	27.9	**28.0**	27.6	**40.0**	37.3
Telugu	67.8	68.7	**72.9**	**77.4**	**77.4**
Turkish	18.8	23.2	**25.3**	**41.1**	34.8
Average	44.5	**48.0**	45.6	52.5	**52.8**
Std. dev.	16.9	**15.0**	16.0	**11.8**	12.0
Arabic	**37.0**	35.3	30.7	**41.3**	34.6
Bulgarian	64.4	**66.0**	60.3	67.4	**68.5**
Catalan	56.3	**61.5**	58.5	72.4	**72.4**
Greek	**63.1**	62.3	59.6	63.8	**64.1**
Spanish	59.9	**64.3**	60.4	72.7	**72.7**
Estonian	67.5	**70.5**	67.4	**72.0**	71.7
Persian	30.9	**32.5**	29.5	**33.3**	28.6
Finnish	**41.9**	41.7	41.5	**47.1**	44.7
Hindi	24.1	24.6	**26.2**	27.2	**32.7**
Hungarian	55.1	56.5	**57.4**	51.2	**53.0**
Italian	52.5	**59.5**	56.0	59.6	**60.1**
Japanese	**29.2**	28.8	27.2	**34.1**	33.0
Average	48.5	**50.3**	47.9	**53.5**	53.0
Std. dev.	**15.2**	16.5	15.6	**16.7**	17.4
Average	46.1	**48.9**	46.5	**52.9**	52.9
Std. dev.	16.1	**15.4**	15.6	**13.7**	14.1

Table 1: UAS on test *tgt* treebanks (upper part of table) and development *tgt* treebanks (lower part).
Conc = Treebank concatenation
Tree = Parse tree combination
Inter = Model interpolation
Average = Average UAS (on test/development/all)
Std. dev. = Standard sample deviation of UAS, serving as an indication of robustness of the method

that weighted model interpolation is a good alternative to weighted tree combination.

In the unweighted setting, the situation is quite different, with model interpolation scoring much lower than tree combination (-2.4%), and only slightly higher than treebank concatenation (+0.4%) on average. This suggests that, contrary to our original intuition, edge scores assigned by the *src* models are not a good proxy for parser confidence, not even when appropriately normalized.[6] Furthermore, the weighted methods generally outperform the unweighted ones (by +4.0% for tree combination and by +6.4% for model interpolation on average), which suggests, among other, that the *src*-*tgt* language similarity is much more important than the exact values of *src* edge scores for resource combination in delex transfer.

[6]The same tendency was observed across all normalization methods evaluated on the development set.

7 Conclusion

We presented trained parser model interpolation as an alternative method for multi-source crosslingual delexicalized dependency parser transfer. Evaluation on a large collection of treebanks showed that in a setting where the source languages are weighted by their similarity to the target language, model interpolation performs comparably to the parse tree combination approach. Moreover, model interpolation is significantly less computationally demanding than the tree combination when parsing the target text, as the interpolation can be efficiently performed beforehand, thus only requiring to invoke a single parser at runtime, while in the tree combination approach, each source parser has to be invoked individually.

In the unweighted setting, model interpolation consistently performed much worse than tree combination, which we find rather surprising, and we therefore plan to further investigate this in future. Still, the weighted methods generally outperformed the unweighted ones, and as the language similarity measure that we used only requires the source treebanks and a target POS-tagged text, i.e. exactly the resources that are required even for the unweighted delex transfer methods, there is little reason not to employ the weighting. Therefore, the low performance of the unweighted model interpolation is of less importance than its high performance in the weighted setting.

In this work, we only used the unlabelled MSTParser for all experiments. We believe that extending our method to other parsers constitutes an interesting path for future research.

Acknowledgments

This research was supported by the grants GAUK 1572314, SVV 260 224 and FP7-ICT-2013-10-610516 (QTLeap). This 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 LM2010013).

References

Željko Agić, Dirk Hovy, and Anders Søgaard. 2015. If all you have is a bit of the bible: Learning POS taggers for truly low-resource languages. In *Proceedings of ACL-IJCNLP*. Hrvatska znanstvena bibliografija i MZOS-Svibor.

Giovanni Cavallanti, Nicolo Cesa-Bianchi, and Claudio Gentile. 2010. Linear algorithms for online multitask classification. *The Journal of Machine Learning Research*, 11:2901–2934.

Yoeng-Jin Chu and Tseng-Hong Liu. 1965. On shortest arborescence of a directed graph. *Scientia Sinica*, 14(10):1396.

Shay B. Cohen, Dipanjan Das, and Noah A. Smith. 2011. Unsupervised structure prediction with non-parallel multilingual guidance. In *Proceedings of EMNLP*, pages 50–61, Stroudsburg, PA, USA. ACL.

Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *The Journal of Machine Learning Research*, 3:951–991.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of ACL-HLT*, pages 600–609. ACL.

Marie-Catherine de Marneffe, Natalia Silveira, Timothy Dozat, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford dependencies: A cross-linguistic typology. In *Proceedings of LREC'14*, Reykjavík, Iceland. ELRA.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the National Bureau of Standards B*, 71(4):233–240.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, pages 79–86.

David McClosky, Eugene Charniak, and Mark Johnson. 2010. Automatic domain adaptation for parsing. In *Proceedings of HLT-NAACL*, pages 28–36. ACL.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005a. Online large-margin training of dependency parsers. In *Proceedings of ACL*, pages 91–98. ACL.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005b. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of HLT-EMNLP*, pages 523–530. ACL.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of EMNLP*, pages 62–72, Stroudsburg, PA, USA. ACL.

Tahira Naseem, Regina Barzilay, and Amir Globerson. 2012. Selective sharing for multilingual dependency parsing. In *Proceedings of ACL*, pages 629–637, Stroudsburg, PA, USA. ACL.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proc. of LREC-2012*, pages 2089–2096, Istanbul, Turkey. ELRA.

Rudolf Rosa and Zdeněk Žabokrtský. 2015. KL_{cpos^3} – a language similarity measure for delexicalized parser transfer. In *Proceedings of ACL-IJCNL*, Stroudsburg, PA, USA. ACL.

Rudolf Rosa, Jan Mašek, David Mareček, Martin Popel, Daniel Zeman, and Zdeněk Žabokrtský. 2014. HamleDT 2.0: Thirty dependency treebanks stanfordized. In *Proceedings of LREC 2014*, pages 2334–2341, Reykjavík, Iceland. ELRA.

Rudolf Rosa. 2015a. MSTperl delexicalized parser transfer scripts and configuration files. `http://hdl.handle.net/11234/1-1485`.

Rudolf Rosa. 2015b. MSTperl parser (2015-05-19). `http://hdl.handle.net/11234/1-1480`.

Kenji Sagae and Alon Lavie. 2006. Parser combination by reparsing. In *Proceedings of HLT-NAACL*, pages 129–132. ACL.

Anders Søgaard and Julie Wulff. 2012. An empirical etudy of non-lexical extensions to delexicalized transfer. In *COLING (Posters)*, pages 1181–1190.

Oscar Täckström, Ryan McDonald, and Joakim Nivre. 2013. Target language adaptation of discriminative transfer parsers. In *Proceedings of HLT-NAACL*, pages 1061–1071.

Daniel Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *IJCNLP 2008 Workshop on NLP for Less Privileged Languages*, pages 35–42, Hyderabad, India. Asian Federation of Natural Language Processing, International Institute of Information Technology.

Daniel Zeman, David Mareček, Martin Popel, Loganathan Ramasamy, Jan Štěpánek, Zdeněk Žabokrtský, and Jan Hajič. 2012. HamleDT: To parse or not to parse? In *Proceedings of LREC'12*, Istanbul, Turkey, May. ELRA.

Daniel Zeman. 2008. Reusable tagset conversion using tagset drivers. In *Proceedings of LREC 2008*, pages 213–218, Marrakech, Morocco. ELRA.