Semi-supervised experiments at LORIA for the SPMRL 2014 Shared Task

Christophe Cerisara
LORIA, UMR7503
FRANCE
cerisara@loria.fr

Abstract

This paper describes the LORIA first participation at the SPMRL Shared Task. The focus of this work is on exploring several options to take advantage of the unlabeled data to improve the performances of a baseline dependency parser, which has neither be tuned to the specificities of the shared task nor evaluation languages. The semi-supervised approaches investigated include LDA word classes and super-tags predicted by a linear classifier trained with self-training. None of these options resulted in increased parsing accuracy.

1 Introduction

This paper describes my first participation in the series of SPMRL syntactic parsing shared tasks (Seddah et al., 2014). This is the result of about 3 weeks of personal work on this task, starting from vanilla versions of parsers available on the Web. For the shared task, I provided results on all languages, including Korean (Choi et al., 1994; Choi, 2013), Hungarian (Vincze et al., 2010a; Csendes et al., 2005a), German (Brants et al., 2002; Seeker and Kuhn, 2012), Basque (Aduriz et al., 2003), French (Abeillé et al., 2003), Polish (Świdziński and Woliński, 2010a) and Swedish (Nivre et al., 2006). I also used the shared task Hebrew data set (Sima'an et al., 2001; Tsarfaty, 2013; Goldberg, 2011) and the Arabic data set, originally provided by the LDC (Maamouri et al., 2004), specifically its SPMRL 2014 dependency instance, derived from the Columbia Catib Treebank (Habash and Roth, 2009; Habash et al., 2009).

The objective of this work is to implement and test some basic solutions to exploit a large unlabeled corpus in addition to the standard labeled training dataset used in supervised parsing. The unlabeled data provided in the 2014 SPMRL shared task edition is a very interesting testbed to investigate such approaches.

The proposed system follows the next guidelines:

- The parser is not tuned to any language; this implies to consider every CoNLL input file as originating from an unknown language, and thus not using any specific knowledge, nor POS-tagger, tokenizer, segmenter. This also implies to only use the baseline set of features as configured in the baseline parser, and not to tune this basic set of features.
- Exploit the unlabeled corpus provided in each language to explore a few options to try and improve parsing performances.

2 Baseline system

The system I used to report the results on the test set is based on the MATE parser version 3.61 (Bohnet, 2010). I also used the Malt parser version 1.8 (Nivre et al., 2007) to explore different options for exploiting the unlabeled corpus, because the Malt parser is faster than MATE although its accuracy is not as good. As mentioned in the introduction, I used the default set of features for both parsers. Segmentation,

tokenization, lemmas and part-of-speech (POS) tags are provided within the CoNLL file and used as is in the following experiments. Because of the conversion from the CoNLL06 to the CoNLL09 formats, the choice between one of the two (broad and detailed) POS-tag columns has to be made, and following the suggestions from (Björkelund et al., 2013), I used the broad POS-tags (column 4) for BASQUE and KOREAN, and the fine POS-tags (column 5) for all other languages. Furthermore, because of a very large number (416) of different dependency labels in HUNGARIAN, which caused memory failures with the MATE parser, I only kept the 50 most frequent labels in the full training corpus. For the other labels, I split their name based on the hyphen symbols and mapped them to their first sub-name. Hence, ROOT-VAN-ATT-VAN-PRED becomes ROOT, and COORD-ELL-COORD-ELL-COORD-ELL-ATT becomes COORD.

The respective performance of these baseline systems is given in Table 1.

Language	**MATE**	**Malt**
BASQUE	83.1	77.1
FRENCH	84.2	78.0
GERMAN	91.0	82.4
HEBREW	75.8	71.1
HUNGARIAN	82.9	75.3
KOREAN	82.4	82.8
POLISH	85.9	79.5
SWEDISH	76.7	70.5
ARABIC	84.7	81.3

Table 1: Baseline LAS performances on the DEV corpus (full training, pred)

3 Final submission system

3.1 System description

This Section describes the system that has been used to submit the official results to the SPMRL2014 evaluation campaign. This system is the baseline MATE parser described in Section 2 with one additional feature, which encodes word classes computed on the first 10 million words extracted from each unlabeled corpus.

These word classes are computed in a similar way as in (Chrupala, 2011), but using a syntactic context instead of a left-right context. So the procedure to compute these word classes is as follows:

1. Extract the first sentences from the unlabeled corpus, up to 10 million words;
2. Parse these sentences with the baseline Malt parser described in Section 2;
3. Extract the set $\mathcal{H}(w)$ of all governors (word forms) for each word w;
4. Build one "document" per word in the vocabulary, where each document is composed of the set of tokens $\mathcal{H}(w)$, i.e., the set of all governors that have been observed for this word;
5. Train a standard Latent Dirichlet Allocation (LDA) topic model on this set of documents, with parameters arbitrarily set to 100 topics and 1000 iterations. The software used to train the LDA model is Mallet (McCallum, 2002).
6. Compute for each document, i.e. each word in the vocabulary, the most probable topic

$$\hat{t} = \arg\max_t P(t|w)$$

and assign the word w to the word class $\hat{t}$.

7. Add this word class into the FEATS column in the CoNLL train and test files, and retrain the MATE parser with this additional feature.

A number of previous works in the semi-supervised litterature suggest word classes as a potentially powerful approach to exploit a large unlabeled corpus within an otherwise standard supervised existing parser (Koo et al., 2008), which motivates the approach chosen here. Classically, Brown clustering is used to compute the word classes, but (Chrupala, 2011) recently showed that LDA clustering may provide as good clusters as with Brown clustering, but at a lower computational cost. In his work, G. Chrupala exploits a traditional left-right context in order to compare fairly with Brown clustering. In the proposed system, I rather opted for a syntactic context, which may be more relevant for the parsing task.

3.2 System results and discussion

The results obtained on the DEV set with this final system are reported in Table 2.

Language	DEV baseline	DEV with LDA word class	Official TEST results
BASQUE	83.1	**83.1**	83.2
FRENCH	84.2	**84.3**	84.1
GERMAN	91.0	**91.1**	88.0
HEBREW	**75.8**	75.7	75.6
HUNGARIAN	**82.9**	82.6	83.8
KOREAN	82.4	**82.5**	81.9
POLISH	**85.9**	85.4	80.1
SWEDISH	76.7	**77.0**	80.0
ARABIC	84.7	**84.8**	84.0

Table 2: Baseline vs. final LAS performances on the DEV corpus (full training, pred) with LDA syntactic word classes. The final column shows the official results on the TEST corpus with LDA word classes.

We can observe that the impact of LDA word classes is null, at least with this configuration. It is however not possible to draw any definitive conclusion with regard to the usefulness of LDA clustering in this context, because of the limited set of experiments realized in this work. In particular, there are more alternative options than the chosen one that should have been tested, would I had enough time to do so: isn't the part of the unlabeled corpus used too small ? Shouldn't rare words or stop words be filtered-out ? Shouldn't low-confidence dependency arcs be removed ? Shouldn't a wider syntactic context, with grand-parents and children, be used ? etc.

Additional experiments are described in the next Section.

4 Additional experiments

4.1 With LDA classes

Here is a short list of some additional experiements that I have realized with LDA classes. I briefly summarize them to better explain the context in which the final results have been submitted:

- **convergence of LDA:** Because the Mallet implementation of LDA relies on random sampling, two runs of the same system with a fixed number of iterations may produce different results, because Gibbs sampling may not have converged precisely enough. I have made a few quick tests to estimate how much iterations would be required to limit this effect, and obtained a variability in LAS of about $\pm 0.5\%$ with 1000 iterations, $\pm 0.3\%$ with 2000 iterations and $\pm 0.1\%$ with 5000 iterations.

- **linear vs. syntactic context:** I have also used the classical left-right context to compute LDA word classes instead of the proposed syntactic context, just like in (Chrupala, 2011), but obtained similar results.

- **richer LDA features:** I have tested with two features: the two best topics per word, instead of just the best topic given by LDA, but again with similar results.
- **back-off for rare words:** I have tested by replacing all word forms that occur less than 100 times by their POS-tag to train LDA, but this does not give really more convincing results.
- **Brown classes:** In order to compare LDA vs. Brown classes, I have also tried Liang's implementation of Brown clustering (Liang, 2005), but only obtained slight degradation of LAS. It also ran into complexity issues, as shown in Table 3, which shows the required computation time of Brown clustering on my machine.

# unlabeled sentences	# topics	Required computation time in seconds
1k	100	6
1k	200	14
1k	400	32
10k	100	49
100k	100	264
1000k	100	1265

Table 3: Computation time required to compute Brown clusters on a standard PC

4.2 With self-training

Another traditional way to exploit unlabeled data is to perform self-training. I have tested a basic self-training scheme, but it impacts negatively the performances of the MATE parser, which is consistent with previous published works that also concluded that it is very hard to obtain reliable improvement with self-training for parsing. I have also trained a linear classifier to discriminate between incorrectly and correctly predicted dependency arc, in order to use this classifier as a confidence measure to filter-out the most likely erroneous sentences from the unlabeled corpus, but this did not help much.

4.3 With predicted label

Because self-training of a structured model may be too difficult, I rather tried using an unstructured classifier on which it may be easier to apply self-training. This classifier outputs a new feature that is then added into the CoNLL training and test files of the parser. The feature used here is the predicted label of the dependency arc. Two-stage parsers traditionally predict first the unlabeled structure, and then the labels on this arc. Conversely, the system proposed here consists in first predicting the label without knowing the structure, and then predicting the structure given the label. This approach is motivated by a small experiment, which uses oracle labels as input features into the MATE parser. Then, the resulting LAS increases from 83% up to 94% for BASQUE, and from 86% up to 95% for POLISH.

This proves that such a feature may be very useful, although it is obviously very challenging to predict this label without knowing the structure. Nevertheless, the rationale behind this proposal is that it may be easier to apply self-training on the first unstructured classifier and thus benefit from the unlabeled corpus at least in this first unstructured stage.

The proposed linear model to predict dependency labels is trained with the LIBLINEAR software (Fan et al., 2008) with the default configuration, and exploits the linear-context of each word as features: 7-gram POS-tags and 3-gram word forms. All the following experiments are realized on the French corpus. The proposed linear classifier obtains a label predicting accuracy of 77.6%. This linear model is then trained with 10-fold cross-validation on the French training corpus, and its predicted labels are added into the training and test CoNLL files. The Malt parser is then retrained on this new corpus and its predicted LAS slightly increases from 78.0% up to 78.7%.

In order to improve its accuracy, the unstructured linear classifier is self-trained on part of the French unlabeled corpus as follows:

1. Predict the label on the first million sentences extracted from the unlabeled corpus.
2. Removes all low-confidence words from this predicted corpus. Low-confidence words correspond to words for which the score margin, i.e., the score of the winning label minus the score of the second best label, is smaller than 5.
3. Add the remaining words into the training set of the linear classifier and retrain it.
4. A single iteration was performed in this experiment.

This self-training set-up indeed improves the accuracy of the linear classifier up to 83%, but this increase does not translate into a better Malt parsing model, which LAS remains at 78.7%. I have also observed that the use of the confidence measure is very important, because adding too many unreliable instances into the training corpus dramatically reduces the accuracy of the linear classifier.

The conclusions that can be drawn from these preliminary experiments are thus that:

- Predicting first the label and then the dependency structure may be useful.
- It is indeed easier to apply self-training on an unstructured classifer than directly on the parser.
- Very small improvement can be observed on the Malt parser LAS, but not on the MATE parser.

Additional thoroughly experiments would be interesting to realize to better explore this research track.

5 Conclusion

This paper describes some of the experiments realized at LORIA to try and exploit unlabeled data to improve dependency parsing. Note that the baseline system may not be very accurate, as compared to the competing systems, because I simply used a state-of-the-art parser as it is, and did not tune it in any way to improve its accuracy on any of the target language. I also did not use any specific processing for any language: the proposed system was only focused on trying to exploit the unlabeled corpus in a semi-supervised way, which turned out to be not very effective, at least in the settings explored in this work and in a constrained time frame. Despite these negative results, I believe at least two conclusions can be drawn from these experiments: (i) LDA clustering always performed better than Brown clustering in my experiments, and exploring such topic models may be interesting, not only for unsupervised parsing, as it has already been done many times, but rather for semi-supervised parsing with an already large labeled training corpus available. (ii) Self-training of unstructured classifier being easier to achieve, it may be interesting to pre-process the data with a discriminative classifier that predicts the dependency labels, or some other syntactically-related feature, and thus exploit the unlabeled corpus without inferring any structure.

Acknowledgements

I would like to thank very much the organizers of the SPMRL shared task for having compiled and formated this extremely valuable set of resources for research on parsing. I also express my gratitude to the treebank providers for each language: Arabic (Maamouri et al., 2004; Habash and Roth, 2009; Habash et al., 2009), Basque (Aduriz et al., 2003; Aldezabal et al., 2008), French (Abeillé et al., 2003; Candito et al., 2010), Hebrew (Sima'an et al., 2001; Tsarfaty, 2010; Goldberg, 2011; Tsarfaty, 2013), German (Brants et al., 2002; Seeker and Kuhn, 2012), Hungarian (Csendes et al., 2005b; Vincze et al., 2010b), Korean (Choi et al., 1994; Choi, 2013), Polish (Woliński et al., 2011; Świdziński and Woliński, 2010b; Wróblewska, 2012), and Swedish (Nivre et al., 2006).

I also would like to thank the following institutions for providing the unlabeled data set: The LDC for the Arabic Gigaword, the IXA NLP Research Group and Elhuyar Fundazioa for the Elhuyar Corpus Version, the French CNRTL for the Est Republicain Corpus, the IMS group and the university of Heidelberg for the German Wikidump, the Bar Ilan University for the Hebrew Wikidump, the University of Szeged for their Hungarian Newsire corpus, the KAIST for their Large Scale Korean Corpus, the Polish Institute

of Science for their Wikipedia corpus, and the Språbanken group at the University of Gothenburg for their Parole corpus.

References

Anne Abeillé, Lionel Clément, and François Toussenel. 2003. Building a treebank for french. In Anne Abeillé, editor, *Treebanks*. Kluwer, Dordrecht.

I. Aduriz, M. J. Aranzabe, J. M. Arriola, A. Atutxa, A. Díaz de Ilarraza, A. Garmendia, and M. Oronoz. 2003. Construction of a Basque dependency treebank. In *Proc. TLT*, pages 201–204.

I. Aldezabal, M.J. Aranzabe, A. Diaz de Ilarraza, and K. Fernández. 2008. From dependencies to constituents in the reference corpus for the processing of Basque. In *Procesamiento del Lenguaje Natural, n 41 (2008)*, pages 147–154. XXIV edición del Congreso Anual de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN).

Anders Björkelund, Richárd Farkas, Thomas Müller, and Wolfgang Seeker. 2013. (re) ranking meets morphosyntax: State-of-the-art results from the spmrl 2013 shared task. In *Proc. of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 135–145, October.

B. Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proc. COLING*, Beijing, China.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In Erhard Hinrichs and Kiril Simov, editors, *Proceedings of the First Workshop on Treebanks and Linguistic Theories (TLT 2002)*, pages 24–41, Sozopol, Bulgaria.

Marie Candito, Benoit Crabbé, and Pascal Denis. 2010. Statistical French dependency parsing: Treebank conversion and first results. In *Proceedings of LREC*, Valletta, Malta.

Key-Sun Choi, Young S Han, Young G Han, and Oh W Kwon. 1994. Kaist tree bank project for korean: Present and future development. In *Proceedings of the International Workshop on Sharable Natural Language Resources*, pages 7–14. Citeseer.

Jinho D. Choi. 2013. Preparing korean data for the shared task on parsing morphologically rich languages. In *Proceedings of the EMNLP 2013 Workshop of Statistical Parsing of Moprhologically-Rich Languages (SPMRL 2013)*, Seattle, US.

G. Chrupala. 2011. Efficient induction of probabilistic word classes with lda. In *Proceedings of IJCNLP*.

Dóra Csendes, Janós Csirik, Tibor Gyimóthy, and András Kocsor. 2005a. The Szeged treebank. In Václav Matoušek, Pavel Mautner, and Tomáš Pavelka, editors, *Text, Speech and Dialogue: Proceedings of TSD 2005*. Springer.

Dóra Csendes, János Csirik, Tibor Gyimóthy, and András Kocsor. 2005b. The Szeged treebank. In *Proceedings of the 8th International Conference on Text, Speech and Dialogue (TSD)*, Lecture Notes in Computer Science, pages 123–132, Berlin / Heidelberg. Springer.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Yoav Goldberg. 2011. *Automatic syntactic processing of Modern Hebrew*. Ph.D. thesis, Ben Gurion University of the Negev.

Nizar Habash and Ryan Roth. 2009. Catib: The columbia arabic treebank. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 221–224, Suntec, Singapore, August. Association for Computational Linguistics.

Nizar Habash, Reem Faraj, and Ryan Roth. 2009. Syntactic Annotation in the Columbia Arabic Treebank. In *Proceedings of MEDAR International Conference on Arabic Language Resources and Tools*, Cairo, Egypt.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proc. ACL/HLT*.

P. Liang. 2005. Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*.

Andrew Kachites McCallum. 2002. MALLET: A machine learning for language toolkit. `http://mallet.cs.umass.edu`.

Joakim Nivre, Jens Nilsson, and Johan Hall. 2006. Talbanken05: A Swedish treebank with phrase structure and dependency annotation. In *Proceedings of LREC*, pages 1392–1395, Genoa, Italy.

Joakim Nivre, Jens Hall, Jens Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov, and E. Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

Djamé Seddah, Reut Tsarfaty, Sandra K´ubler, Marie Candito, Jinho Choi, Matthieu Constant, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przepiorkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska, and Eric Villemonte de la Clérgerie. 2014. Overview of the spmrl 2014 shared task on parsing morphologically rich languages. In *Notes of the SPMRL 2014 Shared Task on Parsing Morphologically-Rich Languages*, Dublin, Ireland.

Wolfgang Seeker and Jonas Kuhn. 2012. Making Ellipses Explicit in Dependency Conversion for a German Treebank. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 3132–3139, Istanbul, Turkey. European Language Resources Association (ELRA).

Khalil Sima'an, Alon Itai, Yoad Winter, Alon Altman, and Noa Nativ. 2001. Building a Tree-Bank for Modern Hebrew Text. In *Traitement Automatique des Langues*.

Marek Świdziński and Marcin Woliński. 2010a. Towards a bank of constituent parse trees for Polish. In *Text, Speech and Dialogue: 13th International Conference (TSD)*, Lecture Notes in Artificial Intelligence, pages 197—204, Brno, Czech Republic. Springer.

Marek Świdziński and Marcin Woliński. 2010b. Towards a bank of constituent parse trees for Polish. In *Proceedings of Text, Speech and Dialogue*, pages 197–204, Brno, Czech Republic.

Reut Tsarfaty. 2010. *Relational-Realizational Parsing*. Ph.D. thesis, University of Amsterdam.

Reut Tsarfaty. 2013. *A Unified Morpho-Syntactic Scheme of Stanford Dependencies*. Proceedings of ACL.

Veronika Vincze, Dóra Szauter, Attila Almási, György Móra, Zoltán Alexin, and János Csirik. 2010a. Hungarian dependency treebank. In *LREC*.

Veronika Vincze, Dóra Szauter, Attila Almási, György Móra, Zoltán Alexin, and János Csirik. 2010b. Hungarian Dependency Treebank. In *Proceedings of LREC*, Valletta, Malta.

Marcin Woliński, Katarzyna Głowińska, and Marek Świdziński. 2011. A preliminary version of Składnica—a treebank of Polish. In *Proceedings of the 5th Language & Technology Conference*, pages 299–303, Poznań, Poland.

Alina Wróblewska. 2012. Polish Dependency Bank. *Linguistic Issues in Language Technology*, 7(1):1–15.