Experiments on Semi-supervised Dependency Parsing of a Morphologically Rich Language

Aitziber Atutxa, Nerea Ezeiza, Iakes Goenaga, Koldo Gojenola
University of the Basque Country UPV/EHU / IXA NLP Group
aitziber.atucha@ehu.eus, n.ezeiza@ehu.eus,
iakes.goenaga@ehu.eus, koldo.gojenola@ehu.eus

Abstract

This paper[1] presents a set of preliminary experiments that have the aim of improving dependency parsing of Basque by using a semi-supervised technique. Our approach will make use of large unannotated corpora (over 140M word forms). We will investigate the use of information induced from a large raw corpus as well as an automatically parsed version. The first results show encouraging improvements with respect to previous work. We have also made a first attempt to induce metafeatures with the objective of reducing data sparseness.

1 Introduction

In last year's Shared Task on Dependency Parsing of morphologically rich languages (Seddah et al., 2014), results of the different teams showed no significant or little improvements in parsing using unsupervised data taken from big corpora, in the form of Brown clusters (Koo et al., 2008) or word embeddings (Mikolov et al., 2013). For example, only the second system obtained small improvements for two languages (Basque and Swedish) making use of Brown clusters and word embeddings (Goenaga et al., 2014), although the results improved considerably when making a voted ensemble parser using unsupervised data. Andreas and Klein's (2014) experiments seem to point out on the same direction as well. They claim that extra information from embeddings appears to make little or no difference to a constituency parser of English with adequate training data. These results contrast with the initial successful experiments of (Koo et al., 2008), where Brown clusters considerably increased the performance of two parsers for English and Czech using a simple technique, directly substituting wordforms and parts of speech by clusters. Or with posterior works on a constituency parser for French (Candito and Seddah, 2010) and a dependency parser for German (Bohnet and Nivre, 2012) where improvements in F-score were reported. There can be several reasons for this divergence between the first successful results and the ones in the SPMRL 2014 Shared Task. One of the reasons could be the size of the corpus used for inducing unsupervised knowledge, or the parser used, or even the intrinsic nature of morphologically rich languages, where the high number of word forms from each lemma can be a challenge. For example German shows only four grammatical cases, while languages like Basque show more than fifteen different case markers, and being agglutinative each case mark has also several definitiveness and plurality variations increasing sparsity. Additionally, another reason could be that the techniques applied so far produce a big increase in the number of very sparse features, that cannot be managed by the used parsers.

With this objective, we present a set of preliminary experiments that will try to improve dependency parsing of Basque by using semi-supervised techniques. On one hand, we will study the effect of increasing the size of the unsupervised corpora and, on the other hand, we will also try to reduce the number of features.

In the rest of the paper, after presenting related work in Section 2, Section 3 describes the experimental setting of our work. Section 4 discusses the results that have been obtained, while Section 5 presents the main conclusions and some avenues for future work.

2 Related work

Supervised parsing, as any other NLP supervised task, suffers from a fundamental data bottleneck problem. The corresponding model parameters for

[1] Authors appear in alphabetical order

Proceedings of the 6th Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2015), pages 7–13,
Bilbao, Spain, July 23rd 2015.

rare or unseen words in the labeled treebank data are poorly estimated resulting in an accuracy decay. Although many works have tried to incorporate different types of information from treebank external resources, this research area has not still reached a definitive answer to the problem of how to effectively add knowledge to improve parser performance.

There are several approaches to include treebank external information, such as inducing unsupervised word representations in the form of word-clusters (Koo et al., 2008; Candito and Seddah, 2010; Haffari et al., 2011; Täckström et al., 2012; Bengoetxea et al., 2014) or word-embeddings (Andreas and Klein, 2014; Bansal et al., 2014). Another approaches consist of incorporating information from existing lexico-semantic databases such as WordNet (Agirre et al., 2008; Bengoetxea et al., 2014) or self-training (McClosky et al., 2006).

Koo et al. (2008) pursued one of the first experiments using word-clusters induced from unlabel data in parsing. Clusters were built by means of the well known Brown algorithm (Brown et al., 1992), a hierarchical agglomerative clustering algorithm which clusters words selecting at each step the pair of clusters that maximizes the average mutual information of bigrams of the current clustering. These clusters allowed them to use a new range of cluster-based feature set in addition to the baseline features. Their experiments were carried out on two different language-corpora; the Penn treebank and Prague Dependency treebank, using constituency parser and dependency parser respectively. They report consistently positive results for both languages (a 1.14% accuracy gain for English and 1% for Czech).

Along the same lines, Agirre et al. (2014) study the impact of including unsupervised semantic information in several state-of-the-art dependency parsers (MST, MALT and ZPar) and their combinations. Agirre et al. (2014) applied semantic information in the form of synsets and semantic files from a external resource, namely WordNet2.1, in addition to Brown clustering information. Their results show small improvements for the single parser experiments, the best improvement being +1.12 LAS for the MST parser using Brown clusters.

Chen et al. (2013) attempt to tackle the sparsity problem by using large amounts of unannotated data as well. They propose to use what they call *metafeatures* by transforming base features to a higher-level space. Basically the idea is to group the base features according to their frequencies, so that each group relates to a metafeature. For that purpose, a large size raw corpus is parsed and frequencies of base features were collected. There are basically four types of metafeatures for each base feature: H, M, L, O (High, Medium, Low and Others respectively). As for the base features, they consider different types, ranging from first-order features, head-dependent related features, and second-order linear or hierarchical ones. Thus sparse low-frequency base features in the training set can result in a metafeature. For example, when a standard first-order feature template named *Head-w-Child-w* will produce features like *Head-ate-Child-chicken* from the training corpus, their parser can obtain an abstract metafeature *Head-w-Child-w-High* if the pair *ate-chicken* were found with high frequency in the automatically parsed large corpus. In the evaluation they show that their system outperforms the results obtained by several other semi-supervised systems on Chinese and English (among them Koo et al. (2008)).

From another point of view, in the last year there has been an increasing interest in using word embeddings to solve many NLP tasks, and parsing is not an exception. Andreas and Klein (2014) investigate this venue as a way of exploiting unlabelled data to enhance the results of a constituency parser (Petrov and Klein, 2007). They focused on the main potential benefits from including word embeddings, which can be useful to help assigning probabilities to unseen words. So whenever an unseen word arose it was replaced by its closest embedding neighbor. They also tried augmenting the statistics of related words, ensuring that similarly-embedded words are preferentially assigned the same preterminal tag. They tried to capture these ideas by removing the morphological features from the parser, retaining indicators on a discretized version of the embeddings. Their results show that though there are subtle improvements in small training sets over the base parser but these improvements disappear as the size of the training set increases.

In the SPMRL 2014 Shared Task (Seddah et al., 2014) several systems competed to parse various morphologically rich languages. Several of them tried to exploit unlabeled data but they did not ob-

tain significant improvements. In fact, the best system (Björkelund et al., 2014) did not make use of such information, though they address this issue in their conclusions, explaining that they made an attempt to include Brown cluster features in the constituency reranker they use, but they had no success exploiting the unlabeled data.

3 Experimental framework

In this section we will first present the parsers used in the experiments, followed by a brief description of the treebank and the corpora we have used. Finally, we explain the main trials we have made with the aim of incorporating unsupervised data into the parsing process.

3.1 Parsers

We have made use of MaltParser (Nivre et al., 2007), MST (McDonald et al., 2005; McDonald et al., 2006) and Mate (Bohnet, 2010), three dependency parsers representing the dominant approaches in data-driven dependency parsing, and that have been successfully applied to typologically different languages and treebanks.

MaltParser is a representative of local, greedy, transition-based dependency parsing models, where the parser obtains deterministically a dependency tree in a single pass over the input using two data structures: a stack of partially analyzed items and the remaining input sequence. We will use one of its latest versions (MaltParser version 1.7). To fine-tune Maltparser we have used MaltOptimizer (Ballesteros and Nivre, 2012a; Ballesteros and Nivre, 2012b). This tool is an interactive system that first performs an analysis of the training set in order to select a suitable starting point for optimization and then guides the user through the optimization of parsing algorithm, feature model, and learning algorithm. Empirical evaluation on data from the CoNLL 2006 and 2007 shared tasks on dependency parsing shows that MaltOptimizer consistently improves over the baseline of default settings and sometimes even surpasses the result of manual optimization.

MST[2] represents global, exhaustive graph-based parsing (McDonald et al., 2005; McDonald et al., 2006) that finds the highest scoring directed spanning tree in a graph. The learning procedure is global and, contrary to greedy algorithms, which make a series of local decisions, the parsing algorithm looks for the best overall tree. The system can be trained using first or second order models. We modified the system in order to add semantic features, combining them with wordforms and POS tags, on the parent and child nodes of each arc.

The Mate parser (Bohnet, 2010) is a development of the algorithms described in (Carreras, 2007; Johansson and Nugues, 2008). In particular, this parser exploits a hash kernel, a new parallel parsing and feature extraction algorithm that improves accuracy as well as parsing speed (Bohnet, 2010).

3.2 Data

We will make use of the dependency treebank presented at the SPMRL 2014 Shared Task on Dependency Parsing of morphologically rich languages (Seddah et al., 2014) together with its associated unannotated corpus of 140 million words.

3.3 Adding external knowledge to the parser

In a previous experiment (Goenaga et al., 2014) pursued in the context of the SPMRL 2014 Shared Task (Seddah et al., 2014), incorporating information from unlabeled data did not have any positive impact over the results of single parsers. As the unlabelled corpus used at that time had a size of 40M word forms, one could argue that lack of improvement followed from the small size of the unlabeled data. Having this in mind, in this work we use a significantly bigger unlabeled corpus (140M word forms). Another relevant difference between present and previous work is the decision of using purely raw text with no previous lemmatization and morphological disambiguation. On one hand, lemmatization allows to generalize alleviating data sparseness. On the other hand, morphological information is crucial in several syntactic phenomena, so this information has to be encoded somehow. One way to do it is to separate lemmas and suffixes as if they were two different word forms. Indeed this was the way pursued in (Goenaga et al., 2014). But Brown clusters are calculated over bigrams and therefore relevant information as argument-verb subcategorization information was not captured. Word embeddings (Mikolov et al., 2013) are calculated over wider local context than bigrams, so we wanted to evaluate whether increasing the size of the corpus, word embeddings were leading to any improvement at

[2] http://mstparser.sourceforge.net

all. We have created word clusters over the embeddings applying the cosine similarity using the K-means algorithm.

In order to create Brown Clusters we applied the implementation by (Liang, 2005), setting the number of the clusters in 800 (c=800). As for the embeddings, we used Mikolov's word2vec tool (Mikolov et al., 2013) to create 800 clusters. For building the word-embeddings we have employed the Continuous Bag of Words algorithm (CBOW).

In line with the work of Chen et al. (2013) we will experiment the use of metafeatures taken from an automatically parsed version of the unannotated corpus. They use a set of metafeatures that are based directly on wordforms (e.g., they generate a first order feature named "Head-w-Child-w-H" when they find a head-child candidate pair of word forms such as "ate-chicken" with high frequency of occurrence in the unsupervised data). Our approach differs from theirs in that we will pursue a generalization of this idea by using metafeatures that are based on the semantic groups induced by Brown clusters or word embeddings. For example, the metafeature "Head-c-Child-c-H" (represented by the clusters corresponding to head and child, respectively) could be obtained from the pair "ate-duck" assuming that duck and chicken belong to the same cluster, and that "ate-chicken" appears with high frequency. This way, even when the exact pair of word forms "ate-duck" did not appear in the training corpus, the unsupervised corpus could provide evidence about this pair, assuming that *duck* and *chicken* belong to the same semantic cluster and that *ate-chicken* appears with high frequency. Due to time constraints, we only were able to test these features in MST, because the metafeatures cannot be directly introduced into a standard parser as input features, but they must be generated at parsing time.

4 Results

Table 1 presents the results of the experiments. We can see that Brown clusters give considerable improvements of 0.56, 0.83 and 0.74 for Malt, MST and Mate, respectively, over the baselines. Looking at the use of clusters based on word embeddings, we see that there is a decrease in MaltParser (-0.33), a small increase for MST (+0.34) and a higher one for Mate (+0.64). Overall, the results show that, while Brown clusters help improve the scores in all the parsers, word embeddings show a more heterogeneous behavior, being only effective for graph-based parsers.

We conducted a detailed analysis over the results of the two best performing parsers, MST and Mate, focusing on some of the most frequent dependency relations. The goal was to find out to what extent the results were homogeneous, that is, whether the best combination (Mate+BC) overcomes the rest on any dependency or if using Brown clusters is consistently better for any relation. Table 2 shows significant divergences in F-measure values among the different combinations over the selected dependencies. While Mate+BC is the best combination tagging *ROOT* dependency relations with a difference of 1.05 over the second one, and 2.44 over the worst one, this is not true for the rest of the dependencies. For *lot* (coordination), *ccomp_obj* (clausal object complement) and *lotat* (discourse related coordination) dependencies, Mate+SC shows the best F-measure value overcoming the second best in 3.92, 0.49 and 1.41 respectively. The MST+BC combination displays the best results for *ncsubj* and *ncmod* relations. These results reveal that the combinations used are complementary and therefore merging them using an ensemble system should boost the overall performance as preliminarily explored by Goenaga et al. (2014) and Bansal et al. (2014).

It is important to notice that metafeatures did not obtain the best results for any of the analyzed dependencies. As explained in Section 2, there are several ways of obtaining metafeatures, for example, by combining first or second order features. The work presented here is preliminary in that only first order features have been used to generate the metafeatures. We envisage trying to get metafeatures from second order features and combinations of first and second order as well.

5 Conclusions

We have presented several experiments trying to make use of unsupervised learning from a big unannotated corpus on a morphologically rich language. Our experiments show that significant improvements have been obtained for both Brown clusters and clusters based on word embeddings. We have also made a first try to using metafeatures based on these clusters, with a slight but not significant increase.

We think that there are several avenues for future work:

	MaltOptimizer	MST	Mate
Baseline	%80.0	%82.69	%83.00
+ BC (all bits)	%80.56 (+0.56)	%83.52 (+0.83)	%83.74 (+0.74)
+ SC	%79.67 (-0.33)	%83.06 (+0.37)	%83.64 (+0.64)
+ Metafeats		%82.81 (+0.12)	

Table 1: Results (LAS) of different experiments on the test set (BC = Brown clusters, SC = similarity clusters based on word embeddings).

	MST+MF	MST+BC	MST+SC	Mate+BC	Mate+SC
ROOT	87.96	88.29	88.06	**90.40**	89.35
lot	76.69	77.07	77.06	77.07	**80.99**
ccomp_obj	73.59	74.93	76.74	74.01	**77.23**
lotat	86.88	86.07	86.53	87.39	**88.80**
ncsubj	69.52	**71.96**	71.19	70.02	70.02
ncmod	82.08	**82.87**	81.57	82.22	81.41

Table 2: Results (F-m) of different dependency relations on the test set (BC = Brown clusters, SC = similarity clusters based on word embeddings, MF = metafeatures).

- Inducing knowledge based on lemmas versus wordforms. Although we have used word forms with encouraging results, morphologically rich languages share the fact that using lemmas can diminish the sparsity of the data and, in this respect, separating lemmas and morphemes seems an aspect that should be evaluated. On the other hand, some of the used tools (e.g. the one we have used for obtaining Brown clusters was based on bigrams) would need an adaptation.

- Using an ensemble system. As mentioned in Section 4, the results obtained by the different combinations tested in this work are complementary with respect to the F-measure obtained over each dependency relation, therefore we plan to use a blender to ameliorate the overall results.

- Metafeatures. We have performed some initial experiments on using metafeatures based on automatically induced data. For the near future we plan to extend these tests following several research paths:

  – At the moment, we only have made use of a very reduced subset of the features employed by Chen et al. (2013), with a slight but not significant increase. We plan to use the full set of metafeatures to see whether the results are improved. Additionally, we also plan to use the same metafeatures of Chen et al. (2013), that is, based on word forms instead of automatically induced clusters, as this will also help to see if the results for English and Chinese also apply to morphologically rich languages.

  – Due to time constrains, we only could apply the new set of metafeatures to MST. As a logical continuation, we also think of performing a similar experiment with Mate.

References

Eneko Agirre, Timothy Baldwin, and David Martinez. 2008. Improving parsing and PP attachment performance with sense information. In *Proceedings of ACL-08: HLT*, pages 317–325, Columbus, Ohio, June. Association for Computational Linguistics.

Eneko Agirre, Kepa Bengoetxea, Joakim Nivre, Yue Zhang, and Koldo Gojenola. 2014. On wordnet semantic classes and dependency parsing. In *Proceedings of the 52th Annual Meeting of the Association of Computational Linguistics*, pages 649–655, Baltimore (Maryland) - (USA), June. Association for Computational Linguistics.

Jacob Andreas and Dan Klein. 2014. How much do word embeddings encode about syntax? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 822–827, Baltimore, Maryland, June. Association for Computational Linguistics.

Miguel Ballesteros and Joakim Nivre. 2012a. Maltoptimizer: A system for maltparser optimization. In *LREC*, pages 2757–2763.

Miguel Ballesteros and Joakim Nivre. 2012b. Maltoptimizer: an optimization tool for maltparser. In *Proceedings of the Demonstrations at the 13th Conference of the European Chaptr of the Association for Computational Linguistics*, pages 58–62. Association for Computational Linguistics.

Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2014. Tailoring continuous word representations for dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 809–815.

Kepa Bengoetxea, Eneko Agirre, Joakim Nivre, Yue Zhang, and Koldo Gojenola. 2014. On wordnet semantic classes and dependency parsing. pages 649–655. ACL 2014.

Anders Björkelund, Özlem Çetinoğlu, Agnieszka Faleńska, Richárd Farkas, Thomas Mueller, Wolfgang Seeker, and Zsolt Szántó. 2014. Introducing the ims-wrocław-szeged-cis entry at the spmrl 2014 shared task: Reranking and morpho-syntax meet unlabeled data. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, pages 97–102, Dublin, Ireland, August. Dublin City University.

Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea*, pages 1455–1465.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China*, pages 89–97.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479, December.

Marie Candito and Djamé Seddah. 2010. Parsing word clusters. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 76–84, Los Angeles, CA, USA, June. Association for Computational Linguistics.

Wenliang Chen, Min Zhang, and Yue Zhang. 2013. Semi-supervised feature transformation for dependency parsing. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1303–1313, Seattle, Washington, USA, October. Association for Computational Linguistics.

Iakes Goenaga, Koldo Gojenola, and Nerea Ezeiza. 2014. Combining clustering approaches for semi-supervised parsing: the basque team system in the sprml2014 shared task. In *First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages: Shared Task on Statistical Parsing of Morphologically Rich Languages*, Dublin, Ireland, August. Dublin City University.

Gholamreza Haffari, Marzieh Razavi, and Anoop Sarkar. 2011. An ensemble model that combines syntactic and semantic clustering for discriminative dependency parsing. In *he 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA - Short Papers*, pages 710–714.

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

Percy Liang. 2005. Semi-supervised learning for natural language. In *MASTERS THESIS, MIT*.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159, New York City, USA, June. Association for Computational Linguistics.

R. McDonald, K. Crammer, and F. Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of ACL*.

R. McDonald, K. Lerman, and F. Pereira. 2006. Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of CoNLL 2006*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Joakim Nivre, Johan Hall, Jens Nilsson, Chanev A., Glsen Eryiit, Sandra Kbler, Marinov S., and Edwin Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics, Proceedings of the Main Conference*, pages 404–411.

Djamé Seddah, Sandra Kübler, and Reut Tsarfaty. 2014. Introducing the spmrl 2014 shared task on parsing morphologically-rich languages. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, pages 103–109, Dublin, Ireland, August. Dublin City University.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 477–487, Montréal, Canada, June. Association for Computational Linguistics.