Towards an N-version Dependency Parser

Miguel Ballesteros[1], Jesús Herrera[1], Virginia Francisco[1], and Pablo Gervás[2]

[1] Departamento de Ingeniería del Software e Inteligencia Artificial
[2] Instituto de Tecnología del Conocimiento
Universidad Complutense de Madrid
C/ Profesor José García Santesmases, s/n E–28040 Madrid, Spain
{miballes, jesus.herrera, virginia}@fdi.ucm.es, pgervas@sip.ucm.es

Abstract. Maltparser is a contemporary dependency parsing machine learning-based system that shows great accuracy. However 90% for Labelled Attachment Score (LAS) seems to be a *de facto* limit for such kinds of parsers. Since generally such systems can not be modified, previous works have been developed to study what can be done with the training corpora in order to improve parsing accuracy. High level techniques, such as controlling sentences' length or corpora's size, seem useless for these purposes. But low level techniques, based on an in-depth study of the errors produced by the parser at the word level, seem promising. Prospective low level studies suggested the development of n-version parsers. Each one of these n versions should be able to tackle a specific kind of dependency parsing at the word level and the combined action of all them should reach more accurate parsings. In this paper we present an extensive study on the usefulness and the expected limits for n-version parser to improve parsing accuracy. This work has been developed specifically for Spanish using Maltparser.

1 Introduction

In the 10th edition of the Conference of Computational Natural Language Learning (CoNLL) a first shared task on Multilingual Dependency parsing was accomplished [1]. Thirteen different languages including Spanish were involved and parsing performance was studied. In this Shared Task, participants implemented a parsing system that could be trained for all these languages. Maltparser 0.4 is the publicly available software that is contemporary of the system presented by Nivre's group to the CoNLL-X Shared Task, in which Spanish was proposed for parsing and Nivre's group achieved great results.

Dependency parsing machine learning-based systems show exceptional accuracy. However 90% for Labelled Attachment Score (LAS) seems to be a *de facto* limit for such kinds of parsers. Since generally such systems can not be modified, we developed some works to study what can be done with the training corpora in order to improve parsing accuracy. High level techniques, such as controlling sentences' length or corpora's size, seem useless for these purposes. However they appeared useful for the design of systematic processes for building training corpora [2]. Low level techniques, based on an in-depth study of the errors produced by the parser at the word level, seem promising. Prospective low level studies suggested the development of n-version parsers. Each one of these n versions should be able to tackle a specific kind of dependency parsing at the

Petr Sojka, Aleš Horák, Ivan Kopeček and Karel Pala (Eds.): TSD 2010, LNAI 6231, pp. 40–47, 2010.

word level and the combined action of all them should reach more accurate parsings. Since n-version parsers could be a valid tool for improving parsing accuracy, we present in this paper an in-depth study on their usefulness and expected limits, as a continuation of our previous work described in [3].

The paper is organized as follows: Section 2 describes the CoNLL-X Shared Task focusing on Spanish participation. In Section 3 we describe the n-version parsing model developed. In Section 4 we analyze the values obtained both for local accuracy and overall accuracy. Finally, Section 5 shows the conclusions of the presented work and suggests some future work.

2 The CoNLL-X Shared Task

The goal of the CoNLL-X Shared Task [1] was to label dependency structures by means of a fully automatic dependency parser. This task provided a benchmark for evaluating parsers accross 13 languages, one being Spanish. Systems were scored by computing their Labelled Attachment Score (LAS), i.e. the percentage of "scoring" tokens for which the system had predicted the correct head and dependency label [4], their Unlabelled Attachment Score (UAS), i.e. the percentage of "scoring" tokens for which the system had predicted the correct head [5] and their Label Accuracy (LA), i.e. the percentage of "scoring" tokens for which the system had predicted the correct dependency label [6].

The results for Spanish across the 19 participants ranged from 47% to 82.3% LAS, with an average of 73.5%. The treebank used was AnCora [7,8]. The two participant groups with the highest total score for Spanish were [9] and [10] with 82.3% and 81.3% LAS, respectively. We are especially interested in Nivre's group research because we used their system (Maltparser 0.4) for the experiments presented in this paper and in our previous ones on improving parsing accuracy [2,3]. The evaluation shows that the approximation given by Nivre gives competitive parsing accuracy for the languages studied. More specifically Spanish parsing scored 81.3% LAS; it was only 1 point under the best one [9], which did not use the Nivre algorithm but an Eisner's bottom-up span algorithm.

In our work, the first step was to replicate the participation of Nivre's group in the CoNLL-X Shared Task for Spanish [3]. We obtained the same results as Nivre's group, i.e., LAS = 81.30%, UAS = 84.67% and LA = 90.06%. These results served as a baseline for this work to determine ways to improve them.

3 The Development of N Specific Parsers

Considering the baseline experiment described in Section 2, despite a high overall parsing accuracy only 358 wordforms of the test corpus obtain a 100% LAS, UAS and LA in all parsed sentences, i.e., only 6.3% of the wordforms. If considering sentences, only 38 sentences of the test corpus (18.4% of them) were parsed without errors. An end user should usually expect a high local parsing accuracy (at the sentence level) rather than a high overall parsing accuracy. But nowadays a remarkable percentage of sentences in Spanish shows almost one error when parsed by Maltparser.

As described in [3], when analizing the results after parsing the test corpus, we found that there is a small set of words that show an incorrect attachment, labelling or both. These words are the prepositions “a” (*to*), “de” (*of*), “ en” (*in*), “con” (*with*), “por” (*for*), the conjunction *and*, which has two wordings: “y” or “e”, and the nexus “que” (*that*). For instance there are 20 sentences (340 wordforms), in the test corpus presented in Section 2, with only one error after parsing. That is 9.7% of the corpus’ sentences and 5.98% of its wordforms. We found that in 10 of these 20 sentences the only failure is caused by one of the words listed above.

Our hypothesis is that by enhancing local accuracy, not only overall accuracy should be enhanced, but end user satisfaction should be increased. We carried out a set of experiments to confirm or reject this hypothesis. The basic idea was to do an in-depth study of each one of the words listed above. This study, as described in [3], identified the set of different cases in which each word could be attached and labelled and train a specific parser for each case found. By doing so, we analyzed the conjunction and the preposition “a” in order to determine the feasibility of the technique. We found four different cases in which the conjunction could be attached and labelled, and six cases for the preposition “a”. So we trained 10 different specific parsers for covering the set of cases given for the conjunction and the preposition “a”. After this, the test set was parsed by combining the action of the parser described in Section 2 and the other 10 specific parsers. This way, when parsing a conjunction or a preposition “a”, the output of the general parser was ignored and was substituted by the output given by the specific parser for the given case. So the attachment and the label given for this word by the general parser were substituted by the attachment and the label given by the specific one. By doing so, overal LAS was increased by 0.87%, UAS by 0.84% and LA by 0.26%. These results encouraged us to continue with the experiment by training specific parsers for the rest of the words listed previously. The results obtained for all these words are shown in Table 1. They are usually better when using a specific parser than when using the general parser described in Section 2. But sometimes the specific parsers reach the same accuracy than the general parser, so it does not make sense to use the specific parser in such cases. For instance, when parsing the word *de* when attached to an adjective or an adverb, both the general parser and the specific parser show 100% LAS. Only when the word *y* (or *e*) acts as a nexus in coordinated copulative sentences could we not find a specific parser better than the general parser (the general parser reaches 81.3% $\text{LAS}_{y/e}$ and the specific parser reaches 75% $\text{LAS}_{y/e}$). In 21 of the 28 identified cases it was found better to use the specific parsers. Further research may produce better results for the specific parsers that do not reach 100% LAS yet.

In some cases the given improvement seems quite impressive. For instance, when parsing the word *de* when attached to a verb, the general parser shows 0% LAS and the specific parsers show 100% LAS. It is due to the little amount of samples present in the test corpus. For instance, if the test set contains only one sample for a specific case and this sample is correctly parsed then LAS = 100%. But it does not mean that the parser will parse every given sample of this case with 100% LAS. For the given example the test corpus contained only 4 samples. All these samples where wrongly parsed by the general parser but perfectly parsed by the two involved specific parsers. So LAS was enhanced from 0% to 100%, but this is for the given test corpus. If the test corpus had

Table 1. Attachment and labelling of all the studied words in AnCora. Found cases and specific LAS for each word and case, before and after the application of our method. The left arrow ($^{\leftarrow}$) after a part of speech indicates that this part of speech is before the considered word in the sentence. The right arrow ($^{\rightarrow}$) indicates that the part of speech is after the word.

		Case					
Word		#1	#2	#3	#4	#5	#6
y/e	**Label**	–	–	–	–		
	Attached to a	verb$^{\leftarrow}$	proper noun$^{\leftarrow}$	common noun$^{\leftarrow}$	adjective$^{\leftarrow}$		
	LAS$_{y/e}$ before	***81.3***%	80%	66.7%	80%		
	LAS$_{y/e}$ after	75%	***100***%	***80***%	***100***%		
a	**Label**	CD	CI	CC	CREG	–	–
	Attached to a	verb$^{\leftarrow}$					noun$^{\leftarrow}$
	LAS$_{a}$ before	62.5%	42.9%	60%	25%	***0***%	50%
	LAS$_{a}$ after	***87.5***%	***100***%	***100***%	***75***%	***0***%	***100***%
de	**Label**	CC	CREG	–	–		
	Attached to a	verb$^{\leftarrow}$		adverb$^{\leftarrow}$ adjective$^{\leftarrow}$	noun$^{\leftarrow}$		
	LAS$_{de}$ before	0%	0%	***100***%	83.3%		
	LAS$_{de}$ after	***100***%	***100***%	***100***%	***96.7***%		
que	**Label**	SUJ	–	SUJ			
	Attached to a	verb$^{\rightarrow}$		verb$^{\leftarrow}$			
	LAS$_{que}$ before	88.5%	86.4%	0%			
	LAS$_{que}$ after	***92.3***%	***95.5***%	***100***%			
en	**Label**	CC	CC	CREG	–		
	Attached to a	verb$^{\rightarrow}$	verb$^{\leftarrow}$		noun$^{\leftarrow}$		
	LAS$_{en}$ before	***83.3***%	92.6%	50%	62.5%		
	LAS$_{en}$ after	***83.3***%	***100***%	***100***%	***87.5***%		
con	**Label**	CC	CREG	–	–		
	Attached to a	verb$^{\leftarrow}$			noun$^{\leftarrow}$		
	LAS$_{con}$ before	60%	40%	***100***%	66.7%		
	LAS$_{con}$ after	***80***%	***100***%	***100***%	***83.3***%		
por	**Label**	–	CAG	CAG			
	Attached to a	noun$^{\leftarrow}$	comma$^{\leftarrow}$	adjective$^{\leftarrow}$			
	LAS$_{por}$ before	***100***%	***100***%	80%			
	LAS$_{por}$ after	***100***%	***100***%	***100***%			

contained more samples perhaps the specific parsers could not have reached 100% LAS. Usually the local improvement reached by the specific parsers is very high, but as said before it must be considered cautiously because of the limited amount of samples in our test corpus, that usually are between 2 and 10 for each case, being 30 the maximum. Nevertheless, as said in [2], parsing accuracy is reasonably homogeneus and similar accuracies should be expected even when increasing the number of samples in the test set.

In addition, we found that the word *de* attached to a verb with the label "–" is a given case in the training corpus that is not given in the test corpus. Of course, for this situation no error is given by the general parser, but how can we know if the parser can tackle such a situation if it is not present in the test corpus? This is because, if we want to obtain a high performing parser we must carefully build the train and test corpora.

4 Overall Accuracy, Local Accuracy and Their Limits

As seen in [3] and in Section 3, as a result of the use of specific parsers local accuracy can be improved and this redounds to the improvement of the overall accuracy. Dependency parsers can be useful for human end users, that presumably would use such parsers to analyze little pieces of text. So end users would feed dependency parsers with isolated sentences. In this case, even a single error in the parsing of one sentence is not acceptable. This is because the developers of dependency parsers should care for a high local accuracy. After parsing the test corpus with our n-version parser we got that 42 (20.3%) of the parsed sentences show no parsing errors, while 38 (18.4%) of them where perfectly parsed with the general parser. This improvement of the local accuracy, as shown in [3], has as a consequence not only a better experience for human end users but an improvement of the overall accuracy. When parsing the test corpus by combining the action of the general parser and our proposed specific parsers, we obtained the following results for overall accuracy: LAS = 82.68%, UAS = 85.73% and LA = 90.84%. It means an improvement of 1.38% LAS, 1.06% UAS and 0.78% LA in overall accuracy with respect to the results of the general parser alone.

N-version parsers are a way to improve parsing accuracy by systematically avoiding the errors given by a general parser. Nevertheless our experiments show a slight improvement. This improvement is bigger when eliminating the errors caused by a frequent word, as shown in Figure 1, 2 and 3. In each figure, each set of bars shows the increments of LAS, UAS and LA when adding the action of specific parsers for each word considered. The first word for which we added the action of its specific parsers was the conjunction (*y* o *e*). This is because the conjunction was most frequently parsed wrong by the general parser. Following this idea, we cumulatively added the action of specific parsers for each one of the considered words, firstly those that caused more parsing errors when using the general parser. In the end, when adding the action of specific parsers for the word *por*, we got the action in synergy of all the specific parsers listed in Table 1 and the general parser. We can observe in Figure 1, 2 and 3 that LAS, UAS and LA increased notably when adding the action of specific parsers for the conjunction and for the preposition *a*. In fact LA did not increase when the action of a specific parser for the conjunction was added, but this is because the general parser did not fail when attaching the conjuntion. Thus, the specific parsers could not improve this perfect attachment. In any case, in general terms the more infrequent the word that causes parsing errors the less the contribution of its specific parsers to the overall action. So the effort for building specific parsers may not be worth the obtained improvement. It is of interest to note that the conjunction causes 56 parsing errors with the general parser, *a* causes 48 errors, *de* 44 errors, *que* 42 errors, *en* 37 errors, *con* 17 errors and *por* 16 errors. Also, the increments obtained are not regular and this is because of the number

of samples of each considered case present in the test corpus and the accuracy of their specific parsers.

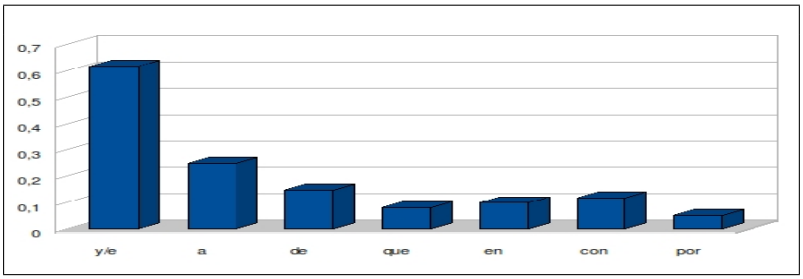

Fig. 1. Increments of overall LAS due to the action of specific parsers that avoid the more frequent errors, given by certain words

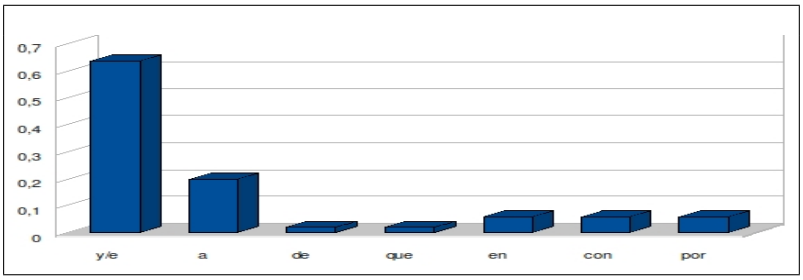

Fig. 2. Increments of overall UAS due to the action of specific parsers that avoid the more frequent errors, given by certain words

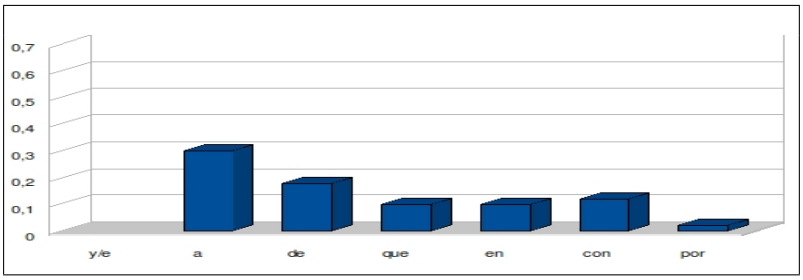

Fig. 3. Increments of overall LA due to the action of specific parsers that avoid the more frequent errors, given by certain words

Although a big percentage of the more frequent errors given are eliminated with the n-version parser, a significate number of errors remains. After our efforts, a 17,32% improvement in LAS is still required to reach a perfect parsing. Since specific parsers have been developed only for a small set of words, some other words remain without a specific parsing solution and continue causing errors. This means that to reach a perfect parsing an additinal significant effort is needed. A lot of errors could be avoided by implementing more complex n-version parsers, covering a large number of "difficult" words than the ones presented here. But some other errors could be inherent to the implementation of Maltparser and can not be avoided. Also, as suggested in [2] and in Section 3, some other errors could be avoided by carefully building the training corpora.

5 Conclusions and Future Work

In the present paper we show that n-version parsers are useful for improving dependency parsing accuracy in the case of machine learning-based systems.

We developed a n-version parser that improved the performance of a general parser alone. To do this we identified the seven words that were most frequently parsed incorrectly by the general parser. After this, we found the set of cases in which these words were given in the corpus and we trained Maltparser 0.4 to obtain a specific parser for each case. The improvements of this n-version parser are 1.38% LAS, 1.06% UAS and 0.78% LA better than the results of the general parser. Although it means a slight improvement was acquired, n-version parsers appear to be a useful method when developing high performing dependency parsers. But n-version parsers are not the definitive solution – they must be used in synergy with a systematic developement of training and test corpora and the improvement of the implementation and settings of machine learning-based dependency parsing generators. These results are statistically significant because we only focused in a small set of words. Also, it is important to notice that by improving the parsing of those words, more well-formed dependency trees are given. This is specially useful when a word, such as prepositions, that is the head of a subtree is correctly attached. By doing so all the subtree will be correctly attached.

Future work may be a more in-depth research on n-version parsers and the implementation of programs that must accurately send each word to the more appropriated specific parser.

Furthermore, this work which has focused on Spanish language using Maltparser 0.4 could similarly be applied for parsing other languages.

Acknowledgments

This work has been partially funded by *Banco Santander Central Hispano* and *Universidad Complutense de Madrid* under the *Creación y Consolidación de Grupos de Investigación* program, Ref. 921332–953.

References

1. Buchholz, S., Marsi, E.: CoNLL-X shared task on Multilingual Dependency Parsing. In: Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X). (2006) pp. 149–164
2. Ballesteros, M., Herrera, J., Francisco, V., Gervás, P.: Improving Parsing Accuracy for Spanish using Maltparser. Journal of the Spanish Society for NLP (SEPLN) **44** (2010)
3. Ballesteros, M., Herrera, J., Francisco, V., Gervás, P.: A Feasibility Study on Low Level Techniques for Improving Parsing Accuracy for Spanish Using Maltparser. In: Konstantopoulos, S., Perantonis, S., eds.: Proceedings of the 6th Hellenic Conference on Artificial Intelligence, SETN 2010, LNAI vol. 6040, Springer-Verlag (2010) pp. 39–48
4. Nivre, J., Hall, J., Nilsson, J.: Memory-based Dependency Parsing. In: Proceedings of CoNLL-2004, Boston, MA, USA (2004) pp. 49–56
5. Eisner, J.: Three New Probabilistic Models for Dependency Parsing: An Exploration. In: Proceedings of the 16th International Conference on Computational Linguistics (COLING-'96), Copenhagen (1996) pp. 340–345
6. Yamada, H., Matsumoto, Y.: Statistical Dependency Analysis with Support Vector Machines. In: Proceedings of International Workshop of Parsing Technologies (IWPT'03). (2003) pp. 195–206
7. Palomar, M., Civit, M., Díaz, A., Moreno, L., Bisbal, E., Aranzabe, M., Ageno, A., Martí, M., Navarro, B.: 3LB: Construcción de una base de datos de árboles sintáctico–semánticos para el catalán, euskera y español. In: Proceedings of the XX Conference of the Spanish Society for NLP (SEPLN), Sociedad Española para el Procesamiento del Lenguaje Natural (2004) pp. 81–88
8. Taulé, M., Martí, M., Recasens, M.: AnCora: Multilevel Annotated Corpora for Catalan and Spanish. In: Proceedings of 6th International Conference on Language Resources and Evaluation. (2008)
9. McDonald, R., Lerman, K., Pereira, F.: Multilingual Dependency Analysis with a Two-Stage Discriminative Parser. In: Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X). (2006) pp. 216–220
10. Nivre, J., Hall, J., Nilsson, J., Eryiğit, G., Marinov, S.: Labeled Pseudo-Projective Dependency Parsing with Support Vector Machines. In: Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X). (2006) pp. 221–225