Procesamiento del Lenguaje Natural, Revista nº 44, marzo de 2010, pp 83-90 recibido 20-01-10 revisado 16-02-10 aceptado 05-03-10

Improving parsing Accuracy for Spanish using Maltparser*

Mejora de la Precisión del Análisis para el Español con Maltparser

Miguel Ballesteros†, Jesús Herrera†, Virginia Francisco‡, Pablo Gervás‡
†Departamento de Ingeniería del Software e Inteligencia Artificial
‡Instituto de Tecnología del Conocimiento
Universidad Complutense de Madrid
C/ Profesor José García Santesmases s/n, Madrid, Spain
{miballes, jesus.herrera, virginia}@fdi.ucm.es, pgervas@sip.ucm.es

Resumen: En los últimos años los sistemas basados en aprendizaje automático desarrollados para realizar análisis sintáctico de dependencias han alcanzado una gran precisión, pero ésta está normalmente por debajo del 90% en *Labelled Attachment Score* (LAS). Maltparser es un ejemplo de ese tipo de sistemas. El aprendizaje automático permite obtener analizadores para cada lengua para la que se disponga de un corpus de entrenamiento adecuado. Dado que generalmente tales sistemas no pueden ser modificados, surge la siguiente cuestión: ¿Se puede mejorar el 90% en LAS utilizando mejores corpora de entrenamiento? En este artículo describimos trabajos prospectivos sobre la cuestión, estudiando estrategias en las que se consideran tanto el tamaño del corpus como las longitudes de sus frases con el fin de obtener una mejor precisión en el análisis.

Palabras clave: Análisis sintáctico de dependencias, Maltparser, español, precisión

Abstract: In the last years, dependency parsing has been accomplished by machine learning–based systems showing great accuracy but usually under 90% for Labelled Attachment Score (LAS). Maltparser is one of such systems. Machine learning allows to obtain parsers for every language having an adequate training corpus. Since generally such systems can not be modified the following question arises: Can we beat this 90% LAS by using better training corpora? In the present paper we show some prospective works on it. We studied some strategies considering training corpus' size and its sentences' length in order to obtain better parsing accuracy.

Keywords: Dependency parsing, Maltparser, Spanish, Accuracy

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 (Buchholz and Marsi, 2006). 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.

Inspired by ideas given by (McDonald and Nivre, 2007) and (Herrera and Gervás, 2008), our aim was not to give error measurements, theoretical justifications, demonstrations or to develop an implementation, but to empirically study the effect of training corpus' size and sentence's length on parsing accuracy. Therefore, we developed some prospective experiments whose results could be used as a guideline for further studies.

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. Since Spanish was the language for which we decided to develop the present work and we have developed some

* 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.

previous work on dependency parsing using Maltparser (Herrera and Gervás, 2008; Herrera et al., 2007a; Herrera et al., 2007b), we used Maltparser 0.4 to carry out our experiments.

The paper is organized as follows: Section 2 describes the CoNLL–X Shared Task focusing on Spanish participation; also we show our results when replicating the participation of Nivre's group in this task. Section 3 shows an experiment that tries to determine how stable is the accuracy of a given parser applying it to different texts. Section 4 studies the effect of training corpus' size on parsing accuracy. A training corpus usually contains sentences of different lengths; that is why the effect of this fact on parsing accuracy is studied in Section 5. Finally, Section 6 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 (Buchholz and Marsi, 2006) 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 (Nivre, Hall, and Nilsson, 2004), their Unlabelled Attachment Score (UAS), i.e. the percentage of "scoring" tokens for which the system had predicted the correct head (Eisner, 1996) and their Label Accuracy (LA), i.e. the percentage of "scoring" tokens for which the system had predicted the correct dependency label (Yamada and Matsumoto, 2003).

For Spanish, the results across the 19 participants ranged from 47.0% to 82.3% LAS, with an average of 73.5%. The treebank used was AnCora (Palomar et al., 2004), (Taulé, Martí, and Recasens, 2008), a 95,028 wordforms corpus containing open–domain texts annotated with their dependency analyses. AnCora was developed by the Clic group at Barcelona University. The two participant groups with the highest total score for Spanish were (McDonald, Lerman, and Pereira, 2006) and (Nivre et al., 2006) 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. Other participants that used the Nivre algorithm in the CoNLL–X Shared task were Johansson's group (Johansson and Nugues, 2006) and Wu's group (Wu, Lee, and Yang, 2006). Their scores on Spanish parsing were 78.2% (7th place) and 73.2% (13th place), respectively. 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 (McDonald, Lerman, and Pereira, 2006), which did not use the Nivre algorithm but a Eisner's bottom–up span algorithm in order to compute maximum spanning trees.

In our work, the first step was to replicate the participation of Nivre's group in the CoNLL–X Shared Task for Spanish. Therefore we trained Maltparser 0.4 (setted as referred by Nivre's group in (Nivre et al., 2006)) with the 89,334 wordforms training corpus provided in the CoNLL–X Shared Task. The parser obtained was tested with the 5,694 wordforms test corpus provided in the task. 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. It does not mean that we expected to beat these results, but instead to determine ways to improve them.

3 *Does a Parser Perform a Homogeneus Accuracy?*

After replicating the great results obtained by Nivre's group in the CoNLL–X Shared Task the next question arises: Could we expect the same results for every text parsed with a model trained with Maltparser 0.4? In order to find an answer for this question the following experiment was accomplished.

3.1 Configuration of the Experiment to Find if the Accuracy is Homogeneus

First of all we divided the whole AnCora corpus into 21 disjoint subcorpora of about 4,500 wordforms each, in order to use test corpora with a size similar to the one used in the CoNLL–X Shared Task. We distributed AnCora's sentences homogeneously among the 21 subcorpora according to their length, so every subcorpus contained a similar number of sentences for each sentence's length present in AnCora. Then we trained Maltparser 0.4 (set as referred by Nivre's group for the CoNLL–X Shared Task in (Nivre et al., 2006)) with each subcorpus, so we obtained 21 models ready for parsing. With each model we parsed the 20 subcorpora that were not used for training it. This way we obtained 21 × 20 = 420 parsed corpora that we evaluated. As evaluation metrics we computed not only LAS, UAS and LA but the following set of metrics that we consider useful to find further conclusions:

- The correlation coefficient between the LAS data series and the UAS data series performed by a model ($r_{LAS,UAS}$), the correlation coefficient between the LAS data series and the LA data series performed by a model ($r_{LAS,LA}$), the correlation coefficient between the UAS data series and the LA data series performed by a model ($r_{UAS,LA}$), all of them across the 20 evaluations. These coefficients indicate if a certain correlation exists between every pair of metrics independently of the parsed text.
- The maximum LAS (max_{LAS}), the minimum LAS (min_{LAS}), the maximum UAS (max_{UAS}), the minimum UAS (min_{UAS}), the maximum LA (max_{LA}) and the minimum LA (min_{LA}) performed by each model across the 20 evaluations. These values give us an idea of every metric's range of variation when parsing different texts.

3.2 Results and Conclusions of the Experiment

If we consider the 21 models, the one showing the maximum variation among its LAS series' values is A_8 with a difference of 5.21 points. The one showing the minimum variation among its LAS series' values is A_4 with a difference of 3.06 points. For UAS, the maximum variation is performed by A_2 with a difference of 9.43 points, and A_1 performs the minimum variation showing a difference of 2.41 points. Finally, A_2 reaches the maximum difference not only for UAS but also for LA (10.61 points), while the minimum variation occurs again with the model (A_4 with 2.26 points) that obtained the mininum variation for LAS. From these values we can conclude that each model can perform a relatively wide range of accuracy values depending on the texts that are used as input. But the overall results across the 21 models are more homogeneous, as can be seen in the last three rows of Table 1, from the fifth column to the last. In conclusion, we could say that when training two models with different corpora having a similar size and a similar number of sentences for every sentence's length, they should perform similar overall accuracies. However, each model could probably get notably different accuracy values depending on the specific text used as input. This is because we consider that training corpus' size and its sentence's length could contribute to the parsing accuracy performed. This way we carried out the experiments described in Sections 4 and 5. Moreover, the results discussed here may encourage a more complex evaluation for dependency parsing systems that not only assess the quality of the accuracies obtained but also assess the stability of this accuracy.

3.3 N–version parsers for a better performance

Based on the different accuracies obtained by parsing a subcorpus with different models, a last study was developed using the data obtained from the 420 trainings. If every A_i is parsed with the model trained with $A_j, j \neq i$ will allow us to obtain the best possible value LAS of A_i and the best overall LAS value by combining the actions of all the models. Table 2 shows which model must be used to parse every A_i. This experiment should encourage the future development of n–version parsers. These parsers should consist of several specific models each one trained to obtain a high accuracy for a small range of sentences. Therefore, the system should select the specific model that would better parse the sentence that is used as input.

4 Does Training Corpus' Size Affect parsing Accuracy?

The present section shows an experiment focused on the analysis of the effect of training corpus' size on parsing accuracy. To analyse this effect, we incrementally built a training corpus and we evaluated the parsing performance for every trained model, as follows:

- First of all we selected the $A_i, 1 \leq i \leq 20$, for which we obtained a better

Sub–corpus	**r**$_{LAS,UAS}$	**r**$_{LAS,LA}$	**r**$_{UAS,LA}$	**max**$_{LAS}$	**min**$_{LAS}$	**max**$_{UAS}$	**min**$_{UAS}$	**max**$_{LA}$	**min**$_{LA}$
A_0	0.84	0.78	0.44	72.78%	69.06%	78.22%	74.20%	85.03%	82.45%
A_1	0.87	0.85	0.65	72.86%	68.81%	**77.29%**	**74.88%**	84.85%	82.36%
A_2	0.92	0.87	0.91	73.55%	68.40%	**78.47%**	**69.04%**	**85.85%**	**75.24%**
A_3	0.92	0.80	0.54	72.68%	69.01%	77.58%	74.42%	85.39%	82.35%
A_4	0.88	0.87	0.71	**72.02%**	**68.96%**	77.09%	74.31%	**84.71%**	**82.45%**
A_5	0.82	0.89	0.61	72.74%	69.32%	77.32%	74.90%	84.99%	82.40%
A_6	0.90	0.88	0.69	71.90%	68.42%	76.88%	74.32%	84.78%	82.20%
A_7	0.86	0.88	0.63	72.55%	68.16%	77.27%	73.61%	85.24%	81.95%
A_8	0.94	0.87	0.71	**72.92%**	**67.71%**	77.55%	73.65%	85.52%	82.03%
A_9	0.91	0.76	0.56	72.27%	68.23%	77.47%	73.99%	84.85%	82.35%
A_{10}	0.87	0.89	0.69	71.76%	68.19%	77.11%	73.00%	84.62%	81.74%
A_{11}	0.91	0.85	0.69	73.30%	68.37%	78.27%	73.68%	85.73%	82.52%
A_{12}	0.92	0.78	0.60	73.01%	69.27%	78.32%	74.62%	85.39%	82.43%
A_{13}	0.89	0.85	0.64	72.96%	69.61%	78.22%	74.46%	85.60%	82.19%
A_{14}	0.92	0.81	0.62	73.04%	68.29%	77.93%	74.07%	85.04%	82.27%
A_{15}	0.94	0.85	0.76	71.37%	67.81%	76.21%	72.60%	85.01%	82.42%
A_{16}	0.87	0.76	0.44	72.51%	68.83%	76.83%	73.89%	85.50%	82.17%
A_{17}	0.92	0.79	0.58	73.23%	68.82%	77.58%	74.17%	85.78%	82.77%
A_{18}	0.95	0.78	0.63	72.50%	67.40%	77.63%	73.18%	84.70%	81.82%
A_{19}	0.82	0.86	0.54	72.25%	68.99%	77.65%	74.19%	85.39%	82.99%
A_{20}	0.95	0.84	0.73	72.73%	68.28%	77.55%	73.68%	85.19%	82.07%
max	0,95	0,89	0,91	**73,55%**	**69,61%**	**78,47%**	**74,90%**	**85,85%**	**82,99%**
avg	0,90	0,83	0,64	**72,62%**	**68,57%**	**77,54%**	**73,76%**	**85,20%**	**81,96%**
min	0,82	0,76	0,44	**71,37%**	**67,40%**	**76,21%**	**69,04%**	**84,62%**	**75,24%**

Table 1: Results obtained by the models trained with the 21 subcorpora in which AnCora corpus was splitted.

LAS when we parsed it with the model trained with A_0 in the experiment described in section 3.2. This subcorpus was A_6 and it was the first one added to the incremental training corpus.

- In every iteration we trained Maltparser 0.4 with the incremental corpus and we tested the trained model by parsing A_0 with it.
- In every iteration we added to the incremental corpus the non–used A_i, for which we obtained a better LAS when we parsed it with the model trained with A_0 in the experiment described in section 3.2.
- We iterated 20 times until every A_i was added to the incremental training corpus.

Therefore, in each iteration the training corpus maintains the percentages of sentences according to their lengths.

Input	**Parsed by model**	**LAS**
A_0	A_{14}	72.58 %
A_1	A_{18}	72.20 %
A_2	A_{17}	70.93 %
A_3	A_1	70.99 %
A_4	A_{12}	72.47 %
A_5	A_{13}	70.17 %
A_6	A_2	73.55 %
A_7	A_0	71.58 %
A_8	A_4	71.57 %
A_9	A_{17}	73.23 %
A_{10}	A_3	71.91 %
A_{11}	A_3	71.78 %
A_{12}	A_3	71.27 %
A_{13}	A_7	69.83 %
A_{14}	A_6	71.90 %
A_{15}	A_5	70.24 %
A_{16}	A_5	70.07 %
A_{17}	A_1	70.77 %
A_{18}	A_{12}	71.94 %
A_{19}	A_{17}	71.40 %
A_{20}	A_{17}	70.79 %
Avg LAS		71.48 %

Table 2: models trained with $A_j, j \neq i$ that must be used to parse every A_i to obtain the best possible average value of LAS.

The results of this experiment are showed in Figure 1. Considering LAS, from the first to the second iteration it becomes almost 3 points higher. From the second to the third iteration LAS increases 1.38 points. In the fourth iteration LAS is 1.2 points higher. And it increases almost 1 point in everyone of the fifth and the sixth iterations. By adding about 22,600 wordforms to the training corpus that we had in the first iteration we obtained a LAS increment of 7.56 points. But by adding another 22,600 wordforms LAS increments only 1.63 points. Taking into account the considerations given in section 3.2 this last increment is not signifi-

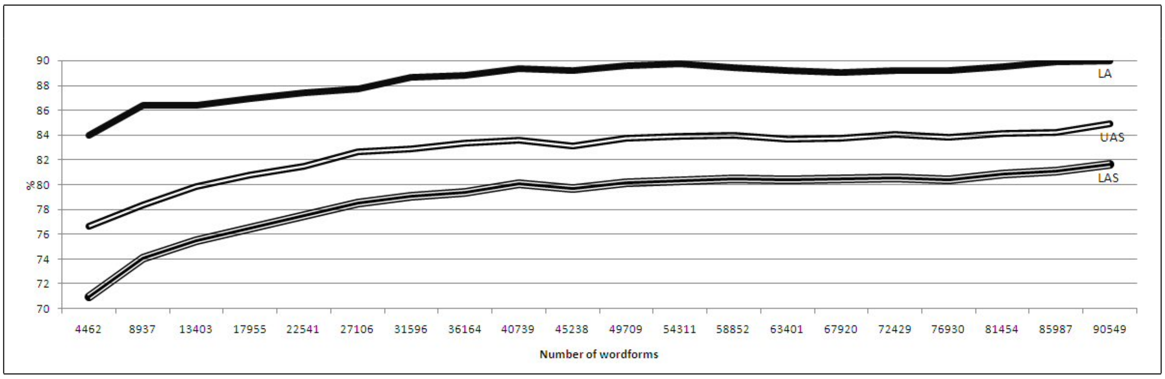

Figure 1: LAS, UAS and LA depending on the number of wordforms contained in the training corpus.

cant. So it might seem that after a certain limit the training corpus' size does not add a relevant contribution to parsing accuracy. UAS and LA show a similar behaviour than LAS. A similar behaviour is observed when using each different A_i for training and the other subcorpora for testing by following the method described above.

5 *Does Sentence's Length Affect parsing Accuracy?*

Considering previous research done by (McDonald and Nivre, 2007) (in which parsing errors related to sentence's length are studied) and (Herrera and Gervás, 2008) (in which the effect of sentence's length on accurate parsing is studied), the other parameter on which we focused this work was sentence's length. We have developed the following experiments to determine if the length of the sentences contained in the training corpus could affect parsing accuracy.

5.1 Training corpora containing one sized sentences

As a first test, we divided the section of the AnCora corpus that was provided as training corpus in the CoNLL–X Shared Task into 102 subcorpora, each one containing sentences of a unique length. So we obtained a subcorpus with 1 sentence of 143 wordforms, another subcorpus with 1 sentence of 130 wordforms, another subcorpus with 2 sentences of 128 wordforms and so on. By training Maltparser 0.4 with each one of these subcorpora, we obtained 102 different models. We parsed with each model the section of the AnCora corpus that was provided as test corpus in the CoNLL–X Shared Task and we obtained the results that Figure 2 shows. In this figure we plot LAS, UAS and LA for each parsing. In the x axis we represent the length of the sentences contained in the subcorpus used as a training corpus. LA values are represented by the upper line, UAS values are represented by the following line and LAS values are represented by the last line. The inner graph shows the distribution of sentences in the AnCora corpus according to their length, so in the x axis we represent the value of the length and in the y axis the number of sentences.

As can be observed, training corpora containing long length sentences gave remarkable accuracy despite its small size (in wordforms). For example a 143 wordform corpora, with only one long length sentence, gave us 45.88% LAS, 51.16 % UAS and 66.15% LA. AnCora has 35 sentences with 80 or more than 80 wordforms. Figure 2 shows that long length sentences are a very small part of the corpora. Could we get good results, considering only long length sentences? To test it, we sistematically added sentences creating new training subcorpora with only long length sentences, and we trained a model with each subcorpus. Starting with sentences of 120 wordforms or more, then we repeated the same experiment but adding sentences of 110 wordforms or more to the previous subcorpus. Next we added sentences of 100 wordforms or more, next of 90 wordforms or more and finally of 80 wordforms or more. Table 3 shows LAS, UAS and LA values for these subcorpora. The results are notably high, comparable to the ones obtained with a bigger corpus of more than 4,500 wordforms and containing sentences of all kind of sizes.

To compare these results we trained with only short length sentences and we obtained the results shown in Table 4. There are 571

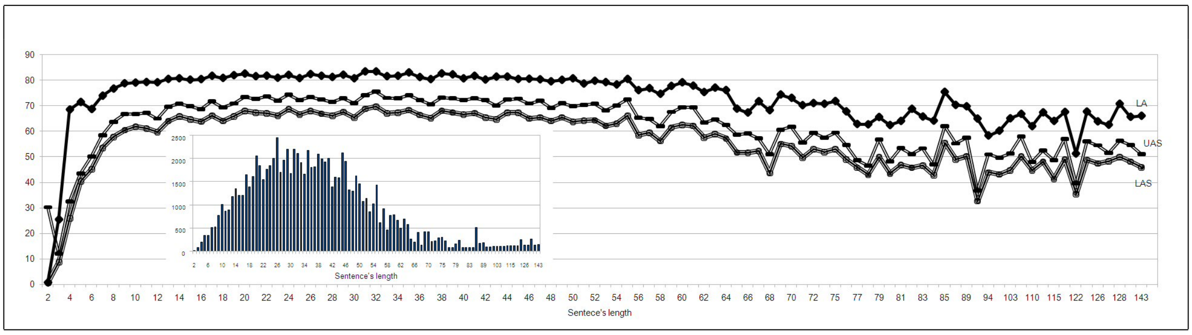

Figure 2: LAS, UAS and LA when training with corpora corpora containing one sized sentences.

Subcorpus	Size	LAS	UAS	LA
120..143	1154	61.07%	67.60%	77.18%
110..143	1612	64.64%	70.23%	80.24%
100..143	1919	66.68%	71.99%	81.41%
90..143	2104	66.86%	72.17%	81.90%
80..143	3538	68.60%	74.11%	82.72%

Table 3: Results obtained by the models trained with the subcorpora, only long length sentences (size in wordforms).

Subcorpus	Size	LAS	UAS	LA
2..7	1471	57.60%	62.19%	78.10%
2..8	1991	62.16%	66.86%	79.75%
2..9	2765	64.51%	68.90%	81.52%
2..10	3775	66.84%	72.30%	82.79%

Table 4: Results obtained by the models trained with the subcorpora, only short length sentences (size in wordforms).

sentences in AnCora with 10 wordforms or less. From these results it can be concluded that longer sentences contribute more than shorter sentences to overall accuracy. Thus, a training corpus containing only long sentences needs less wordforms than a training corpus containing only short sentences to achieve similar accuracies.

5.2 Training corpora containing the best performing sentences

Next, we developed another experiment focused on sentence's length as follows: We incrementally built a training corpus and we evaluated the parsing performance for every trained model, as follows:

- First of all, we selected the subcorpus for which we obtained a better value of LAS in the previous experiment described in this section.
- In every iteration we trained Malparser 0.4 with the incremental corpus and we tested the trained model by parsing the section of the AnCora corpus that was provided as test corpus in the CoNLL–X Shared Task with it.
- In every iteration we added to the incremental corpus the non–used subcorpora for which we obtained a better value of LAS in the previous experiment described in this section.
- We iterated 102 times until every subcorpus was added to the incremental training corpus.

Figure 3 shows LAS, UAS and LA values for each parsing. In the x axis we represent the number of wordforms contained in the incremental training corpus in each iteration. From these values, we could conclude that the selection of training corpus' sentences according to their lengths permits the obtention of better overall LAS, UAS and LA with a smaller number of wordforms than when sentence's length is not considered.

6 Conclusions and Future Work

Nowadays dependency parsing systems show a notably high overall accuracy for a wide range of languages. Maltparser is one of the better representatives of this kind of system. But 90% LAS seems to be a *de facto* limit for contemporary dependency parsers. These parsers could, of course, be tuned to obtain better results. But if we consider the evolution of the results obtained by this kind of system, it is obviously a difficult task. On the other hand, not only systems but training corpora could be tuned. Moreover, dependency parsing systems with a more complex architecture could be implemented.

From Section 3 we learned that similar training corpora (i.e., with a similar size in

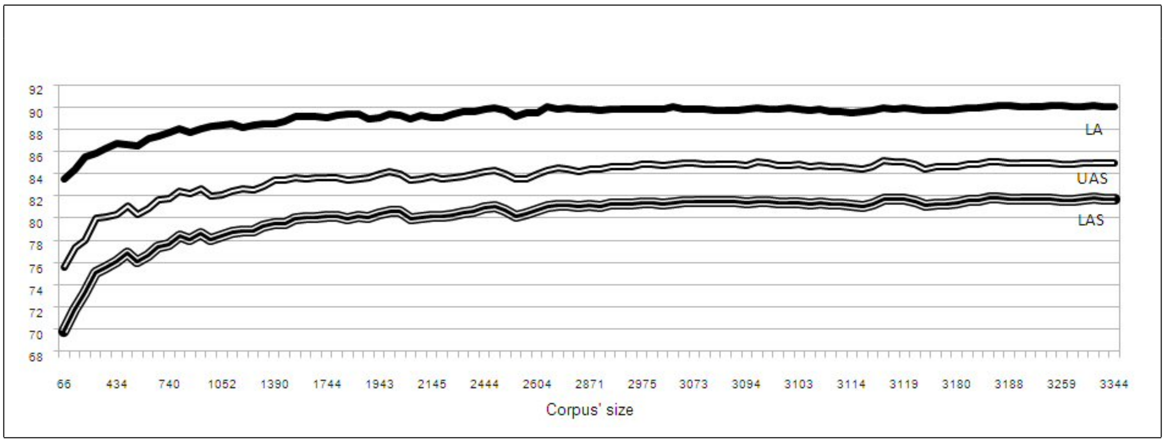

Figure 3: LAS, UAS and LA when considering sentece's length to build the training corpus.

wordforms and a similar distribution of sentences according to their length) used to train Maltparser 0.4, produce models that achieve similar maximum and minimum parsing accuracy values. In this aspect Maltparser 0.4 shows a stable behaviour. But these maximum and minimum values are not obtained for the same pieces of parsed text. Each model not only better parses a certain piece of text, but also shows notably different accuracies for each piece of parsed text. Stability is still not achieved when considering overall and local accuracy. However, users should usually expect not only a high but a stable accuracy for every parsed piece of text (normally one sentence). This is because this analysis could encourage the development of better parsers for real life. This way, the accomplishment of more complex assessment tasks for dependency parsers would be recommendable. These tasks should evaluate local accuracy and eventually consider only a small set of kinds of sentences. Edition by edition systems would be adapted to tackle new kinds of sentences and they would become more and more useful for real life applications. Also, future research could include a study on how to avoid overtraining on this kinds of corpora.

From Section 4 and other previous works such (Herrera and Gervás, 2008) it can be concluded that training corpus' size does not guarantee a high parsing accuracy by itself. When training samples are not elected one by one a big training corpus' size statistically permits the presence of a wider range of samples, and permits the presence of elements that induce noise. This seems to be a justification for the behaviour observed in Figure 1, i.e., the inclusion of new wordforms after a certain limit does not contribute to enhance accuracy.

As seen in Section 5 sentences' length is another fact to consider when building training corpora. A high overall accuracy can be obtained by training with a relatively small corpus containing carefully selected sentences according to their length.

The obtention with Maltparser 0.4 of a model not only able to perform high local accurate parsings but to perform a LAS higher than 90% seems to be a difficult task nowadays. Thus, as suggested in Section 3.3, n–version parsers could be studied as a way to reach these objectives. Specific models should be trained to obtain n–version parsers. Each specific model would parse one kind or a small range of some kinds of sentences. A specific training corpus should be built to obtain each specific model. Corpus' size and its sentences' length should be considered when building these specific models, trying to reach high local parsing accuracy and to avoid noise in training.

This work focused on Maltparser 0.4 and Spanish, but similar analises could be accomplished to study other languages and/or parsers, complementing the present one.

References

Buchholz, S. and E. Marsi. 2006. CoNLL–X shared task on Multilingual Dependency Parsing. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL–X)*, pages 149–164.

Eisner, Jason. 1996. Three New Probabilistic Models for Dependency Parsing:

An Exploration. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING–96)*, pages 340–345, Copenhagen.

Herrera, J. and P. Gervás. 2008. Towards a Dependency Parser for Greek Using a Small Training Data Set. *Journal of the Spanish Society for NLP (SEPLN)*, 41:29–36.

Herrera, J., P. Gervás, P.J. Moriano, A. Moreno, and L. Romero. 2007a. Building Corpora for the Development of a Dependency Parser for Spanish Using Maltparser. *Journal of the Spanish Society for NLP (SEPLN)*, 39:181–186.

Herrera, J., P. Gervás, P.J. Moriano, A. Moreno, and L. Romero. 2007b. JBeaver: un Analizador de Dependencias para el Español Basado en Aprendizaje. In *Proceedings of the 12th Conference of the Spanish Society for Artificial Intelligence (CAEPIA 07), Salamanca, Spain*, pages 211–220. Asociación Española para la Inteligencia Artificial.

Johansson, R. and P. Nugues. 2006. Investigating Multilingual Dependency Parsing. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL–X)*.

McDonald, R., K. Lerman, and F. Pereira. 2006. Multilingual Dependency Analysis with a Two-Stage Discriminative Parser. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL–X)*, pages 216–220.

McDonald, R. and J. Nivre. 2007. Characterizing the Errors of Data–Driven Dependency Parsing Models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 122–131. Association for Computational Linguistics.

Nivre, J., J. Hall, and J. Nilsson. 2004. Memory–based Dependency Parsing. In *Proceedings of CoNLL–2004*, pages 49–56. Boston, MA, USA.

Nivre, J., J. Hall, J. Nilsson, G. Eryiğit, and S. Marinov. 2006. Labeled Pseudo–Projective Dependency Parsing with Support Vector Machines. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL–X)*, pages 221–225.

Palomar, M., M. Civit, A. Díaz, L. Moreno, E. Bisbal, M. Aranzabe, A. Ageno, M.A. Martí, and B. Navarro. 2004. 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)*, pages 81–88. Sociedad Española para el Procesamiento del Lenguaje Natural.

Taulé, M., M.A. Martí, and M. Recasens. 2008. AnCora: Multilevel Annotated Corpora for Catalan and Spanish. In *Proceedings of 6th International Conference on Language Resources and Evaluation.*

Wu, Y., Y. Lee, and J. Yang. 2006. The Exploration of Deterministic and Efficient Dependency Parsing. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL–X).*

Yamada, H. and Y. Matsumoto. 2003. Statistical Dependency Analysis with Support Vector Machines. In *In Proceedings of International Workshop of Parsing Technologies (IWPT'03)*, pages 195–206.