Procesamiento del lenguaje Natural, nº 41 (2008), pp. 29-36 recibido 7-05-2008; aceptado 16-06-2008

Towards a Dependency Parser for Greek Using a Small Training Data Set*

Jesús Herrera†, **Pablo Gervás**‡
†Dep. de Ingeniería del Software e Inteligencia Artificial
‡Instituto de Tecnología del Conocimiento
Universidad Complutense de Madrid
C/ Profesor José García Santesmases, s/n, E-28040 Madrid
jesus.herrera@fdi.ucm.es, pgervas@sip.ucm.es

Resumen: Se han llevado a cabo experimentos con el fin de determinar estrategias de interés para la construcción de un corpus de pequeño tamaño con el que poder entrenar un analizador de dependencias de precisión para el griego, mediante una herramienta de aprendizaje automático. Para ello se han estudiado empíricamente diferentes problemas como la cobertura sintáctica, el efecto del orden de las palabras o el efecto de la morfología en lenguas en las que los roles sintácticos se expresan morfológicamente. En función de los resultados obtenidos se pretende establecer los fundamentos para el desarrollo sistemático y efectivo para el desarrollo de analizadores de dependencias cuando no se dispone de grandes corpora de entrenamiento. Las ideas presentadas podrían ser utilizadas no sólo para el griego sino también para otras lenguas.
Palabras clave: Análisis de dependencias, corpus de entrenamiento, griego, aprendizaje automático

Abstract: Some experiments have been accomplished in order to determine strategies that should be followed to build a small corpus capable to train accurately a dependency parser for Greek, using a Machine Learning tool. Thus, several problems that should be treated such syntactic coverage, effect of word order or effect of morphology in languages with syntactic roles expressed morphologically, are empirically studied. With the results presented we would like to lay the foundations for a systematic and effective way to develop dependency parsers when lacking huge training corpora. The ideas outlined could be used not only for Greek but for other languages.
Keywords: Dependency analysis, training corpus, Greek, machine learning

1 Introduction

In the last years, dependency parsing has been considered as a useful tool for Natural Language Processing. It could be observed in several international meetings, such the Recognizing Textual Entailment Challenge[1] or the Cross Language Evaluation Forum[2]. Initially, dependency parsing tools were available only for English; for instance, Minipar (Lin, 1998) is perhaps the mostly used software for English Dependency Parsing. But the need for dependency parsing tools for every language considered in Natural Language Processing research, led the organization of international evaluation tasks devoted to dependency parsing tools for several languages. For example, in the CoNLL Shared task dependency parsers were evaluated for the following languages: Arabic, Bulgarian, Czech, Danish, Dutch, German, Japanese, Portuguese, Slovene, Spanish, Swedish and Turkish, in the 2006 edition; and Arabic, Basque, Catalan, Chinese, Czech, English, Greek, Hungarian, Italian and Turkish, in the 2007 edition.

Dependency parsing tools for Greek were not documented until the CoNLL Shared Task 2007[3]. This suggests that there is room for research in this area. But annotated corpora with dependency analyses for Greek,

* We are very grateful to *Χριστίνα Γιαμαλή* and *Βασιλική Γιαμαλή* for their contribution to this work. This work has been partially supported by the Spanish Ministry of Education and Science (TIN2006-14433-C02-01 project).

[1] http://www.pascal-network.org/Challenges/RTE/

[2] http://www.clef-campaign.com

[3] http://depparse.uvt.nl/depparse-wiki/SharedTaskWebsite/

necessary for training systems, are not freely available. In addition, as an example of the effort necessary for building a dependency annotated corpus, Prokopis Prokopidis et al. reported in (Prokopidis et al., 2005) the process related to the *Greek Dependency Treebank*, which was used to train Maltparser for its participation in the CoNLL Shared Task 2007. It took a full months work to thirty annotators. The *Greek Dependency Treebank* contains 70,000 words. This situation gave rise to the idea of relying on an alternative kind of training based on small corpora. It is not easy to determine the wide range of cases involved in dependency parsing for a language, in order to generate accurate sets of training samples. Then, relatively big corpora are used for training, obtaining a "statistically guaranteed" high recall. Nivre et al. (Nivre et al., 2007) reached interesting results for Italian by using a "small"[4] training corpus of 1,500 sentences. But it is still a respectable amount of annotated sentences. The starting point for the approach presented here is the assumption that syntactic patterns can be found in every language; then, if such patterns can be identified in some way, a single example (or a few examples) for every pattern should be sufficient for training an accurate model for dependency parsing. In fact, one of the strategies used when humans learn languages is the memorization of syntactic patterns, that is widely exploited in learning methods. A good way to obtain syntactic patterns for the Greek language could be to analyze the sentences contained in a method for learning Greek. This is because the texts used in these first experiments were obtained from the online Greek course *Φιλογλωσσία* (Filoglossia)[5], provided by the *Ινστιτούτο Επεξεργασίας του Λόγου* (Institute for Language and Speech Processing, ILSP)[6]. In addition, the method *Επικοινωνήστε Ελληνικά* (Communicate in Greek) (Arvanitakis and Arvanitaki, 2003) was used too.

In summary, the goal of the present work is to analyze the possibilities for obtaining a dependency parser for Greek, from a good tool for dependency parser generation based on Machine Learning, but with the disadvantage of lacking a big corpus annotated with dependency analyses. This is a prospective study which does not try to demonstrate the validity of the techniques here proposed but to show a set of them that seem promising for the objectives stated. While our efforts were focused on the obtention of a dependency parser for Greek, the basis given here could be used in order to generate training corpora for different languages.

2 Training JBeaver

JBeaver is a publicly available tool configured originally as a dependency parser for Spanish (Herrera et al., 2007). But JBeaver provides features in order to easily reconfigure it as a dependency parser for virtually any language. This is because JBeaver is powered by Maltparser (Nivre et al., 2007) (Nivre et al., 2006a) (Nivre et al., 2006b), which is a dependency parser generator based on Machine Learning techniques that acts as a module of JBeaver. By supplying appropriate corpora to JBeaver, Maltparser models can be trained; these models are used by JBeaver as the core for its parsing action. Maltparser models need at their input not only the text to be analyzed but every word's part of speech (POS) tag. Since one of the goals when developing JBeaver was to offer to the user the possibility of parsing plain text, without any tagging at all, JBeaver needs to accomplish the POS tagging of the text at its input. This action is delegated to another Machine Learning tool acting as a module of JBeaver: Treetagger (Schmid, 1994). Treetagger is a tool for annotating text with POS and lemma information based on Decision Trees.

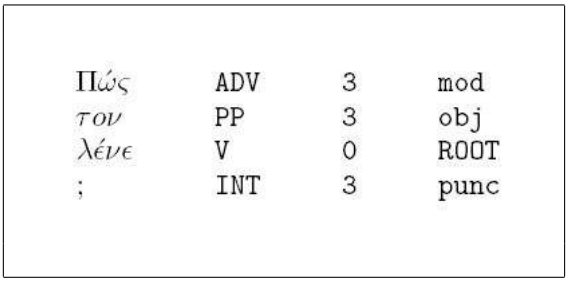
Since JBeaver uses Maltparser as its Machine Learning core, the training files must contain, for every word, its POS tag, a tag describing its syntactic function, and a pointer to the word that modifies, as required by Maltparser. In Figure 1, a excerpt of a training corpus for JBeaver (Maltparser) can be observed. The fields of every line of the file, from left to right, are the following: the word form, its POS tag, the numeric identifier of the phrase's word that acts as head of the actual word, and its syntactic function tag.

In addition, a pair of files containing the complete set of POS tags and syntactic function tags must be prepared beside the file

[4] This is the smaller training corpus reported by Nivre et al. in (Nivre et al. 2007).

[5] `http://www.xanthi.ilsp.gr/filog/default.htm`

[6] `http://www.ilsp.gr/`



Πώς	ADV	3	mod
την	PP	3	obj
λένε	V	0	ROOT
;	INT	3	punc

Figure 1: Excerpt of a training corpus for JBeaver.

mentioned above.

Maltparser provides two Machine Learning methods, i.e., a Memory–based Learning algorithm and a Support Vector Machine one (Nivre et al., 2007). This latter showed better results than the previous one in other works on dependency parsing, but Memory–based Learning was used because is the only one supported by JBeaver and the obtention of high accurate results was not a goal of the present work.

The set of features used to train a Maltparser model can be arbitrarily defined by the user, combining POS features, lexical features and dependency type features. The experiments presented here were developed using several sets of features, in order to study possible different behaviors when analyzing sentences. More specifically, the sets *m2*, *m4* and *m7*[7] were used. The *m2* set contains features related to: the POS of two items from the input string (I), the POS of one item from the stack of partially processed tokens (S), the dependency relation linked to one item from I and the dependency relation linked to three items from S; i.e., this set do not consider word forms at all. The *m4* set contains features related to: the POS of four items from I, the POS of one item from S, the dependency relation linked to one item from I and the dependency relation linked to three items from S, the complete word form of one item from I and the complete word form of one item from S. The *m7* set contains features related to: the POS of four items from I, the POS of two items from S, the dependency relation linked to one item from I and the dependency relation linked to three items from S, the complete word form of two items from I and the complete word form of two items from S. Alternatively, lexical features can be defined not only as complete word forms but as suffix features too (Nivre et al., 2006b). Following the idea given by Nivre et al. in (Nivre et al, 2006b), we trained JBeaver considering first of all the *m2* set, because for very small datasets it may be useful to exclude lexical features, in order to counter the sparse data problem.

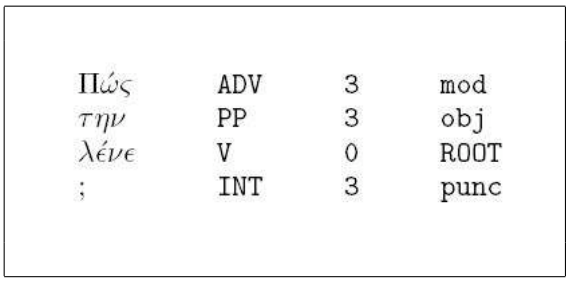
[7]See `http://w3.msi.vxu.se/~nivre/research/MaltParser.html` for a detailed description of *m2*, *m4* and *m7*.

3 First Step: Could a working system be trained using a small data set?

Previous work on training a statistical dependency parser for Chinese using small training data sets have been developed (Jinshan et al., 2004), and they obtained interesting results (80.25% precision) with a training corpus of 5,300 sentences. Furthermore, similar results were obtained for Italian by Nivre et al. (Nivre et al., 2007), with a training corpus of 1,500 sentences. Thus, the size of the corpus should not necessarily be a problem.

Some preliminary work showed us that, when analyzing a huge set of sentences, some syntactic structures occur frequently. Thus, if a single example for every kind of syntactic structure could be captured, a complete set of possible statements in a language might be modeled by means of a restricted set of samples that cover all of their syntactic patterns.

To study this approach, a first experiment was carried out in order to determine if having supplied a single analyzed sentence to JBeaver for training, the resulting model could parse accurately a set of different sentences showing the same syntactic structure as the first one. A model was obtained by training JBeaver with the following sentence: Πώς την λένε; (What is her name?), annotated as seen in figure 2.



Πώς	ADV	3	mod
την	PP	3	obj
λένε	V	0	ROOT
;	INT	3	punc

Figure 2: Example of one–sentence training file for JBeaver.

The following set of sentences were successfully parsed with the learned model:

Πότε το είδες; (When did you see it?), Πού τον βρήκες; (Where did you find him?), Ποιός το θέλει; (Who wants it?). The model was trained using the *m2* set of features, which means that only POS and dependency type features were considered. This (simple) set of features applied to a single training sample was sufficient for capturing a wide range of phenomena. It can be observed that past and present tenses could be treated; this is because no auxiliary is necessary for building these tenses in Greek. In contrast, other tenses such as future need different models for training because an auxiliary is mandatory (*θα*, *να*). In addition, every kind of adverb and personal pronoun can be successfully analyzed.

However, one of the characteristic features of the Greek language presented an additional problem: a sentence's components not necessarily follow a strict order. For instance, *είμαι ο Κώστας* and *ο Κώστας είμαι* contain the same words in different orders while meaning the same (I am Kostas). This relative independence of word order is a feature of some languages that makes dependency parsing more useful than constituency parsing to capture their full complexity, so it is important that the benefits of simplifying training do not come at the cost of losing the ability to model this feature. In search for a solution to this problem within the scarce data approach to training, the following experiment was carried out: having trained JBeaver with the dependency analysis of the sentence *Μετά θα πάμε στο θέατρο* (We will go to the theater later), the sentences:

- *Θα πάμε μετά στο θέατρο*
- *Θα πάμε στο θέατρο μετά*

(meaning both "We will go to the theater later") were analyzed, and the results showed some errors. This suggests that, when selecting a sentence as a model for a determined syntactic structure, every possible reordering of its words must be considered if the training corpus is to have adequate coverage. Following this lesson, satisfactory experiments were carried out with training models capable of dealing with word reordering in the same dependency structure. For all these experiments, the models were trained by using *m2* and *m4* sets of features, obtaining correct results in both cases. Therefore, the consideration or not of lexical features seems not to be relevant for this kind of samples.

This last experiment shows that, despite the fact that parsing errors were produced when word reordering was not considered for training, some sections of the sentence were correctly analyzed. These sections correspond to dependency subtrees that are included in the training sample. For instance, the subtree having as nodes the first four words of the sentence *Θα πάμε στο θέατρο μετά* was correctly analyzed because this subtree was present in the training sample. This observation lead to the design of the experiment described in the following section.

4 Second Step: Could some samples for training include others?

Some syntactic substructures are common to a wide range of sentences; for instance (as shown in figure 3), the dependency tree of the sentence *Κάθε πρωί ο Πέτρος και η Αντιγόνη θα πίνουν καφέ στη θάλασσα* (Petros and Antigoni will drink coffee by the sea every morning) includes the dependency tree of every one of these other sentences:

- *Κάθε πρωί η Θεοδώρα βλέπει τηλεόραση* (Theodora watches television every morning).
- *Ο Φοίβος θα γράψει τραγούδια* (Phoibos will write songs).
- *Η Ελένη και ο Κώστας κάνανε βόλτα στο βουνό* (Eleni and Kostas went for a walk on the mountains).

Two inverse experiments were accomplished in order to determine whether it is better to use for training a sentence with a dependency tree as general as possible, or several sentences with a simpler dependency tree but such that their intersection covers a dependency tree equal to the more general one.

The first experiment consisted on training a model with the sentence *Κάθε πρωί ο Πέτρος και η Αντιγόνη θα πίνουν καφέ στη θάλασσα*, using it to parse the other sentences, i.e.: *Κάθε πρωί η Θεοδώρα βλέπει τηλεόραση. Ο Φοίβος θα γράψει τραγούδια. Η Ελένη και ο Κώστας κάνανε βόλτα στο βουνό.*

In the second experiment, the model was obtained by training with the three phrases

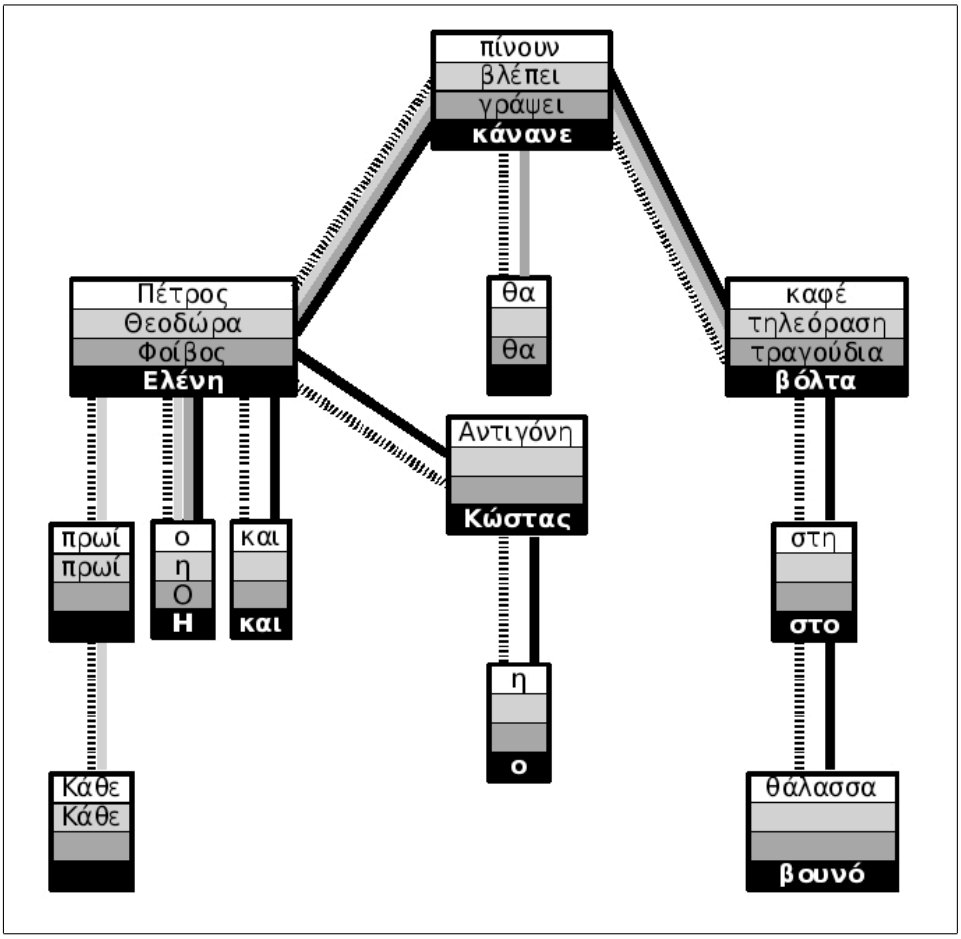


Figure 3: Included dependency trees.

used before in the parsing step: *Κάθε πρωί η Θεοδώρα βλέπει τηλεόραση. Ο Φοίβος θα γράψει τραγούδια. Η Ελένη και ο Κώστας κάνανε βόλτα στο βουνό.* The model was used to parse the sentence that in the first experiment was used for training, i.e., *Κάθε πρωί ο Πέτρος και η Αντιγόνη θα πίνουν καφέ στη θάλασσα.*

As a result, it was obtained that one sentence in the first experiment (*Ο Φοίβος θα γράψει τραγούδια*) was not correctly analyzed even if changing the set of features used for training, i.e., the consideration of different features did not improve the parsing action. But the second experiment gave satisfactory results, i.e., the sentence *Κάθε πρωί ο Πέτρος και η Αντιγόνη θα πίνουν καφέ στη θάλασσα* was correctly analyzed, but only when using a *m7* set of features, excluding the lexical ones, for training. As a conclusion, when considering a relatively complex syntactic structure, it seems better to use for training several sentences with a simpler dependency tree but such that their intersection covers a dependency tree equal to the more general one. In addition, a rich set of features trying to exclude lexical features should be used. Further studies should be developed in order to determine if this approach is appropriated for more complex structures or new strategies should be found.

5 Third Step: What about declination?

Greek is an inflected language that uses case to encode grammatical relations. For this reason, we developed an experiment to observe the effect of declination when trying to train JBeaver. Let us consider the following two sentences: *Το λεωφορείο του ΚΤΕΛ είναι του Θανάση* (The KTEL's bus belongs to Thanasis) and *Ο άνδρας της Ελένης είναι ο Θανάσης* (Thanasis is Eleni's husband). Both sentences follow the same order considering POS, i.e., determiner, common noun, determiner, common noun, verb, determiner, proper noun. But very important differences exist between them. For instance,

in the first sentence the word Θανάση lacks the last letter with respect to the same word in the second sentence (Θανάσης); this lacking letter indicates a possessive function and this proper noun acts as object in the first sentence, while in the second sentence the same proper noun acts as subject. Case in Greek is expressed by means of the word's suffix. Then, if the training sets for JBeaver must contain every word's POS, a training set conformed by these two sentences should induce errors in parsing time if lexical features were not considered when training. Thus, we trained a model using these two sentences, labeled as shown in Figure 4.

Το	DET	2	det
λεωφορείο	NC	5	s
του	DET	2	mod
ΚΤΕΛ	NP	3	gen
είναι	VBE	0	ROOT
του	DET	5	pred
Θανάση	NP	6	gen
.	PF	5	punc
Ο	DET	2	det
άνδρας	NC	5	pred
της	DET	2	mod
Ελένης	NP	3	gen
είναι	VBE	0	ROOT
ο	DET	5	det
Θανάσης	NP	6	s
.	PF	5	punc

Figure 4: Training sample for studying declination.

A sentence that should be parsed as the second one used for training (*Ο άνδρας της Ελένης είναι ο Θανάσης*) was submitted to the model learned. This new sentence was the following: *Ο πατέρας της Θεοδώρας είναι ο Λεωνίδας* (Leonidas is Theodora's father). As a result, a correct parsing was obtained only when training with an *m7* set of features. It means that to fully account for the complexity of the information contained in declination phenomena requires a relatively huge set of training features containing, of course, lexical features.

But the lexical features considered in the standard *m7* are complete word forms of the sentence. This could negatively affect the attempts to obtain a training corpus using small data sets for training, because under such circumstances the set of different word forms for every syntactic structure would not be very rich. Thus, it led us to evaluate the possibility of training a model capable of correctly analyzing declination phenomena. Then, we repeated successfully the experiment shown in this section, but using a modified *m7* model. In this new *m7* model every lexical feature consisted of the last character of every word form involved in the standard *m7*.

6 Evaluation

As seen in the previous sections, there are some facts that indicate that a working system could be trained using a small data set. But while the examples presented here work properly isolated, it is important to verify that all them can work together, in order to build an effective training corpus. The only inconvenience we could find is that different sets of features were needed for an accurate training of every case studied, while the model must be trained using a common set of features. Thus, a complementary proof was accomplished: the complete set of training samples seen in the previous sections was used to train a new model, and the set of features selected was the more restrictive of the ones considered along this work, i.e., the modified *m7* model where every lexical feature consisted of the last character of every word form involved in the standard *m7*. Then, this training corpus was conformed by the following sentences:

- Πώς *την λένε*; (What is her name?)
- *Μετά θα πάμε στο θέατρο* (We will go to the theater later)
- Θα *πάμε μετά στο θέατρο* (We will go to the theater later)
- Θα *πάμε στο θέατρο μετά* (We will go to the theater later)
- *Κάθε πρωί η Θεοδώρα βλέπει τηλεόραση* (Theodora watches television every morning)
- *Ο Φοίβος θα γράψει τραγούδια* (Phoibos will write songs)
- *Η Ελένη και ο Κώστας κάνανε βόλτα στο βουνό* (Eleni and Kostas went for a walk on the mountains)

- *Το λεωφορείο του ΚΤΕΛ είναι του Θανάση* (The KTEL's bus belongs to Thanasis)
- *Ο άνδρας της Ελένης είναι ο Θανάσης* (Thanasis is Eleni's husband)

After it, the following sentences were correctly parsed:

- *Πότε το είδες*; (When did you see it?)
- *Πού τον βρήκες*; (Where did you find him?)
- *Ποιός το θέλει*; (Who wants it?)
- *Θα πάμε πρώτα στο θέατρο* (We will go to the theater sooner)
- *Θα πάμε στο θέατρο μετά* (We will go to the theater later)
- *Κάθε πρωί ο Πέτρος και η Αντιγόνη θα πίνουν καφέ στη θάλασσα* (Petros and Antigoni will drink coffee by the sea every morning)
- *Ο πατέρας της Θεοδώρας είναι ο Λεωνίδας* (Leonidas is Theodora's father)

As a conclusion, the use of the modified *m7* set of features that we used for the experiment proposed in Section 5 seems valid for training a complete training corpus for Greek, considering a relatively important range of syntactic and morphological phenomena.

After this last experiment, it was interesting to determine empirically if a small training set could be sufficient to produce an accurate parser able to deal with a wide range of sentences. For that, a training set of 15 sentences was used to train a new parser. They were selected by following the recommendations learned from sections 3 to 5. The 15 sentences were, appart from the ones of the previous experiment, the following:

- *Η ώρα είναι δύο* (It is two o'clock)
- *Με λένε Γιώργο Οικονόμου* (My name is Giorgo Oikonomou)
- *Τις λένε Μαρία και Ελένη* (Their names are Maria and Eleni)
- *Είμαι σε ένα ξενοδοχείο* (I am in a hotel)
- *Τρέχει στο γήπεδο* (It runs on the ground)
- *Τώρα βλέπω τηλεόραση* (Now I am watching television)

The lexical features considered were, again, the ones pertaining to the modified *m7* model where every lexical feature consisted of the last character of every word form involved in the standard *m7*. 82 sentences were selected at random from the methods for learning Greek referred in Section 1, covering different levels of complexity pertaining to the A1 and A2 levels of the Common European Framework of Reference for Languages[8]. They were submitted to the parser generated and the following measures were computed: Labeled Attachment Score (LAS), Unlabeled Attachment Score (UAS) and a Label Accuracy (LA). 39 of these 82 sentences were perfectly analyzed, i.e., they ranked 100% LAS, UAS and LA. The overall values obtained (LAS = 67.32%, UAS = 77.27% and LA = 75.54%) are near to the results reported for Greek dependency parsing in the CoNLL Shared Task 2007[9]. It does not mean that they are comparable works, but it could be interpreted as a positive sign in order to consider the strategies exposed in this paper.

The next question to answer is if the parser is able to analyze correctly every sentence with a syntactic structure equal to one of those that were previously parsed without errors. For this, the following experiment was accomplished: every one of the 39 syntactic structures that were perfectly parsed was replicated several times by obtaining a set of different sentences for every syntactic structure. 100 sentences were thus generated and parsed, ranking again 100% LAS, UAS and LA.

7 Discussion and Future Work

After the set of experiments presented here, we can conclude that it could be possible to obtain an accurate dependency parser for Greek by means of a small training corpus. For this, we rely on the use of tools such JBeaver and we propose the following basic strategies to develop such training corpus:

[8] http://www.coe.int/t/dg4/linguistic/CADRE_EN.asp

[9] http://depparse.uvt.nl/depparse-wiki/AllScores/

- To select an adequate source for sentences in the language considered. This source should provide a wide range of samples containing as varied syntactic patterns as possible. Such kind of source could be a method for learning the language, that usually presents sentences with an incremental syntactic complexity.
- To extract sentences from the source to be analyzed. These sentences should cover as varied as possible typical cases of syntactic patterns in the language. If a sentence shows a complex syntactic structure, it is recommendable to use for training several sentences with a simpler dependency tree but such that their intersection covers a dependency tree equal to the more general one.
- To build the training corpus in an incremental way, verifying that the new sentences added in a step do not affect to the overall performance of the trained model. In addition, the set of features used for training should be as simple as possible, trying to avoid lexical features or using only words' suffixes. In case of a new syntactic pattern needing a richer set of features than the ones considered so far, it is necessary to verify that it does not come at the cost of losing accuracy during the parsing action.
- To evaluate specific phenomena of the language considered such as, for example, declination.

The present work covers preliminary studies on the question showing positive results that suggest that there is room for more research on it. The effective development of a training corpus, under the considerations exposed here, should reveal new problems to deal with. The solutions given to treat them should conform a useful and complete guide for the development of training corpora for dependency parsing using small training data sets. In addition, the results obtained would be used to empirically evaluate the advantages and disadvantages of the method exposed here versus prior existing ones.

References

K. Arvanitakis and F. Arvanitaki (*K. Αρβανιτάκης και Φ. Αρβανιτάκη*). *Επικοινωνήστε Ελληνικά [Communicate in Greek]*. Deltos Publishing, 2003.

J. Herrera, P. Gervás, P.J. Moriano, A. Muñoz, and L. Romero. JBeaver: Un Analizador de Dependencias para el Español Basado en Aprendizaje. In *Proceedings of the XII Conference of the Spanish Association for Artificial Intelligence, Salamanca, Spain*, 2007.

M. Jinshan, Z. Yu, L. Ting, and L. Sheng. A Satistical Dependency Parser of Chinese Under Small Training Data. In *IJCNLP–04 Workshop: Beyond Shallow Analyse-Formalisms and Statistical Modeling for Deep Analyses*, 2004.

D. Lin. Dependency-based Evaluation of MINIPAR. In *Proceedings of the Workshop on Evaluation on Parsing Systems, Granada, Spain*, 1998.

J. Nivre, J. Hall, and J. Nilsson. MaltParser: A Data-Driven Parser Generator for Dependency Parsing. In *Proceedings of the 5th International Conference on Language Resources and Evaluation, LREC-2006, Genoa, Italy*, 2006.

J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryiğit, S. Kóbler, S. Marinov, and E. Marsi. MaltParser: A Language-Independent System for Data-Driven Dependency Parsing. In *Natural Language Engineering 13 (2)*. Cambridge University Press, Cambridge, United Kingdom, 2007.

J. Nivre, J. Hall, J. Nilsson, G. Eryiğit, and S. Marinov. Labeled Pseudo-Projective Dependency Parsing with Support Vector Machines. In *Proceedings of the CoNLL-X Shared Task on Multilingual Dependency Parsing, New york, USA*, 2006.

P. Prokopidis, E. Desipri, M. Koutsombogera, H. Papageorgiou1, S. Piperidis. Theoretical and Practical Issues in the Construction of a Greek Dependency Corpus. In *Proceedings of The Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005), Barcelona, Spain,*, 2005.

A. Schmid. Probabilistic Part–of–Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing, Manchester, United Kingdom*, 1994.