Experiments for Dependency Parsing of Greek

Prokopis Prokopidis
Institute for Language
and Speech Processing
Athena Research Center
Athens, Greece
prokopis@ilsp.gr

Haris Papageorgiou
Institute for Language
and Speech Processing
Athena Research Center
Athens, Greece
xaris@ilsp.gr

Abstract

This paper describes experiments for statistical dependency parsing using two different parsers trained on a recently extended dependency treebank for Greek, a language with a moderately rich morphology. We show how scores obtained by the two parsers are influenced by morphology and dependency types as well as sentence and arc length. The best LAS obtained in these experiments was 80.16 on a test set with manually validated POS tags and lemmas.

1 Introduction

This work describes experiments for statistical dependency parsing using a recently extended dependency treebank for Greek, a language with a moderately rich morphology. Relatively small training resources like the one we use here can set severe sparsity obstacles for languages with flexible word order and a relatively rich morphology like Greek. This work presents ongoing efforts for evaluating ways of improving this situation. The rest of this paper is structured as follows: We describe the treebank and the tools for preprocessing it in section 2. After mentioning some relevant work, we present in section 4 different settings for experiments involving manually validated and automatically pre-processed data for morphology and lemmas. In section 5 we include a comparison of the output of two well-known statistical parsers in reference to a set of criteria. Section 6 describes work on using sentences from relatively large auto-parsed resources as additional training data.

2 Treebank

We use the Greek Dependency Treebank (Prokopidis et al., 2005) for all experiments. GDT includes texts from open-content sources and from corpora collected in the framework of research projects aiming at multilingual, multimedia information extraction. A first version of the GDT (GDT-2007) contained 70223 tokens and 2902 sentences, and it was used in the CoNLL 2007 Shared Task on Dependency Parsing (Nivre et al., 2007a). A recently extended version of the resource (henceforth GDT-2014) amounts to 130753 tokens (including punctuation) and 5668 sentences. The current version of the resource contains 21827 unique types, 11005 lemmas and 10348 hapax legomena (excluding dates, digits and proper names). The average sentence length is 23.07 tokens. GDT consists of 249 whole documents and can thus be used for the annotation of other, possibly inter-sentential, relations like coreference. Each document has 22.76 sentences on average.

The dependency-based annotation scheme used for the syntactic layer of the GDT is based on an adaptation of the guidelines for the Prague Dependency Treebank (Böhmová et al., 2003), and allows for intuitive representations of long-distance dependencies and non-configurational structures common in languages with flexible word order. Most trees are headed by a word that bears the `Pred` relation to an artificial root node. Other tokens depending on this root node include sentence-final punctuation marks

First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages, pages 90–96,
Dublin, Ireland, August 23-29 2014.

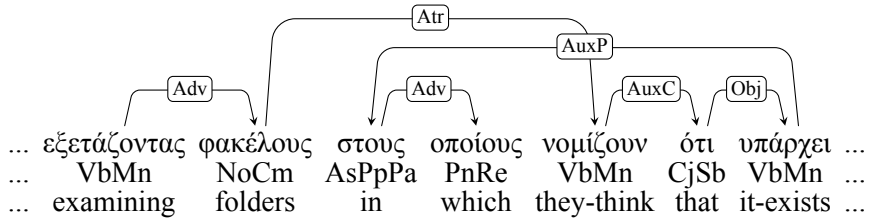

Figure 1: An analysis for a sentence fragment with a non-projective arc

and coordinating conjunctions. Coordinating conjunctions and apposition markers head participating tokens in relevant constructions. Table 1 contains some of the most common dependency relations used in the treebank, while Figure 1 presents a sentence fragment that contains a non-projective arc connecting the verb of a complement clause and its extraposed argument. In GDT-2014, 12.86% of the trees include at least one non-projective arc.

The relatively free word order of Greek can be inferred when examining typical head-dependent structures in the resource. Although nouns are almost always preceded by determiners and adjectives, the situation is different for arguments of verbs. Of the 5414 explicit subjects in GDT, 31% occur to the right of their parent. The situation is more straightforward for non-pronominal objects, of which only 4% occur to the left of their head. Of those subjects and objects appearing in "non-canonical" positions, 21% and 31%, respectively, are of neuter gender. This fact can pose problems to parsing, since the case of nominative and accusative neuter homographs is particularly difficult to disambiguate, especially due to the fact that articles and adjectives often preceding them (e.g. *το/the κόκκινο/red βιβλίο/book*) are also invariant for these two case values.

Dep. Rel	**Description**	**Dep. Rel.**	**Description**
Pred	Main sentence predicate	Adv	Adverbial dependent
Subj	Subject	Atr	Attribute
Obj	Direct object	Coord	A node governing coordination
AuxC	Subord. conjunction node	AuxP	Prepositional node

Table 1: Common dependency relations in the Greek Dependency Treebank

Apart from the addition of new material, another difference from previous versions is that GDT-2014 sentences have been manually validated for POS, morphosyntactic features and lemmas. The tagset used contains 584 combinations of basic POS tags (Table 2) and features that capture the rich morphology of the Greek language. As an example, the full tag `AjBaMaSgNm` for a word like *ταραχώδης/turbulent* denotes an adjective of basic degree, masculine gender, singular number and nominative case. The three last features are also used for nouns, articles, pronouns, and passive participles. Verb tags include features for tense and aspect, while articles are distinguished for definiteness.

Manual annotation at these levels allows to examine how the parser's accuracy is affected in realistic, automatic pre-processing scenarios. In these settings, POS tagging is conducted with a tagger (Papageorgiou et al., 2000) trained on a manually annotated corpus of Greek texts amounting to 455K tokens. During automatic processing, the tagger assigns to each token the most frequent tag in a lexicon compiled from the training corpus. A list of suffixes guides initial tagging of unknown words. When all tokens have been assigned a tag, a set of about 800 contextual rules learned during training, is applied to correct initial decisions. The tagger's accuracy reaches 97.49 when only basic POS is considered. When all features (including, for example, gender and case for nouns, and aspect and tense for verbs) are taken into account, the tagger's accuracy drops to 92.54. As an indication of the relatively rich morphology of Greek, the tags/word ratio in the tagger's lexicon is 1.82. Tags for a word typically differ in only one or two features like case and gender for adjectives. However, distinct basic parts of speech (e.g. `Vb/No`) is also a possibility.

Following POS tagging, a lemmatizer retrieves lemmas from a lexicon containing 66K lemmas, which

in their expanded form extend the lexicon to approximately 2M different entries. When a token under examination is associated in the lexicon with two or more lemmas, the lemmatizer uses information from the POS tags to disambiguate. For example, the token+POS input *εξετάσεις*/VbMn guides the lemmatizer to retrieve the lemma *εξετάζω (examine)*, while the lemma *εξέταση (examination)* is returned for *εξετάσεις*/NoCm.

POS	Description	POS	Description
Ad	Adverb	AsPpPa	Prep. + Article combination
AjBa	Adjective (basic degree)	CjCo	Coordinating conjunction
AsPpSp	Preposition	CjSb	Subordinating conjunction
AtDf	Definite article	NoCm	Common noun
AtId	Indefinite article	PnPo	Possessive pronoun
VbMn	Finite verb	PnRe	Relative pronoun

Table 2: Fine grained POS tags in GDT

3 Relevant work

Nakagawa (2007) was the best system in parsing the GDT in the CoNLL 2007 shared task, showing a 76.31 Labeled Attachment Score. Nakagawa's two-stage parser first constructed unlabeled dependency structures using sentence and token features, and then labeled the arcs using SVMs. The second best score for Greek was Hall et al. (2007), who scored 74.65 LAS using an ensemble system combining the output of six different Maltparser configurations. In recent work discussing the cube-pruned dependency parsing framework, Zhang and McDonald (2014) report a 78.45 LAS on the CoNLL dataset.

4 Experiments

In this section, we report on experiments using statistical parsers trained on automatically preprocessed and manually validated versions of GDT-2014. In all experiments we report the Labeled and Unlabeled Attachment Scores (LAS and UAS) and the Label Accuracy (LACC), with punctuation tokens counting as scoring tokens. We split the data of GDT-2014 in 90% and 10% training and test sets (5,101/567 sentences; 117,581/13,172 tokens). In this partitioning scheme, unknown tokens and lemmas when parsing the test set are 27% and 16%, respectively. We performed experiments with the transition-based Maltparser (Nivre et al., 2007b) and the graph-based Mateparser (Bohnet, 2010). For Maltparser, a 5-fold cross validation on the training set using MaltOptimizer (Ballesteros and Nivre, 2012) resulted in the selection of the non-projective stacklazy parsing algorithm as the one yielding an average best 78.96 LAS. Table 3 provides an abbreviated overview of the selected feature model, which is dominated by the top and first three elements in the parser's stack and its lookahead list. For Mateparser we used default settings.

Table 4 summarizes the results of our experiments. We observe a better 79.74 LAS with Mateparser with a larger difference in UAS than in LACC (2.37 vs 1.26). This may suggest that the two parsers agree on the labels they assign but differ more in discovering node heads. Not surprisingly, testing in a more realistic scenario of using automatic PoS, features and lemmas produces more errors (Figure 2). Maltparser shows a relatively smaller decrease in accuracy (-3.05 vs -3.45) in this context. In the next two experiments with Mateparser, we see that in automatic pre-processing scenarios, the tagger clearly contributes more to error increase (-3.34) compared to the lemmatizer (-0.06).

We also trained Mateparser in the MPL setting with POS tagsets of varying granularity, by removing features that were intuitively deemed to increase sparsity without contributing to parsing accuracy. More specifically, we experimented with several combinations of removing for aspect and tense of verbs, gender of nominal elements, definiteness of articles and degree of adjectives. A best LAS of 80.16 (cf. the two final columns of Table 4) was observed after removing features for degree and definiteness. Finally, and in order to examine how the expansion of the treebank has affected performance, we also

Tokens	Form	Lem	PoS	Feats	Dep	Tokens	Form	Lem	PoS	Feats	Dep
st[0]	+	+	+	+		rd(st[0])			+		+
st[1]	+	+	+	+		rd(st[1])					+
st[2]	+		+			hd(st[0])	+				
inp[0]			+			lh[0]	+		+	+	
ld(st[0])			+		+	lh[1]	+		+	+	
ld(st[1])					+	lh[2]			+		

Table 3: Automatically selected Maltparser features. Stack/Input (st/inp) tokens refer to tokens that are/have been in the stack of partially parsed tokens. Lookahead (lh) tokens are tokens that have not been in the stack. Features ld/rd/hd refer to the leftmost/rightmost dependents and and the head. We do not show features resulting from merging two or three features (e.g. merge3(PoS(lh[0]) + PoS(lh[1]) + PoS(lh[2])))

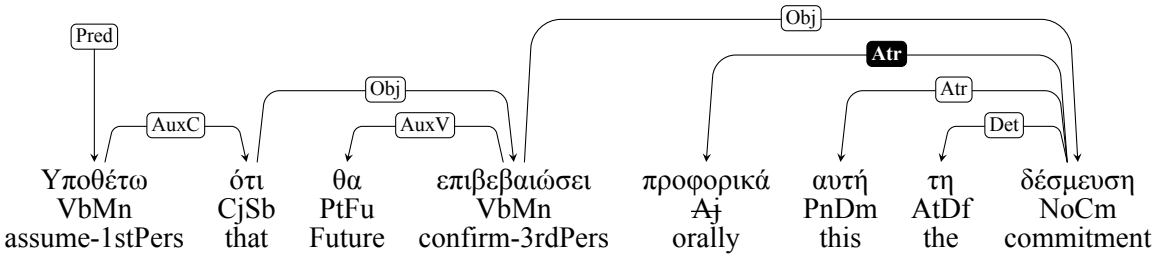

Figure 2: An example of a preprocessing error misguiding the parser: the wrong adjectival tag for the adverb *προφορικά* leads the parser in recognizing it as an attribute to a noun.

trained Mateparser in the MPL scenario using a training set equal in size to the 2.7K sentences of the CoNLL-2007 data. The results observed were 78.39 LAS and 84.77 UAS.

	MPL		APL		APML	MPAL	APL-AUTO		MFR1	MFR2
	Malt	Mate	Malt	Mate	Mate		Malt	Mate	Mate	
LAS	77.50	79.74	74.45	76.29	76.40	79.68	75.13	76.81	80.05	**80.16**
UAS	83.46	85.83	81.35	83.57	83.69	85.77	81.98	83.94	86.02	**86.29**
LACC	86.68	87.94	84.29	85.67	85.72	87.91	84.92	85.90	88.03	**88.13**

Table 4: Results from parsing GDT with Malt and Mate parsers: MPL refers to training and testing on manually validated POS, morphological features and lemmas; APL is evaluation on automatic POS, features and lemmas; APML is evaluation on automatic morphology and gold lemmas; MPAL on gold morphology and automatic lemmas. APL-AUTO is APL with training data including automatically parsed sentences. MFR1 is MPL after removing features for tense, aspect, degree and definiteness. MFR2 is MPL after removing features for degree and definiteness.

5 Error analysis

In this section we first provide a comparative analysis of errors by the two parsers on the 567 sentences test set. We use the set of length and linguistic factors proposed in the comparison between the Malt and MST parsers in McDonald and Nivre (2007). For example, in Figure 3, we plot sentence length in bins of size 10 and show, as expected, that the accuracy of both parsers decreases when analyzing longer sentences. Maltparser shows a higher accuracy for sentences of size up to 10, possibly because when parsing shorter sentences, early mistakes when making greedy decisions based on local features do not have a chance to lead to large error propagation. We omit details on UAS, where a similar pattern is observed. Figure 4 shows that Mateparser achieves better harmonic means of precision and recall, when longer dependencies are examined. This is again consistent with the fact that Maltparser favors shorter

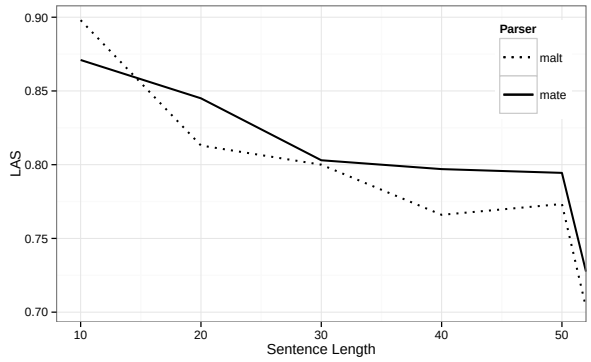

Figure 3: LAS relative to sentence length.

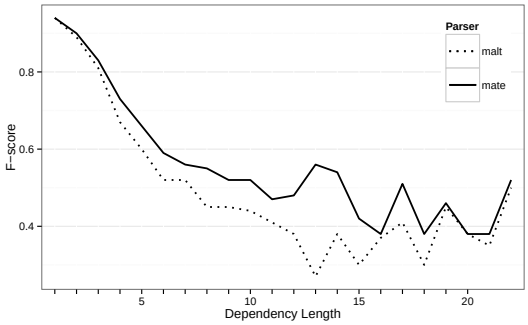

Figure 4: Dependency arc F-score relative to dependency length.

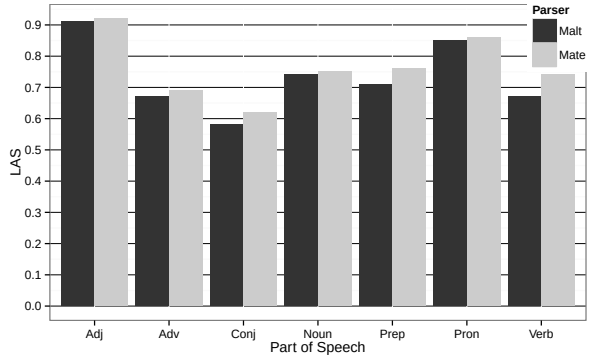

Figure 5: LAS for different POS tags.

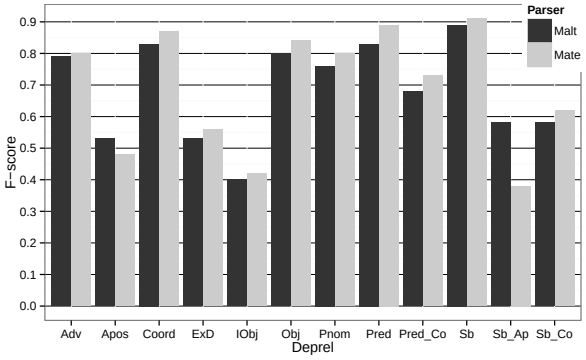

Figure 6: F-score for different relations.

arcs when making decisions based on local features only. We have seen that both parsers exhibit low F1-scores (Malt: 0.36; Mate: 0.30) in detecting non-projective heads.

In Figure 5 we see that Mate's LAS is better for all basic parts of speech. The difference is more evident for verbs, which are typically involved in longer dependencies. Finally, it is clear from Figure 6 that certain relations are particularly difficult for both parsers. For example, indirect object (`IObj`) dependents are low scoring nodes: this is because they are often attached to the correct head but are mislabeled as adverbial dependents (`Adv`) or plain objects (`Obj`). Dependents labeled as ellipsis (`ExD`) or heading appositional (`Apos`) constructions are also more error-prone. The same applies to nodes involved in coordinate structures as subjects headed by coordinative conjuctions (`Sb_Co`). The latter show an almost 0.3 drop in F1-score in comparison to simple subjects (`Sb`).

In the APL setting, errors by both parsers often involve some type of interaction between the relatively free order of Greek sentences and the case feature of nominal homographs. For example, in the case of the sentence *Διαφορετικά/different στοιχεία/figures δίνουν/provide τρεις/three επίσημες/official πηγές/sources για/on την/the ανεργία/unemployment (Three official sources provide different figures on unemployment)*, the two nominal arguments of the verb and all of their modifiers are ambiguous as far as case (Nominative/Accusative) is concerned. Both nominal arguments also agree with the verb in number. These facts, in combination with the OVS order of this and similar fragments present serious challenges to both the tagger and the parsers. In contrast, the case of the noun *ανεργία/unemployment* is easier for the tagger to disambiguate based on the preposition+article combination preceding it. However, attaching the whole subtree headed by the preposition is also problematic: it is part of a non-projective construction that would probably be disallowed in languages with a more strict order.

6 Use of autoparsed data

Following recent efforts in exploiting automatically processed data in training (Chen et al., 2012) and in accelerating treebank creation (Lynn et al., 2012), we conducted an experiment in extending the training set with similar material. We used a corpus of 66 million tokens, obtained by crawling (Papavassiliou et al., 2013) the 2009-2012 online archive of a Greek daily newspaper. We used models induced in the MPL experiment to parse all documents in the data pool with both parsers. We then appended to the original training set 30K randomly selected parsed sentences of 10 to 30 tokens length, for which identical trees were generated by both parsers. After retraining both parsers and testing on the APL test set, we observed (columns 8 and 9 of table 4) absolute LAS improvements of 0.68 and 0.52 for Maltparser and Mateparser.

7 Conclusions and future work

We described a set of experiments for dependency parsing of Greek using Maltparser and Mateparser, two well known representatives of the transition and graph-based families of parsers. Mateparser has exhibited the best accuracy on the test partition of a recently expanded version of the Greek Dependency Treebank, with Maltparser yielding higher scores on shorter sentences. After appending auto-parsed data to a training set manually validated for POS and lemmas, we observed small accuracy improvements that show room for improvement.

Scores obtained by training on datasets of different sizes in Section 4 probably indicate that apart from adding only documents or document fragments to the treebank, we should also consider selecting specific sentences for annotation, after measuring their informativeness and representativeness. In ongoing work, we are investigating ways of selecting sentences for manual annotation based on how much two or more parsers disagree, in combination with criteria like number of coordination/subordination elements and/or number of OOV words. For this purpose, we will also experiment with more members of the two parser families.

Our best LAS scores were obtained after mapping certain morphological features to default values. Since these tagset mappings may not be the most efficient ones, we plan to investigate automatic techniques for selecting optimal feature combinations.

Another line of research will be investigating semi-automatically mapping to different annotation schemes like the one proposed in McDonald et al. (2013). Finally, we plan to examine, as an additional source for resource expansion and domain adaptation, sentences from automatic dialogue transcriptions and/or product reviews.

Acknowledgments

Work by the first author was supported by the European Union Abu-MaTran project (FP7-People-IAPP, Grant 324414). Work by the second author was supported by the POLYTROPON project (KRIPIS-GSRT, MIS: 448306). We would like to thank the three anonymous reviewers and our colleague Vassilis Papavassiliou for their comments.

References

Miguel Ballesteros and Joakim Nivre. 2012. MaltOptimizer: An Optimization Tool for MaltParser. In *EACL*, pages 58–62.

Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká, 2003. *Treebanks: Building and Using Parsed Corpora*, chapter The Prague Dependency Treebank: A Three-Level Annotation Scenario. Kluwer.

Bernd Bohnet. 2010. Very High Accuracy and Fast Dependency Parsing is Not a Contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 89–97. Association for Computational Linguistics.

Wenliang Chen, Jun'ichi Kazama, Kiyotaka Uchimoto, and Kentaro Torisawa. 2012. Exploiting subtrees in auto-parsed data to improve dependency parsing. *Computational Intelligence*, pages 426–451.

Johan Hall, Jens Nilsson, Joakim Nivre, Gülsen Eryigit, Beáta Megyesi, Mattias Nilsson, and Markus Saers. 2007. Single Malt or Blended? A Study in Multilingual Parser Optimization. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 933–939.

Teresa Lynn, Jennifer Foster, Mark Dras, and Elaine Dhonnchadha. 2012. Active Learning and the Irish Treebank. In *Australasian Language Technology Workshop*, December.

Ryan McDonald and Joakim 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 (EMNLP-CoNLL)*, pages 122–131.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal Dependency Annotation for Multilingual Parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria.

Tetsuji Nakagawa. 2007. Multilingual Dependency Parsing Using Global Features. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 952–956.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007a. The CoNLL 2007 Shared Task on Dependency Parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Erygit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007b. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13:95–135, 6.

Harris Papageorgiou, Prokopis Prokopidis, Voula Giouli, and Stelios Piperidis. 2000. A Unified POS Tagging Architecture and its Application to Greek. In *Proceedings of the 2nd Language Resources and Evaluation Conference*, pages 1455–1462, Athens, June. European Language Resources Association.

Vassilis Papavassiliou, Prokopis Prokopidis, and Gregor Thurmair. 2013. A modular open-source focused crawler for mining monolingual and bilingual corpora from the web. In *Proceedings of the Sixth Workshop on Building and Using Comparable Corpora*, pages 43–51, Sofia, Bulgaria, August. Association for Computational Linguistics.

Prokopis Prokopidis, Elina Desypri, Maria Koutsombogera, Haris Papageorgiou, and Stelios Piperidis. 2005. Theoretical and practical issues in the construction of a Greek Dependency Treebank. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories*, Barcelona, Spain, December.

Hao Zhang and Ryan McDonald. 2014. Enforcing Structural Diversity in Cube-pruned Dependency Parsing. In *ACL*.