A Robust Parser for Unrestricted Greek Text

Sotiris Boutsis[1,2], **Prokopis Prokopidis**[1], **Voula Giouli**[1], **Stelios Piperidis**[1,2]

[1]Institute for Language and Speech Processing,
Artemidos 6 & Epidavrou, 151 25 Maroussi, Greece
{sboutsis, prokopis, voula, spip}@ilsp.gr

[2]National Technical University of Athens

Abstract

In this paper we describe a method for the efficient parsing of real-life Greek texts at the surface syntactic level. A grammar consisting of non-recursive regular expressions describing Greek phrase structure has been compiled into a cascade of finite state transducers used to recognize syntactic constituents. The implemented parser lends itself to applications where large scale text processing is involved, and fast, robust, and relatively accurate syntactic analysis is necessary. The parser has been evaluated against a ca 34000 word corpus of financial and news texts and achieved promising precision and recall scores.

1. Introduction

Advances in parsing technology have opened new possibilities for the application of natural language processing techniques in tasks in which accuracy, efficiency, and speed constraints either have made their application impossible in the past or have restrained their use in small data sets. The development of robust and time-conscious deterministic shallow parsers during the last decade has given rise to new applications and has allowed existing ones to benefit from linguistic processing when performing tasks involving large or very large amounts of real - life texts.

Lexicography is an area where the application of shallow parsing techniques is prominent. Bourigault (1992) describes LEXTER, a software package for extracting terminology in which surface grammatical analysis of the text is performed first. On that basis, term extraction takes place, since the grammatical form of terminological units is claimed to be relatively predictable. Boutsis et al. (1999) propose a method which processes bilingual parallel texts aligned at sentence level. The method implements statistical and linguistic techniques examining pattern grammars at both language sides of the corpus in order to extract bilingual associations between the terms of the two texts.

In information retrieval, several approaches have been suggested to raise the indexing unit from the word level to the multi-word or phrase level in order to emphasize on content carrying constituents. Zhai (1997), Evans and Zhai (1996) propose a method for fast noun phrase parsing and report on the application of this technique in order to enhance document indexing performance, in the framework of the CLARIT system (Evans et al., 1995). Stralkowski and Carballo (1995) propose the usage of parsing intensive methods in order to identify terminology, discover inter-term dependencies for building a conceptual hierarchy specific to the texts' domain and process the user's natural language requests into effective search queries.

Parsing techniques are of fundamental importance in information extraction systems. Pattern grammars and finite state techniques are performing remarkably well and some high scoring systems have replaced linguistically principled parsers with more efficient surface-syntactic analyzers (Grishman, 1995; Appelt and Hobbs, 1995).

From the above consideration, it becomes evident that a parser needs to conform with certain criteria in order to lend itself to applications like the mentioned ones. In real world applications, the parser should be able to deal with real life data, that is free texts, and should be robust with regards to phenomena pertinent to these texts. The accuracy of the parser is critical for its application in demanding environments, since in certain cases, it is preferable to partially analyze a phrase instead of producing a potentially noisy or misleading analysis. Also, ambiguity in the output could make the parser hard to integrate in real systems, especially if no later stage of processing can resolve it.

On these grounds, our efforts have focused on the design and implementation of a surface syntactic analyzer for Greek. The analysis is deterministic in the sense that ambiguous structures remain partially annotated, and are enclosed in larger constituents. The parser addresses applications where large scale text processing is needed but no full blown syntactic analysis is necessary. Processing speed, system robustness and relative accuracy of the results are the guidelines of the system's design, and are satisfied by adopting finite state techniques.

2. Background

One of the first deterministic parsers integrated in general purpose systems has been the Fidditch parser, (Hindle, 1983a; Hindle, 1983b). It is based on the principles proposed by Marcus (1980), but it has been targeted at processing free text including transcripts of spontaneous speech and at producing an analysis, partial if necessary, for each sentence. When Fidditch is unable to build larger constituents out of subphrases, it moves on to the next phrase, just including unattached constituents in the resulting partial parse tree.

Another deterministic parser is CASS (Abney, 1990). It is structured as a pipeline of simple filters.

Every filter makes a definite decision about a specific problem, but filters at later stages of processing can revise an earlier decision, in the light of new evidence discovered after parsing has progressed. Correcting such errors does not involve backtracking or unfolding the parser to an earlier state, thus avoiding speed compromises.

In (Brill, 1993b) and (Satta and Brill, 1996), a parsing method is proposed such that a transformational grammar, capable of parsing text into syntactic trees, is automatically learned from a training corpus. Training starts from a naive state and the system learns a set of ordered transformations, which can be applied to reduce parsing error, by repeatedly comparing the current state to the proper phrase structure for each sentence in the training corpus.

Karlsson et al. (1995), Voutilainen (1993), and Karlsson (1990) describe a syntactic annotation algorithm implementing constraint grammar checking. According to this approach, all possible syntactic categories are assigned to each word from a lexicon, in a way similar to part-of-speech (POS) tagging with constraints. Like POS disambiguation constraints, syntactic constraints are used to discard all contextually illegitimate syntactic labels. A flat syntactic description of each sentence is given in the output.

3. Method

The system we present in this paper is based on parsing via finite state techniques, (Abney, 1996; Abney, 1997; Grefenstette, 1996; Ait-Mokhtar and Channod, 1997). A text can be analyzed syntactically on the basis of grammars containing non-recursive rules written in the form of regular expressions, which can be translated into finite state automata or transducers by standard techniques (Roche and Schabes, 1997) and are then connected to form a finite state cascade, so that the output of an automaton or transducer is given as input to the next. Rules are numbered so as to be applied in a certain order and can recognize higher level constituents on the basis of the already described ones. A basic characteristic of this method is that parsing is deterministic and no backtracking takes place. No ambiguity is produced since each stage takes a definite decision about a phenomenon's existence or absence. This does not mean that ambiguities are resolved but that they are enclosed inside syntactic chunks, whose boundaries have been well recognized, although their internal structure may have not been decided. Since ambiguity is kept local, only one partial parse for each sentence is generated. In many cases, rules are designed to be reliable when they are applied using longest match, thus avoiding the need for disambiguation between different length instances of the same syntactic category. We utilize rules (Karttunnen, 1997) that either capture the structure of syntactic constituents or insert brackets at points believed to be the beginning or end of syntactic constituents. Rules are compiled to FST's using the FSA6 package (Van Noord and Gerdemann, 1999) and applied to the text using an efficient C parser.

4. General Architecture

The system architecture is depicted in Figure 1. Processing is performed through a set of pipelined standalone modules. The method takes Greek text at the input and produces a partial syntactic analysis at the output. The individual stages of processing are: text handling, POS-tagging, lemmatization, phrase and clause recognition, and grammatical relations identification.

Recognizing and labeling surface phenomena in the text is a necessary prerequisite for most NLP tasks. At the first stage, basic text handling takes place making use of a MULTEXT like tokenizer (Di Christo et al., 1995). This includes identifying word boundaries, sentence boundaries, dates, abbreviations, etc. Identifying word and sentence boundaries involves resolving ambiguity in punctuation use since structurally recognizable tokens may contain ambiguous punctuation; this may be the case for numbers, alphanumeric references, dates, acronyms and abbreviations. Following common practice, the tokenizer makes use of a regular expression definition of words, coupled with downstream precompiled lists for the Greek language and simple heuristics for distinguishing between these abbreviations or other evident abbreviations and final stops. This proves to be sufficient for recognizing sentences and words effectively.

After text handling has been performed, text is channelled to the part-of-speech (POS) tagging and lemmatisation stage. We use a version of the Brill (1993a) tagger trained on Greek text and a PAROLE compatible tagset, which, conforming to the guidelines set up by TEI and NERC, captures the morphosyntactic particularities of the Greek language. There are 584 different part-of-speech tags. The accuracy is around 90% when all features are examined and around 96% when only basic POS categories are taken into account. First, the tagger assigns initial tags, looking up in a lexicon created from the manually annotated corpus during training. A suffix-lexicon is used for initially tagging unknown words. 799 contextual rules are then applied to improve the initial phase output. After part-of-speech tagging has taken place, the lemmas are retrieved from a Greek morphological lexicon containing 70K lemmas.

Before the FST grammar is applied to the text, the tagset is reduced and tailored to the needs of the parsing task. Using a reduced tagset is advantageous, since it allows for more compact transducers with a smaller number of transitions and results in shorter compilation and parsing times. For instance, gender features are eliminated, since tests have indicated that noun phrase recognition can be performed with high accuracy without the use of such information.

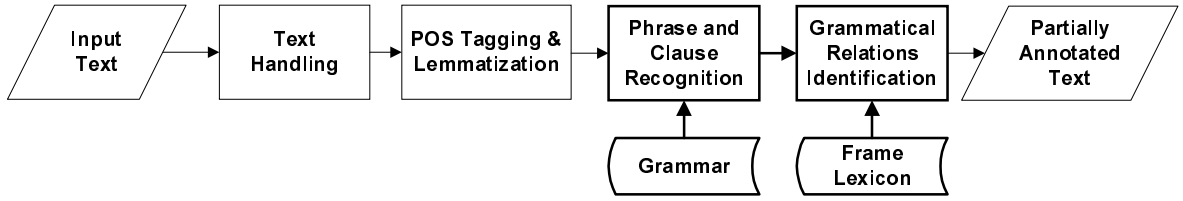

Figure 1: Processing Pipeline

```
%% insert 'stop' symbol before 'as' tags (prepositions)
markup(as, stop, []) o

%% group np's in accusative with post modifying np's in genitive
markup([const('[np_ac', '/np_ac]'), [const('[np_ge', '/np_ge]'), [{cjco,
punct_cm}^, const('[np_ge', '/np_ge]')]*]*], '[npacmax', '/npacmax]') o

%% mark pp with possibly coordinated np's in accusative
markup([stop, as+, {ptng, cjco, const('[np_ge', '/np_ge]'), const('[advp',
'/advp]')}*, const('[npacmax', '/npacmax]'), [{cjco, punct_cm},
const('[npacmax', '/npacmax]')]*], '[pp', '/pp]').

<EOR>    %% End Of Rule (stage 1)

%% remove stop symbol before prepositions already participating in pp's
['[pp', stop] => '[pp' o

%% mark pp with possibly coordinated np's in genitive
markup([stop, as+, {ptng, cjco, const('[advp', '/advp]')}*, const('[np_ge',
'/np_ge]'), [{cjco, punct_cm}^, const('[np_ge', '/np_ge]')]*], '[pp', '/pp]') o

%% delete temporary markers
{'[npacmax', '/npacmax]', stop} => [] o

<EOR>    %% End Of Rule (stage 2)
```

Figure 2: Sequence of rules recognizing prepositional phrases

Reduced features, however, are restorable in the output and can be used by later stages, if needed. Also, adapting the tagset to the parsing task requires the inclusion of lexical information in the POS tag of some words. This is usually the case for prepositions, adverbs, and conjunctions. For example, the preposition *σε* is tagged as *as_se* and the conjunctions *αν*, *εάν*, and *άμα* as *conj_cond*, which is indicative of their use in conditional clauses. Words not displaying the typical syntactic behavior of their POS are tagged differently. For example, the tags of the adjectives *όλος/all* and *ολόκληρος/whole* are given the prefix *olos.* This allows grammar rules to capture the use of these adjectives as predeterminers and postmodifiers. After all modifications, the tagset numbers ca. 180 tags.

After tagging, analyzed text is channeled into the parser. Parsing is performed in two stages. At the first stage, phrases and clauses are recognized on the basis of an FST grammar, while, at the second stage, grammatical relations between recognized constituents are established on the basis of a subcategorization lexicon and a pattern matching mechanism.

5. Corpus

The parser has been evaluated against a text collection composed of news and financial articles from online Greek magazines and newspapers. The total size of the collection is 33869 tokens, punctuation marks excluded. The texts were manually annotated by two linguists who used a Java graphical user interface for this purpose. A number of files were annotated by both linguists to ensure inter-annotator consistency. Inter-annotator agreement is around 95%.

6. Greek Grammar

In order to allow for a syntactic analysis of Greek text, the grammar contains rules recognizing the following phrasal categories: adjective phrase, noun phrase, verb group, prepositional phrase, and adverb phrase. At the clause level, the parser recognizes main and several types of subordinate clauses. This schema follows EAGLES (Leech et al., 1996).

In Figure 2, the set of rules that recognize prepositional phrases is given. A rule *markup(X, y, z)* encloses longest matches of the regular expression *X* in *y* and *z*, while a rule *X => y* replaces longest matches of *X* with *y*. Different rules can be composed

into one using the compose (*o*) operator. As can be seen, recognition of prepositional phrases takes place in two stages. At the first stage, phrases composed of a preposition followed by one or more noun phrases in accusative are recognized. At the second stage, phrases composed of a preposition followed by one or more noun phrases in genitive are recognized. Adverb phrases and negative particles are also taken into account.

A description of the grammar responsible for the recognition of each syntactic category follows.

6.1. Adverb phrases

Adverb phrases build, in principle, on head adverbs possibly modified by other adverbs. After prepositional and noun phrases have been identified at a later stage, they may be included in the AdvP as complements of the adverbs.

*[**advp** Δυστυχώς / Unfortunately **advp**] καμμία πρόταση / no decision δεν έγινε / was made …*

*[**advp** εκτός / apart [**pp** από / from [**np_ac** το Νίκο / Nikos **np_ac**] **pp**] **advp**]*

*[**advp** ανεξαρτήτως / regardless of [**np_ge** αποτελέσματος / the result **np_ge**] **advp**]*

6.2. Adjective phrases

Adjective phrases contain one or more adjectives or passive perfect participles, possibly premodified by one or more adverbs. Clitic pronouns following the head of the phrase are also included. Adjectives and participles separated by commas or coordinating conjunctions, are enclosed in the same AdjP. There is a subclassification of AdjPs according to the case of their head. Thus, *adjp_nm* represents nominative AdjPs, *adjp_ge* genitive AdjPs, and so on.

*Η / The **[adjp_nm** πολύ γρήγορη και αποτελεσματική / very fast and effective **adjp_nm]** απάντηση / response.*

*Η [**adjp_nm** καθημερινή τους **adjp_nm**] / Their everyday ενημέρωση / updating…*

6.3. Noun phrases

Apart from common and proper nouns, rules for the identification of noun phrases accept pronouns, adjectives, and participles as heads.

Any prenominal determiners and modifiers (pronouns, numerals, adjective phrases) are included in the NP, whereas postnominal constituents include adjectives, demonstrative pronouns, and clitic pronouns. Other postnominal modifiers like NPs in genitive or PPs are recognized independently of these base NPs.

The subclassification of AdjPs according to their case holds for NPs as well.

*[**np_nm** Η επικείμενη επίσκεψη / The impending visit **np_nm**] [**np_ge** του κ. Κλίντον / of Mr. Clinton **np_ge**] [**pp** στη / to [**np_ac** χώρα μας / our country **np_ac**] **pp**] θα προκαλέσει / will provoke …*

6.4. Prepositional phrases

Once NPs have been marked, identification of prepositional phrases is straightforward. PPs are composed of a preposition followed by one or more (coordinated) NPs. Postmodifying NPs in genitive are also enclosed in the PP.

*Έχει κερδίσει / He has benefited [**pp** από [**np_ac** την εξαγορά / from the acquisition **np_ac**] [**np_ge** της εταιρείας / of the company **np_ge**] **pp**].*

6.5. Verb groups

Verb groups include the head verbal form together with any auxiliaries for the formation of periphrastic tenses. Negative, future and subjunctive particles are enclosed as well. These complements (clitic pronouns) and modifiers (adverbs), which are "trapped" between the head verbal form and the auxiliaries and particles, while retaining their respective labels, are also included in the verbal group.

*Η Επιτροπή/The comission [**vg** ενέκρινε / approved **vg**] το έργο / the project..*

*Οι γιατροί / The doctors [**vg** δεν [**np_ge** τους **np_ge**] [**np_ac** το **np_ac**] έχουν [**advp** ακόμη **advp**] πει / have not yet said it to them **vg**].*

Labels *vg_s* and *vg_g* are used for the subclassification of verb groups with a subjunctive and present participle verb head, respectively.

*Οι Ευρωπαίοι / The Europeans προσπάθησαν / tried [**vg_s** να μποϋκοτάρουν / to boycott **vg_s**] τη συγχώνευση / the merger.*

*[**vg_g** Αναγνωρίζοντας / Recognizing **vg_g**] την ήττα του / his defeat…*

6.6. Clauses

After the basic phrasal constituents have been identified, the parser tries to capture their organization into clauses. Identification of clauses is guided by a list of accepted clause markers which are used to recognize potential clause boundaries. Subordinating conjunctions, relative pronouns or larger constituents containing them, adverb phrases, are used to mark possible clause boundaries. The existence of exactly one verb group in each clause is a strong criterion governing segmentation into clauses.

Both main and subordinate clauses are recognized. The latter include relative, relative indefinite, time, conditional, and interrogative clauses. Finally, clauses which are introduced by a conjunction that does not unambiguously indicate a certain type of clause, are labeled "other".

The following examples depict the types of subordinate clauses the parser recognizes.

Relative clauses: *[**cl** Για εκείνους / For those [**cl_rel** που / that υποφέρουν / suffer υπό το βάρβαρο καθεστώς / under the barbaric regime **cl_rel**] …**cl**]*

Relative indefinite clauses: *[**cl** [**cl_ri** Όποιοι / Those who εξασφάλισαν /secured την ελευθερία τους / their freedom, **cl_ri**] κέρδισαν / won …**cl**]*

Time clauses: *[**cl_t** Όταν / When η αγορά / the market κατέρρευσε / collapsed **cl_t**], [**cl** οι μέτοχοι / shareholders … **cl**]*

Conditional clauses: *[**cl_c** Αν / If η κυβέρνηση / the government αποφασίσει / decides την καταστολή /*

the suppression της απεργίας / of the strike ***cl_c****], [****cl** *οι εργαζόμενοι / the workers … **cl]***

Interrogative clauses: ***[cl** Συνειδητοποίησα / I realised **cl][cl_ir** πόσο / how much θα βοηθήσει / will help ο νόμος / the law … **cl_ir]***

Other clauses: ***[cl** Η εφημερίδα / The newspaper αποφάσισε decided **cl] [cl_o** να ενημερώσει / to inform τους αναγνώστες της / its readers … **cl_o]***

Relative and relative indefinite clauses are always embedded in other clauses. Clauses of other types are recognized inside larger clauses, only if trapped in them.

***[cl_c** Αν / If, **[cl_o** αφού ολοκληρωθεί η εξαγορά /after the acquisition has been completed **cl_o]**, δεν υπάρχουν διαθέσιμα κεφάλαια / there are no available funds **cl_c]**, **[cl** θα προχωρήσουμε /we will proceed … **cl]**.*

6.7. Grammatical relations

Grammatical relations are recognized by a module that processes texts after all phrase and clause labels have been unambiguously assigned by the finite state parser. First, a REF(erence) number is assigned to each token and syntactic label of the text. Then the module identifies the grammatical relations in each sentence, and indicates them by assigning tags of type STRUCT(ure). The tag consists of the type of the grammatical relation, and the REF numbers of the opening and closing brackets of the dependent constituents. In the example sentence of the Appendix, the verb *ενημέρωσε / informed* is followed by STRUCT<subj_np_nm,949,955>, which links the head verb with the nominative NP that begins at REF<949> and ends at REF<955>.

The module identifies phrase heads and, using their lemmas, retrieves subcategorization frames from a database containing subcat information for the 5927 most frequent verbs, 4950 most frequent nouns, and 375 most frequent adjectives of a general purpose corpus. Frames were manually constructed. A frame may contain mutually exclusive arguments. For instance, the frame for the verb *δίνω / give* includes slots for a noun phrase (subject) in nominative case, a noun phrase (direct object) in accusative, a noun phrase (indirect object) in genitive, and a prepositional phrase (indirect object) headed by the preposition *σε / to*. The last two constituents are alternative realizations of the same grammatical function.

```
δίνω #subj_np_nm# #obj_np_ac#
#ind_obj_np_ge# #ind_pp_se#
```

Possible grammatical roles included in the frame of verbs are nominative subjects, predicative phrases, direct objects in accusative, indirect objects in genitive, prepositional phrases functioning as complements, and clausal arguments. In case a verb has no frame, it is assigned the default frame *#subj_np_nm#*. Nouns and adjectives are examined for nominal and clausal dependents. For heads with more than one frames, all possibilities are examined; the frame with most matches is selected and finally applied.

Let us examine how subject NPs and predicative NPs and AdjPs are identified when this is required by the frame. The module searches at the clause level for constituents with an *np_nm* or an *adjp_nm* label. In case no *np_nm's* are found (a very common situation with a pro-drop language such as Greek), a *null_subj*(ect) label is assigned to the verb. Otherwise, it assigns, by default, the label *subj_np_nm* to nominative NPs and the label *pred*(icative) to AdjPs. In case the frame of the verb requires a predicative phrase and only one nominative NP phrase has been found, it recognises it as a *pred* after checking that it is not headed by a pronoun, in which case the module opts for a *subj_np_nm* label, instead. In case two nominative phrases, separated by punctuation or coordinating conjunctions, have been found, they are joined into a larger unit. If a nominative phrase occurs preverbally and another postverbally, the preverbal one is recognized as the subject and the other as the predicative phrase. Other types of dependents are identified following similar heuristics. Also, genitive NPs are linked to the preceding phrase. Some constituents can be linked to either of more than one heads. For example, an *np_ge* can be attached to a verb as its indirect object or to another NP as its postmodifier. The module resolves the conflict by attaching the dependent to the head it is closest to.

7. Sample output

In the Appendix, the output of the parser for the following sample sentence is given.

```
Η Αγροτική Τράπεζα / The Agricultural Bank ενημέρωσε / informed τους / the αρμόδιους / in charge ερευνητές / researchers της υπόθεσης / of the case ότι / that ανευρέθησαν / were found 33 στελέχη / 33 executives που / who είχαν προχωρήσει /had proceeded σε εικονικές προεγγραφές / to virtual subscriptions.
```

As can be seen from the analysis, one main and two subordinate clauses have been recognized. The parser has enclosed the relative sentence starting with the pronoun *που* in the complement sentence starting with the conjunction *ότι*. The verb *προχωρώ* in the relative sentence has the frame *#subj_np_nm##advp##pp_se#*. Thus, the module has recognized a PP argument starting with the preposition *σε*. It also identified the relative pronoun *που* as its nominative subject, by checking the respective POS tag (PnReNe03Pl**Nm**Xx) assigned by the tagger.

8. Results

Precision and recall measures have been calculated given the definitions that follow:

$$\text{Precision} = \frac{\text{Correct Identified Instances}}{\text{Total Identified Instances}}$$

$$\text{Recall} = \frac{\text{Correct Identified Instances}}{\text{Total Instances}}$$

concerned, speed of parsing (excluding tokenization and POS tagging) is ~260 words/sec in a 550Mhz PC running Windows 98.

Constituent Type	Precision (Corrected Input)	Precision (Non Corrected Input)	Recall (Corrected Input)	Recall (Non Corrected Input)
adjp_nm	0.95	0.85	0.96	0.78
adjp_ge	0.97	0.91	0.96	0.92
adjp_ac	0.96	0.84	0.97	0.89
np_nm	0.93	0.85	0.93	0.83
np_ge	0.94	0.89	0.94	0.93
np_ac	0.95	0.85	0.95	0.88
advp	0.92	0.85	0.91	0.88
pp	0.87	0.84	0.86	0.81
vg	0.94	0.94	0.97	0.97
vg_s	0.95	0.95	0.90	0.90
vg_g	1.00	1.00	1.00	1.00
cl	0.70	0.64	0.81	0.75
cl_r	0.80	0.79	0.80	0.79
cl_ri	1.00	1.00	0.67	0.67
cl_ir	0.75	0.70	0.78	0.75
cl_c	0.77	0.66	0.77	0.66
cl_t	0.78	0.71	0.84	0.76
cl_o	0.73	0.68	0.74	0.71

Figure 3: Phrase Recognition Performance

Grammatical Relations Type	Precision (Corrected Input)	Precision (Non Corrected Input)	Recall (Corrected Input)	Recall (Non Corrected Input)
Subjects	0.95	0.79	0.75	0.56
Null subjects	0.67	0.53	0.96	0.88
Predicative Phrases	0.81	0.76	0.84	0.63
Direct objects in accusative	0.79	0.62	0.82	0.69
PP arguments	0.76	0.72	0.72	0.64
Dependents in genitive	0.91	0.88	0.92	0.71
Clausal arguments	0.84	0.71	0.8	0.58

Figure 4: Grammatical Relations Recognition Performance

Performance estimations per syntactic category and grammatical function are displayed in Figures 3 and 4. We give precision and recall values for two different configurations. In the first configuration, the output of the POS tagger is manually corrected before it is given to the parser. In the second configuration, no intervention in the pipeline of Figure 1 takes place. So, in the first case we measure the performance of the parsing phase alone, while in the second case we measure the performance of the whole pipeline. As far as speed of analysis is

9. Conclusion

The results obtained so far are encouraging. The accuracy of the output ranges for most cases between 70% and 90%, with phrase recognition performance being the highest. Thus, the parser is suitable for integration in application systems where large scale text processing is needed but no full blown syntactic analysis is necessary. Processing speed, system robustness and relative accuracy of results are the system's strong points. At the moment, efforts are focused on improving clause parsing and

subject/complement recognition. Along these lines, we plan to augment the frame database and examine the possibility of automatically acquiring sub-categorization patterns.

10. References

Abney, S., 1990. Rapid Incremental Parsing with Repair. In *Proceedings of the 6th New OED Conference*, Electronic Text Research.

Abney, S., 1996. Partial Parsing via Finite-State Cascades. In *Proceedings of the Robust Parsing Workshop*, ESSLLI.

Abney, S., 1997. Part of Speech Tagging and Partial Parsing. In *Corpus-Based Methods in Language and Speech Processing*, Steve Young and Gerrit Bloothooft (eds.), Kluwer Academic Publishers, pp. 118-136.

Ait-Mokhtar, S. and J.P. Channod, 1997. Incremental Finite State Parsing. In *Proceedings of ANLP*, pp. 72-79.

Appelt, D. and J. Hobbs, 1995. SRI International FASTUS System - MUC6 Test Results and Analysis. In *Proceedings of MUC6*.

Bourigault, D., 1992. Surface Grammatical Analysis for the Extraction of Terminological Noun Phrases. In *Proceedings of the 14th International Conference on Computational Linguistics*.

Boutsis, S., S. Piperidis and I. Demiros, 1999. Generating Bilingual Lexical Equivalences from Parallel Texts, *Applied Artificial Intelligence.*

Brill, E., 1993a. Transformation-Based Error-Driven Parsing. In *Proceedings of the 3rd International Workshop on Parsing Technologies.*

Brill, E., 1993b. *A Corpus-based Approach to Language Learning*, Doctoral Dissertation, University of Pennsylvania.

Di Christo, P., S. Harie, C. de Loupy, N. Ide, and J. Veronis, 1995. Set of Programs for Segmentation and Lexical Look up, Deliverable 2.2.1, MULTEXT, LRE 62-050.

Evans, D. and C. Zhai, 1996. Noun-Phrase Analysis in Unrestricted Text for Information Retrieval. In *Proceedings of the 34th Annual Meeting of Association for Computational Linguistics.*

Evans, D., N. Milic-Frayling, and R. G. Lefferts, 1995. CLARIT TREC-4 Experiments. In *Proceedings of the 4th Text Retrieval Conference (TREC-4).*

Grefenstette, G., 1996. Light Parsing as Finite State Filtering. In *Proceedings of Workshop on Extended Finite State Models of Language, ECAI*.

Grishman, R., 1995. The NYU System for MUC-6 or Where's the Syntax? In *Proceedings of MUC6.*

Hindle, D., 1983a. *User Manual for Fidditch.* Technical Memorandum #7590-142, Naval Research Laboratory.

Hindle, D., 1983b. Deterministic Parsing of Syntactic Non-Fluences. In *Proceedings of the 21st Annual Meeting of the Association of Computational Linguistics.*

Karlsson, F., A. Voutilainen, J. Hekkila, and A. Anttila, (eds.), 1995. *Constraint Grammar: a Language Independent System for Parsing Unrestricted Text*. Mouton de Gruyter.

Karlsson, F., 1990. Constraint Grammar as a Framework for Parsing Running Text. In *Proceedings of 10th International Conference in Computational Linguistics*.

Karttunnen L., 1997. *The Replace Operator*. In Finite State Language Processing, ed. Roche Em. and Schabes Yv., MIT Press

Leech, G., R. Barnett, and P. Kahrel, 1996. *Provisional Recommendations and Guidelines for the Syntactic Annotation of Corpora*, EAGLES DOCUMENT EAG—TCWG—SASG/1.8.

Marcus, M., 1980. *A Theory of Syntactic Recognition for Natural Language*, MIT Press

Papageorgiou, H, 1996. Part of Speech Disambiguation. In *Hybrid Techniques for Bilingual Corpus Processing*, PhD dissertation, National Technical University of Athens.

Roche, E. and Y. Schabes (eds.), 1997. *Finite State Language Processing*. MIT Press

Satta, G. and E. Brill, 1996. Efficient Transformation-Based Parsing. In *Proceedings of the 34st Annual Meeting of the Association of Computational Linguistics.*

Stralkowski, T. and J. P. Carballo, 1995. Natural Language Information Retrieval: TREC-4 Report. In *Proceedings of the 4th Text Retrieval Conference (TREC-4).*

Van Noord, G. and Gerdemann D., 1999. *An Extendible Regular Expression Compiler for Finite-state Approaches in Natural Language Processing*. WIA, Potsdam, Germany

Voutilainen, A., 1993. NPtool, a Detector of English Noun Phrases. In *Proceedings of the Workshop on Very Large Corpora*.

Zhai, C., 1997. Fast Statistical Parsing of Noun Phrases for Document Indexing. In *Proceedings of the 5th Conference on Applied Natural Language Processing.*

11. Appendix

	)SENT	<S>				
REF<948>	SYN	[cl				
REF<949>	SYN	[np_nm				
REF<950>	TOK		Η	ο	AtDfFeSgNm	atdfsgnm
REF<951>	SYN	[adjp_nm				
REF<952>	TOK		Αγροτική	αγροτικός	AjBaFeSgNm	ajbasgnm
REF<953>	SYN	/adjp_nm]				
REF<954>	TOK		Τράπεζα	τράπεζα	NoCmFeSgNm	nosgnm
REF<955>	SYN	/np_nm]				
REF<956>	SYN	[vg				
REF<957>	TOK		ενημέρωσε	ενημερώνω	VbMnIdPa03SgXxPeAvXx	vb
STRUCT<subj_np_nm,949,955>			STRUCT<cl_arg,971,1010>		STRUCT<compl_np_ac,959,965>	
REF<958>	SYN	/vg]				
REF<959>	SYN	[np_ac				
REF<960>	TOK		τους	ο	AtDfMaPlAc	atdfplac
REF<961>	SYN	[adjp_ac				
REF<962>	TOK		αρμόδιους	αρμόδιος	AjBaMaPlAc	ajbaplac
REF<963>	SYN	/adjp_ac]				
REF<964>	TOK		ερευνητές	ερευνητής	NoCmMaPlAc	noplac
					STRUCT<arg_np_ge,966,969>	
REF<965>	SYN	/np_ac]				
REF<966>	SYN	[np_ge				
REF<967>	TOK		της	ο	AtDfFeSgGe	atdfsgge
REF<968>	TOK		υπόθεσης	υπόθεση	NoCmFeSgGe	nosgge
REF<969>	SYN	/np_ge]				
REF<970>	SYN	/cl]				
REF<971>	SYN	[cl_o				
REF<972>	TOK		ότι	ότι	CjSb	cjsb_other
REF<973>	CHUNK		_	_		
REF<974>	SYN	[vg				
REF<975>	TOK		ανευρέθησαν	ανευρίσκω	VbMnIdPa03PlXxPePvXx	vb
					STRUCT<subj_np_nm,977,980>	
REF<976>	SYN	/vg]				
REF<977>	SYN	[np_nm				
REF<978>	DIG		33	33	DIG	dig
REF<979>	TOK		στελέχη	στέλεχος	NoCmNePlNm	noplnm
REF<980>	SYN	/np_nm]				
REF<981>	SYN	[cl_r				
REF<982>	TOK		που	που	PnReNe03PlNmXx	pn_pou
REF<983>	SYN	[vg				
REF<984>	TOK		είχαν	έχω	VbMnIdPa03PlXxIpAvXx	vb_exw
REF<985>	TOK		προχωρήσει	προχωρώ	VbMnNfXxXxXxXxPeAvXx	vb_inf
			STRUCT<pp_arg,987,995>		STRUCT<subj_np_nm,982,982>	
REF<986>	SYN	/vg]				
REF<987>	SYN	[pp				
REF<988>	TOK		σε	σε	AsPpSp	as_se
REF<989>	SYN	[np_ac				
REF<990>	SYN	[adjp_ac				
REF<991>	TOK		εικονικές	εικονικός	AjBaFePlAc	ajbaplac
REF<992>	SYN	/adjp_ac]				
REF<993>	TOK		προεγγραφές	προεγγραφή	NoCmFePlAc	noplac
REF<994>	SYN	/np_ac]				
REF<995>	SYN	/pp]				
REF<996>	SYN	/cl_r]				
REF<997>	SYN	/cl_o]				
REF<998>	PTERM_P		.	.	PTERM_P	punct_fs
	)SENT	</S>				