Multi-level XML-based Corpus Annotation

Harris Papageorgiou, Prokopis Prokopidis, Voula Giouli, Iason Demiros, Alexis Konstantinidis, Stelios Piperidis

Institute for Language and Speech Processing
Epidavrou & Artemidos 6, 151 25 Maroussi, Greece
{xaris, prokopis, voula, iason, akonsta, spip}@ilsp.gr

Abstract

In this paper, we present the methodological principles and the implementation framework of text annotation process in an Information Extraction setting. Due to the recent prevalence of XML as a means for describing structured documents in a reusable format, our team has switched to an XML based annotation schema. In that framework, an XML annotation platform has been built, while processing tools, lexical resources and textual data communicate with each other via this platform. Editing/viewing tools have been implemented, endowed with functionalities that allow annotators to gain access to previous annotation levels as well as necessary lexical resources.

1. Introduction

Annotated corpora have proven to be of great value for several tasks, including training and testing of tools for the automatic analysis of language at different levels. In this paper we present completed and ongoing work carried out at the Language Technology Applications Department of the Institute for Language and Speech Processing, where miscellaneous tools have been used for the annotation of Greek texts, while building resources for an Information Extraction chain.

This paper is organized as follows. In section 2 we provide some initial remarks on our corpus collection, regarding annotation and use. In the following section, we present the general architecture of our IE chain, together with relevant resources developed for each processing stage. We also present new levels of annotation that we have recently started to implement. In section 4, after commenting on problems arising from the annotation format we have been using so far, we present an XML based architecture for annotation and automatic processing of corpora. A multilevel annotation tool is included in this environment, providing a series of facilities to annotators.

2. Corpus Description

We maintain (and regularly augment) a large collection of Greek corpora, annotated at different levels (until recently, only as far as lexical and sentential boundaries, morphosyntactic information and named entities are concerned). Although the majority of our corpus originates from technical and financial online sources, our data collection includes documents from other genres as well, such as texts from the legislative and sports domains, governmental announcements, interviews, dialogue transcriptions etc.

For each annotation level, initial guidelines are provided by the linguists that perform each annotation task. After a brief testing period, samples by all members of the team of annotators are collected and inter-annotator agreement is examined. The guidelines are further augmented with cases that linguists consider exceptional, or where systematic inter-annotator disagreement has been observed.

We have decided to proceed to the annotation of texts at more abstract linguistic levels, including annotation of recursive phrasal constituents, grammatical relations, terms, and intra/intersentential coreference.

3. IE system architecture

The aforementioned corpus collection is mainly used for training and testing of a chain of tools developed in an Information Extraction environment. The main modules of this pipeline include a tokenizer, a POS tagger and a lemmatizer, together with tools that recognize named entities and non-recursive syntactic units, by using grammars compiled into finite state transducers. Modules recognizing recursive syntactic structures, grammatical relations and coreferential links have also been developed. All processing modules are operating-system independent and, until recently, shared a common Tipster-like (Grishman, 1997) data model.

In the following subsections, we briefly present our IE pipeline, together with the textual data that were collected and annotated for training and testing of each module.

3.1. Tokenization

At the initial stage of our IE chain, basic text handling is performed by a MULTEXT-like tokenizer (Di Cristo, 1995) that identifies word and sentence boundaries, abbreviations, digits, and simple dates. Following common practice, the tokenizer makes use of a regular-expression engine, coupled with precompiled lists of abbreviations amounting to 150 entries, and a set of simple heuristics.

The tokenizer has been tested on a corpus of approximately 400K words, manually annotated at the level of lexical and sentential boundaries. The texts belong to a variety of domains covering a wide range of linguistic phenomena and textual structure. They are EU bureaucratic documents, texts from newspapers and magazines, tourist guides and computer manuals. The tool has been found to be quite successful in effectively recognizing sentences and words, with accuracy of up to 95%.

3.2. POS Tagging and Lemmatization

To assign morphosyntactic information to words of a tokenized text, we use a tagger that is based on Brill's

TBL architecture (Brill, 1997), modified to address peculiarities of the Greek language (Papageorgiou et al., 2000). We use a PAROLE (Lambropoulou et al., 1996) compliant tagset of 584 different part-of-speech tags.

Our tagger assigns initial tags by following simple heuristics and looking up words in a precompiled lexicon. For unknown words, we use lexicons of suffix-tag combinations, assuming that enough morphological information is encoded in suffixes of words, due to the rich inflectional system of the Greek language. 799 contextual rules (divided in 4 groups, according to the grammatical features on which they focus) are then applied to improve word and suffix lexicons output.

These rules were automatically acquired from a manually annotated corpus of 210 files from different genres of texts, which amounted to a total size of ca. 447K words. Special attention was paid to the overall balance of the corpus, which was composed of texts from different domains, ranging from financial newswires to political press conferences, and from interviews to computer hardware test reports.

For the creation of this corpus, two linguists worked in parallel for a period of three months, using a graphical tool that was implemented in Tcl/Tk, correcting the output of a previous version of the tagger. They followed guidelines already set in previous work during corpus annotation at ILSP. There was an attempt to augment, clarify and formalize these instructions. Inter-annotator consistency was addressed by having a number of files corrected by both linguists, thus allowing for identification and resolution of discrepancies. Moreover, certain tags, allowing for rare cases such as datives of pronoun forms or numerals, were added to our tagset during the annotation process.

By reserving 20% of the annotated data as a testing corpus, we measured the accuracy of the tagger to be approximately 90% when all features are examined, and around 96% when only POS information is taken into account.

Following POS tagging, lemmas are retrieved from ILSP's Greek morphological lexicon. This resource contains 66K lemmas, which in their expanded form extend the lexicon to approximately 2,000,000 different entries.

3.3. Named Entity Recognition

Specifying the annotation schema for a NE recognition module (Boutsis et al., 2000a), we followed the MUC-7 guidelines (Chinchor, 1997). In particular, we cater for the identification of NE's of types ENAMEX (PERSON, ORGANIZATION, LOCATION), TIMEX (TIME and DATE) and NUMEX (MONEY, PERCENT).

A corpus of Greek texts, comprising articles from financial newspapers, magazines and financial portals was downloaded from the web. The selected articles formed the training and testing corpus, which amounted to ca. 150K words. This corpus was then manually annotated according to our annotation schema. A Tcl/Tk GUI facilitated manual annotation of NE's in the text. The annotated corpus was used for both development and evaluation: 130K words were used to guide system development, e.g. evaluate rule performance, while the remaining 20K words were reserved for testing purposes, showing results of 86% precision and 81% recall.

Automatic recognition of NE's is performed in two steps, i.e. Name Lookup and NE Parsing based on an NE grammar. During the first step, a set of static pre-stored names and regular expressions are matched against the tokenized, tagged, and lemmatized text in order to identify known named entities and trigger words. We compiled lists of person, organization, and location names, combining material from several different sources. The name lists were also enhanced with names extracted from the 130K words of the manually annotated text. After all additions, the company name list had 1,059 entries, the location name list 793 entries, and the person name list 1,496 entries. Name lookup is implemented on the basis of finite state recognizers, scanning the text at high speed and searching for strings and regular expressions appearing in the name lists.

During the second step of the named entity recognition module, rules are applied to the output of the name lookup stage to finalize named entity typing, as well as to recognize names not in the lists. Rules operate on the basis of names recognized at the lookup stage, capitalization information and POS tags.

Rules are written in the form of regular expressions, compiled into finite state transducers that transform input text by inserting or removing special markers. We make use of the FSA6 package (Van Noord and Gerdemann, 1999) for compiling rules into FST's. The grammar consists of 110 rules in total: 17 for persons, 19 for locations, 37 for organizations, 23 for dates, 5 for times, 7 for money and 2 for percentages.

3.4. Shallow Syntactic Analysis

To allow for shallow syntactic analysis of Greek text, a grammar (Boutsis et al., 2000b) has been developed for the automatic recognition of the following non recursive phrasal categories: adjectival, adverbial, prepositional, nominal, and verbal chunks. Main clauses and several types of subordinate clauses are also recognized, mainly on the basis of trigger words. The grammar consists of 186 rules that are compiled into FST's as in the NE recognition task. Again, the FSA6 package is being used.

Two linguists have used a Java based interface to annotate a text collection composed of news and financial articles for the evaluation of the grammar. A number of files were annotated by both linguists to ensure inter-annotator consistency. The total size of the collection amounted to ca. 39K tokens, while precision and recall typically ranged between 75% and 95%, with performance on chunk boundaries identification being the highest.

3.5. Recursive Phrasal Constituents

Moving on to a deeper syntactic analysis of Greek text, we have also annotated the corpus mentioned in 3.4, as far as recursive nominal elements are concerned. These NP elements presuppose syntactic annotation at chunk and clause level. In most cases they comprise a nominal chunk that may be modified by nominal chunks in genitive, prepositional phrases and/or relative clauses. Coordinated nominal chunks and nominal chunks in appositive constructions are also annotated as NP's.

For automatic syntactic analysis at this level, we have developed a grammar that comprises 23 groups of rules, which, compiled into FST's, recognize a subset of the NP's identified by human annotators, mainly those with

nominal chunks modified by relative clauses and/or nominal chunks in genitive case.

3.6. Grammatical relations

For the functional annotation of Greek texts, i.e. the identification of grammatical relations between syntactic units, our annotation schema followed MATE codification (Dybkjaer et al., 1998). Grammatical roles that are identified and annotated include subjects, predicative complements, direct and indirect objects, prepositional phrases functioning as arguments or modifiers, and clausal arguments.

In parallel, a module responsible for the automatic identification of grammatical relations has been developed on the basis of a pattern matching mechanism. The main resource used at this stage is a subcategorization frames lexicon. The entries have been retrieved from a database containing subcategorization information for the 5927 most frequent verbs, 4950 most frequent nouns, and 375 most frequent adjectives of a general purpose corpus.

3.7. Coreference resolution

From the broad set of referential phenomena that characterize Greek language, we have focused on pronominal anaphora resolution. The task is to resolve anaphors that have definite descriptions as their antecedents. The pronoun types that were selected for annotation are the 3rd person possessive and the relative pronoun. Two forms of anaphora are covered: intrasentential, where coreferring expressions occur in the same sentence, and intersentential, where the pronoun refers to an entity mentioned in a previous sentence.

The coreference resolution component is based on the work of Lappin and Leass (1994). The algorithm employs a simple weighting scheme integrating the effects of recency features and syntactically-based preferences. No semantic preferences are employed beyond those enforced by agreement.

The testing corpus of the coreference resolution module was identical to the corpus used for named entity recognition, which amounts to 150K words.

3.8. Term annotation

In the framework of the Cimwos project (CIMWOS, 2002), linguists in our group are currently involved in annotation of terms in transcriptions of Greek news broadcasts. Terms are also associated with a hierarchical list of topics initially compiled by Reuters, and augmented on an as needed basis.

A term detection module has been developed to identify terms in Greek text. It is a hybrid system comprising a regular expression-based term pattern grammar, and a statistical filter, used for the removal of grammar-extracted terms lacking statistical evidence.

4. XML architecture

In the past, metadata for the annotation phases described in subsections 3.1-3.4 were stored in plain text files that followed a Tipster compatible format. For each document in our collection, we had a series of files, with information from previous stages of annotation encoded in separate columns. Although this format allowed for a basic overview of the annotations, files were generally difficult to alter and maintain. Any change, for example, in the morphosyntactic tag assigned to a word had to be tracked down and changed to all files of subsequent annotation levels. Moreover, different annotation tools (Java and Tcl/Tk based) had been developed for each level. As a result, annotators had to familiarize themselves with the different interfaces of each tool.

Recently and due to the prevalence of XML as a means for describing structured documents in a reusable format, our team has decided switching to an XML based annotation platform.

This migration involved:

- converting the annotation files to XML documents whose structure is imposed by appropriate DTD's
- converting tools and resources so that they accept input and produce output in XML
- developing a multi-level annotation environment for viewing and editing XML annotation metadata

For each document in our corpus, a master file is created, containing sentence and word boundaries information, together with a POS tag for each token. In the example of Figure 1 of the Appendix, sentence boundaries are represented as <sent> elements to which a unique *id* number is assigned. Moreover, start and end offsets of the sentences in the original text are also encoded here. Subsequent annotations can enlarge nested information in the <sent> elements, by referring to them using standard XML linking mechanisms. Morphological words are described at this level as <mw> elements with an *id* relevant to the sentence into which they appear. The *tag* attribute encodes basic POS category, as well as further morphosyntactic information of considerable length and granularity.

Annotations at other levels point to the basic elements described in the master file, which, as a convention, has the extension *morph.xml*. For example, a file with functional annotation information like the one in Figure 2 includes entries which point to the *.morph.xml* document. Each grammatical relation is depicted as a <funct> element that contains one or more <dep> elements. Each of them represents a dependent to the <head> element, which in its turn is also a child element of <funct>. Different types of functional relations are specified as different values of the attribute *type* of the <dep> element. In the example, two dependents, i.e. a subject and a direct object are described as <dep> *3_1_1* and *3_1_2*, respectively. Head and dependent boundaries, as well as sentence numbers refer to the master file.

As we said above, different levels of annotation are stored in separate files. Nevertheless, merging different levels into one XML file is also an option.

We are currently working on converting all tools of our IE chain so that they are compliant to the XML structure of the annotation metadata. Moreover, the well-formedness of most of our resources, such as the lexicon with the subcategorization frames of 3.6, or the topics list mentioned in 3.8, is now ensured by predefined DTD's.

Based on the flexibility of this approach, after modifying the tool that automatically identifies grammatical relations, we can now process one or more files and reload the tool's output in the annotation environment described in section 5. This way we can inspect changes or monitor the tool's performance, by

comparing its output with gold data produced by human annotators.

5. Marker

An important component of our XML architecture is an annotation environment called *Marker*. Marker is a GUI that allows annotators to have simultaneous views of all levels of previous annotations, while working at a particular task. Furthermore, it is equipped with comparison facilities that allow for inspection of inter-annotator agreement or tool performance, expressed in precision and recall measures. The environment currently supports annotation at the morphosyntactic level, chunk and recursive phrases level, NE, term and coreference annotation, and annotation of grammatical relations. The tool runs on any PC or workstation equipped with a recent version of Sun's Java 2 Runtime Environment and is available free of charge for research purposes.

We can examine some of the functionalities the tool provides, in the screenshot of Figure 3, where the example sentence "Η υλοποίηση της απόφασης προϋποθέτει την αναζήτηση ενός στρατηγικού επενδυτή" ("The implementation of the decision presupposes searching for a strategic investor") is being annotated at the grammatical relations level. Accepting input from the modules described in subsections 3.1 to 3.5, the pattern matching tool of 3.6 consults the subcategorization lexicon for the lemma "προϋποθέτω" and retrieves information on suitable arguments of the verb. Then, after scanning the sentence for syntactic units fullfiling the constraints imposed by the frame, the tool identifies two dependents for "προϋποθέτει", a subject ("η υλοποίηση της απόφασης") and a direct object ("την αναζήτηση ενός στρατηγικού επενδυτή"), which are realized as a nominative and an accusative NP, respectively.

When viewing the output of the tool via the Marker, all previous annotations are automatically loaded as well. The internal structure of the dependent NP's in our example can be further examined by clicking on the *Chunk* tab. Both NP's will then be represented as two nominal chunks, i.e. "η υλοποίηση" and "την αναζήτηση", modified by nominal chunks in genitive, i.e. "της απόφασης" and "ενός στρατηγικού επενδυτή". The two grammatical roles are represented in different, user-defined colors on the sentence box at the upper part of the Marker. Human annotators have the option of selecting constituents from the tree boxes on the left, identifying them as heads and dependents of relations, thus correcting the output of the pattern matching mechanism.

As another example of the GUI's versatility, the tool encompasses an editor, which allows users to edit the topic list mentioned in 3.8, in the framework of term annotation. Annotators can augment the list with more specific topics and/or rearrange nodes in the topic hierarchy.

Mapping the output of the POS tagger on chunk/phrase trees, e.g. the tag *NoCmFeSgAc* for "αναζήτηση" in our example, is optional and performed online. The environment also provides a text box for the creation and editing of comments, which are stored inside the metadata files, as child elements of the relative <sent> element. Other options include choice of annotation level, expanding and collapsing trees (when editing large sentences), etc. All preferences are stored in user-profile files and can be retrieved each time the tool is run.

Moreover, session metadata are stored separately from annotation data. These metadata elements are inspired by the Dublin Core Metadata Initiative (DCMI) standard, including among others, *Annotator* (an entity responsible for providing the annotation content), *Subject* (what the annotation is about), *Resources* (the resources that have been used in the annotation session), *Language* and *Date* (a date associated with the current session). Subsequent modifications/reviews by the same or other annotators are also kept in the session metadata files.

Classes of XML annotations that share a common vocabulary and structure (morphology, syntax, etc.) are described in DTD's. The Marker looks for the relevant DTD when initiating an annotation session and configures the GUI appropriately by providing the needed functionality to the annotator. This dynamic process of building and customising a GUI on the fly (based on external DTD files) is currently restricted to simple elementary structures which however fulfill most of our current annotation needs. Additionally, a validation step is being performed ensuring that a particular instance is compliant with the prespecified constraints in the DTD's.

6. References

Boutsis, S., I. Demiros, V. Giouli, M. Liakata, H. Papageorgiou and S. Piperidis. (2000a). A System for Recognition of Named Entities in Greek. In *Proceedings of Natural Language Processing 2000*, (pp. 424-436), Patras, Greece.

Boutsis, S., P. Prokopidis, V. Giouli and S. Piperidis. (2000b). A Robust Parser for Unrestricted Greek Text. In *Proceedings of the 2nd Language and Resources Evaluation Conference*, (pp. 467-473), Athens, Greece.

Brill, E. (1997). A Corpus-based Approach to Language Learning. PhD Thesis, University of Pennsylvania.

CIMWOS (2002). Combined IMage and Word Spotting. IST project. http://www.xanthi.ilsp.gr/cimwos/

Chinchor, N. (1997.) MUC-7 Named Entity Task Definition, Version 3.5

DCMI. Dublic Core Metadata Initiative. http://www.dublincore. org/documents/

Di Cristo, P., S. Harie, C. de Loupy, N. Ide, and J. Veronis. (1995). Set of programs for segmentation and lexical look up, MULTEXT LRE 62-050 project Deliverable 2.2.1

Grishman, R. (1997). Tipster architecture design document version 2.3. Technical report, DARPA.

Lambropoulou, P., E. Mantzari, and M. Gavriilidou. (1996). Lexicon - Morphosyntactic Specifications: Language Specific Instantiation (Greek), PP-PAROLE, MLAP 63-386 report.

Lappin, S. and H. J. Leass. (1994). An Algorithm for Pronominal Anaphora Resolution. In *Computational Linguistics*, 20 (4), pp: 535-561.

Dybkjaer, L., N. O. Bernsen, H. Dybkjaer, D. McKelvie and A. Mengel. (1998). The MATE Markup Framework, MATE Deliverable D1.2.

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

Papageorgiou, H., P. Prokopidis, V. Giouli and S. Piperidis. (2000). A Unified POS Tagging Architecture and its Application to Greek. In *Proceedings of the 2nd Language and Resources Evaluation Conference* (pp 1455-1462), Athens, Greece.

7. Appendix: Sample Annotations and Marker Screenshot

```
<?xml version="1.0" encoding="ISO-8859-7" ?>
<Annotation type="morpho">
…
<sent id="s_3" start="111" end="187">
<mw id="mw_3_1" lex="Η" tag="AtDfFeSgNm" lemma="ο" start="0" end="1"></mw>
<mw id="mw_3_2" lex="υλοποίηση" tag="NoCmFeSgNm" lemma="υλοποίηση" start="2" end="11"></mw>
<mw id="mw_3_3" lex="της" tag="AtDfFeSgGe" lemma="ο" start="12" end="15"></mw>
<mw id="mw_3_4" lex="απόφασης" tag="NoCmFeSgGe" lemma="απόφαση" start="16" end="24"></mw>
<mw id="mw_3_5" lex="προϋποθέτει" tag="VbMnIdPr03SgXxIpAvXx" lemma="προϋποθέτω" start="25" end="36"></mw>
<mw id="mw_3_6" lex="την" tag="AtDfFeSgAc" lemma="ο" start="37" end="40"></mw>
<mw id="mw_3_7" lex="αναζήτηση" tag="NoCmFeSgAc" lemma="αναζήτηση" start="41" end="50"></mw>
<mw id="mw_3_8" lex="ενός" tag="AtIdMaSgGe" lemma="ένας" start="51" end="55"></mw>
<mw id="mw_3_9" lex="στρατηγικού" tag="AjBaMaSgGe" lemma="στρατηγικός" start="56" end="67"></mw>
<mw id="mw_3_10" lex="επενδυτή" tag="NoCmMaSgGe" lemma="επενδυτής" start="68" end="76"></mw>
</sent>
<sent id="s_4" start="189" end="264">
<mw id="mw_4_1" lex="«" tag="PUNCT" lemma="«" start="0" end="1"></mw>
<mw id="mw_4_2" lex="Στρατηγικό" tag="AjBaMaSgAc" lemma="στρατηγικός" start="1" end="11"></mw>
<mw id="mw_4_3" lex="επενδυτή" tag="NoCmMaSgAc" lemma="επενδυτής" start="12" end="20"></mw>
<mw id="mw_4_4" lex="»" tag="PUNCT" lemma="»" start="20" end="21"></mw>
<mw id="mw_4_5" lex="για" tag="AsPpSp" lemma="για" start="22" end="25"></mw>
<mw id="mw_4_6" lex="την" tag="AtDfFeSgAc" lemma="ο" start="26" end="29"></mw>
<mw id="mw_4_7" lex="επέκταση" tag="NoCmFeSgAc" lemma="επέκταση" start="30" end="38"></mw>
<mw id="mw_4_8" lex="στις" tag="AsPpPaFePlAc" lemma="στου" start="39" end="43"></mw>
<mw id="mw_4_9" lex="τηλεπικοινωνίες" tag="NoCmFePlAc" lemma="τηλεπικοινωνία" start="44" end="59"></mw>
<mw id="mw_4_10" lex="αναζητεί" tag="VbMnIdPr03SgXxIpAvXx" lemma="αναζητώ" start="60" end="68"></mw>
<mw id="mw_4_11" lex="η" tag="AtDfFeSgNm" lemma="ο" start="69" end="70"></mw>
<mw id="mw_4_12" lex="ΔΕΗ" tag="ABBR" lemma="ΔΕΗ" start="71" end="74"></mw>
<mw id="mw_4_13" lex="." tag="PUNCT" lemma="." start="74" end="75"></mw>
</sent>
…
</Annotation>
```

Figure 1. Master XML file containing morphological information

```
<?xml version="1.0" encoding="UTF-8"?>
<Annotation type="functional_verb">
…
<sent id="s_3">
 <funct id="3_1" type="SFVb">
  <head start="25" end="36" />
  <dep id="3_1_1" start="0" end="24" type="subj" />
  <dep id="3_1_2" start="37" end="76" type="dobj" />
 </funct>
</sent>
…
</Annotation>
```

Figure 2. Functional annotation sample

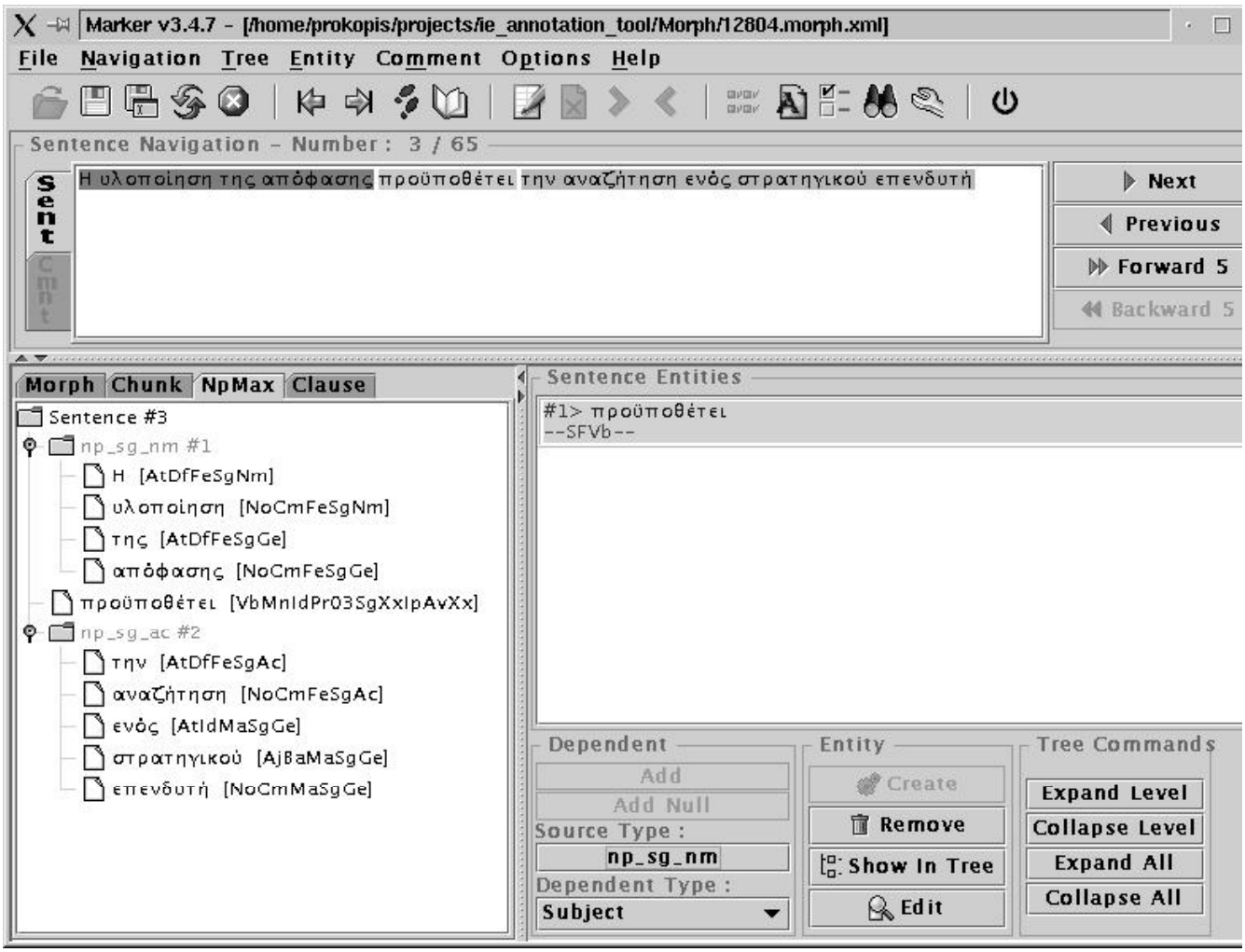

Figure 3. Marker Screenshot