Episimiotis:
A Multilingual Tool for Hierarchical Annotation of Texts

Ilias Koutsis,[1] George Markopoulos[1]
and George Mikros[2]

1. Introduction

Linguistic annotation is crucial for the development and evaluation of natural language processing tools. Machine-learning based approaches to part-of-speech tagging, word sense disambiguation, information extraction or anaphora resolution - just to name a few - rely on annotated corpora.

Furthermore, entering the new era of the Semantic Web, there is a growing need for the construction of large corpora of hierarchically annotated texts and the reusability of annotated language resources.

"Episimiotis" is a tool for annotating a complex hierarchical and linguistic structure of any text. It was primarily designed for the tagging and analysis of errors made in written assessments by students of Modern Greek as foreign language by means of a predefined tagset. However, its scope can be extended to any type of detailed structure annotation using different annotation schemes via a user-friendly interface.

Linguistic annotation in texts is essential for the study of language and the development of NLP tools. Annotation can help researcher to identify and examine a large variety of linguistic phenomena and structures such as:

- Errors (e.g. wrong use of parts of the speech, syntactical errors, orthographical errors)
- Parts of the speech
- Rhetorical structures
- Name entities etc

2. Related work

There are many tools for linguistic data annotation. Some of them are concentrated in textual annotation while others can handle multimedia or even multimodal annotation.

MMAX2 is a Java open-source tool by which a user can annotate texts. MMAX2 supports arbitrarily many levels of annotation by means of stand-off annotation. This means that for each file to be annotated the program creates as many files of annotations as the levels of annotation. The program doesn't change anything in the base file that is being annotated but the annotations of each level reside in a separate file. The user is able to arbitrarily define, annotate and browse many relations between markables (both intra- and inter-level). MMAX has two forms of displaying

[1] Department of Linguistics, University of Athens
e-mail: ilias_k@yahoo.com, gmarkop@phil.uoa.gr
[2] Department of Italian and Spanish Language and Literature, University of Athens
e-mail: gmikros@isll.uoa.gr

the annotations. The first form is to display the text with the annotations. The second form of display is to show both the base text file as well as all the annotations.

Marker is a Java GUI that allows annotators to have simultaneous views of all levels of previous annotations while working on a particular task. Nested structures can be marked and viewed, whereas a comment facility allows users to add notes relevant to the markup task at the given point. The tool is further equipped with comparison facilities that allow for inter-annotator agreement and assessment of the relevant processing tools. 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.

Falko (which stands for ***F****ehler****a****nnotiertes* ***L****erner****k****orpus* 'error-annotated learner corpus') (Lüdeling Anke et al., 2005) consists of a growing corpus and contains several distinct sets of data. Each text is annotated with detailed header information so that the data sets can be combined to sub-corpora according to the needs of the researcher. The core corpus is a highly controlled set of summaries of academic texts written by advanced GFL learners (henceforth L2) and native German speakers (henceforth L1) under comparable conditions. Falko's tagging scheme currently uses the following linguistic levels: orthography, word formation, agreement, government, tense, mood, word order, and expression, which codes errors of appropriacy. Each linguistic level consists of at least three sub-levels, structured according to a pattern widely used in second language acquisition research: identification, description, and explanation.

3. Tool presentation

3.1 Tool design

«Episimiotis» is a tool which provides an hierarchical multi-level ability of annotation. Using "Episimiotis", each user can easily annotate a large number of texts using multiple levels of annotation.

The complex hierarchical and linguistic structure for the annotation of any text is performed through the categorisation of annotations in three levels of analysis as well as with the existence of an additional level which is parallel to the three levels of annotation although it is independent of them.

Particularly important is that the annotation tag-set (set for any kind of annotation) can be changed. Through a user friendly interface every user is able to import a tag-set in "Episimiotis" in order to perform a certain text annotation. "Episimiotis" can handle many tag-set and the user import them easily in the annotation interface. The use of "Episimiotis" is not limited by its initial design (error annotation), but can be extended to any type of detailed annotation of linguistic structures using different levels of annotations.

"Episimiotis" uses from one up to three levels depending on the annotation type that the user wants to use. These hierarchical levels of annotation can be combined with an additional independent annotation level. The user is not obliged to use all the three hierarchical levels. He can use as many levels from the three available as he/she needs (one, two or even three). Moreover, he can also use the independent level of annotation depending on the kind of annotation he/she wants to perform.

Two examples with various levels of annotation are the following:

Example 1: Using one level of annotation
<Part of Speech>_</Part of Speech>

In this annotation example the user annotates only the parts of speech. In other words the user is given only one available level of annotation through which he/she is able to annotate the part of speech of any word or phrase in the text.

Example 2: Using either three or two levels of annotation
Example 2.1.
<Verb> <SimplePresent> <Singular> _ </Singular>
</SimplePresent></Verb>
The three levels of annotation are:
First level: The part of speech (verb)
Second level: The tense of the verb (Simple present)
Third level: The form of the verb (Singular)

Example 2.2. <Article> <Singular>_</Singular></Article>
The two levels of annotation are:
First level: The part of speech (article)
Second level: The form of the article (Singular)

In the above example the user uses all three levels of annotation (due to the case of example 2.1). Depending on the part of speech there is a possibility of selecting three levels of annotation (the part of speech, the tense and the person for each verb) or less levels of annotation (e.g. the part of speech and the form for each word which he/she would annotate as article). Substantially, a tree with an available number of annotations is assigned to each tag-set.

An example which refers to a hypothetical case of annotation is displayed below in a tree form:

Verb
- Simple Present
  - Singular
  - Plural
- Present Continuous
  - Singular
  - Plural
- Simple Past
  - Singular
  - Plural
- Past Continuous
  - Singular
  - Plural

Noun
- Male
  - Singular
  - Plural

Female
Singular
Plural
…
Article
Singular
Plural

3.2 Initial use of "Episimiotis" – Error Annotation

Initially, "Episimiotis" was used for error analysis in texts of foreigners who learn Modern Greek as a foreign language.

The error analysis model of "Episimiotis" is based on James (1998), as well as Granger (2003). The database which accompanies "Episimiotis" contains by default the error analysis tag-set modified in order to capture the peculiarities of Modern Greek.

The selected tag-set has descriptive and hierarchical character. It is descriptive because it does not proceed to the interpretation of error, but simply describes it classifying it in categories based on specific criteria. It is hierarchical because each annotation can by further subdivided in more specific error categories that can be represented in tree form.

The error classification is twofold. It is based on the major linguistic categories (linguistic levels, parts of speech) while at the same time uses annotations for the type of divergence from the learning language (erroneous choice, removal, pleonastic use, erroneous order etc). Particular attention was given to the grammatical errors. A special sub-categorization was created based on the linguistic units (morpheme, phrase, sentence etc) in which the errors were located.

As far as the first level error categories are concerned, there are nine codes:

Morphological errors: at the level of grammatical morpheme (**M**ορφολογικά)
Orthographical errors: at the level of linguistic form (**O**ρθογραφικά)
Phrasal errors: at the level of lexical phrase (**Φ**ραστικά)
Errors of a sentence: at the level of the sentence (**Π**ροτασιακά)
Errors of many sentences: at the level of many sentences (Δ**Ι**απροτασιακά)
Errors of a word: at the level of the word (**Λ**εξικά)
Errors of part of speech: at the level of textual type (**Ε**ίδους λόγου λάθη)
Errors of style (**Υ**φους λάθη)
Errors of hyphenation (Στίξης λάθη)

These codes are followed by one or more sub-codes (errors categories) that provide further information on the type of error. For example, first sub-code at the level of **Morphological** errors (**M**ορφολογικά) is the **Form** (**API**θμό) of names and verbs, while there are also other sub-codes, which annotate divergences in the **Gender** (**ΓΕΝ**ος) and **Case (ΠΤΩ**ση) of names, in the **Tense** (**ΧΡ**ό**Ν**ο (XPN)) and **Voice** (**ΦΩΝ**ή) of the verb etc.

After each sub-code, other sub-codes are used. These new sub-codes annotate the type of error. For each level of errors certain sub-codes are used and they are

common in all categories of error at each level of error. For example, every error category at the **Morphological** level of errors is followed by either the code **Deletion** (**ΑΠ**άλειψης (ΑΠ)), or by the code **Erroneous Choice** (**Λ**ανθασμένης **Ε**πιλογής (ΛΕ)), or by the code **Pleonastic Use** (**Π**λεοναστικής **Χ**ρήσης (ΠΧ)), while the error categories of **Part of Speech** are followed only by the code **Erroneous Choice** (**Λ**ανθασμένης **Ε**πιλογής (ΛΕ)). Each error annotation is completed with the addition of a code for the part of speech which the erroneous linguistic type belongs to.

3.3. Functions

3.3.1 Tag-set management

Through the interface of tag-set management (see figure 1) a user who imports a new tag-set determines the way that the rest of the users of the software (that will use that certain tag-set) will work. When a user imports a tag-set, he/she has the ability of setting rules concerning the operation of this tag-set. These rules are determined during the import of the tag-set in the "Episimiotis" database.

The user has two ways of importing the tag-set in the "Episimiotis" database. The first one is by the automatic import of the tag-set through a file of predetermined configuration. The second one is by manual entry of the tag-set from the keyboard. When the user imports it in the database through the tag-set management interface, he/she should determine:

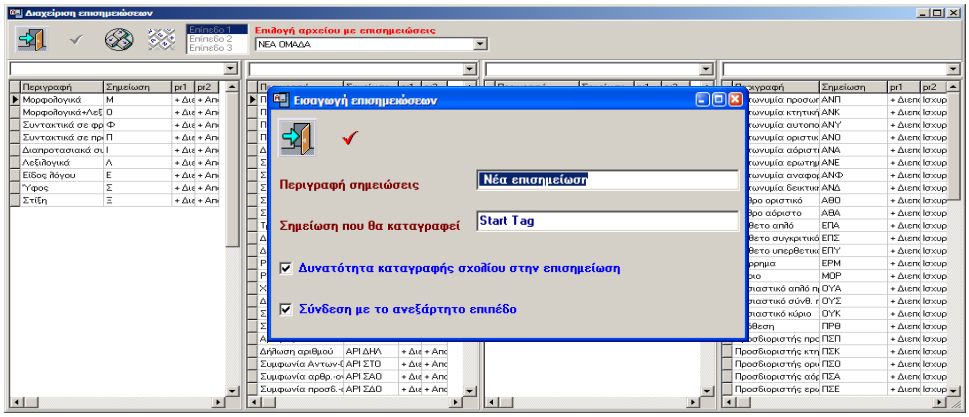

Figure 1: Tag-set creation.

- The number of the hierarchical levels which will be imported.
- Whether the user of the tool will use the independent (not connected to the other hierarchical ones) fourth level separately in each of those hierarchical levels.
- Whether the user of the tool will type the correct text or the comment (either in the case of errors annotation or in the case of simple annotation respectively).

The hierarchies, as well as the above rules that the user-administrator records, are stored in the database of the tools and are readily available to each user.

Each user can access only the specific tags he has predefined. In this way the work of each user will be much easier and a lot of mistakes will be avoided. For example, in the errors annotation, it is possible that the correct text in some annotations to be redundant because of the error type. In this case and at the phase of the tag-set import, the user can select the "correct text" choice to be absent in the user interface.

During the annotation the user is given the opportunity to see on the same screen the annotated text, the source text and all the annotations that are imported. The display of the annotated text together with the source text are determined by the user's appropriate selection in the interface for each of the two above mentioned choices. In the bottom part of the interface all the annotations are presented in a list independently whether the user manipulates the source text or the annotated one. From the list of annotations the user is provided with the ability of managing all imported annotations and he/she can alter them or even delete some of them.

"Episimiotis" has been developed in Delphi. As far as the user interface is concerned, it was given particular importance so that the user is occupied only with the work of annotation and not in learning how to use the software. The user interacts with the software through functional interfaces ergonomically drawn for more complete interactivity between user and software. Facilities that are provided in all the modern softwares are offered such as activation of menus choices using the right key of mouse, hint appearance in most of the control buttons of the interfaces, user profiles in the interfaces determined by each user etc.

3.3.2 Tag management (insert/edit/delete tags)

Each user has the ability of importing up to four levels of annotation by only two mouse clicks. "Episimiotis" also enable the user to store all the annotations in an xls file for further process.

"Episimiotis" provide extensive annotation management (see figure 2). More specifically the user can:

- Display all the annotations one by one.
- Display the attributes of every imported annotation.
- Delete any of the already imported annotations by selecting the correct choice from a menu.
- Import new annotations.

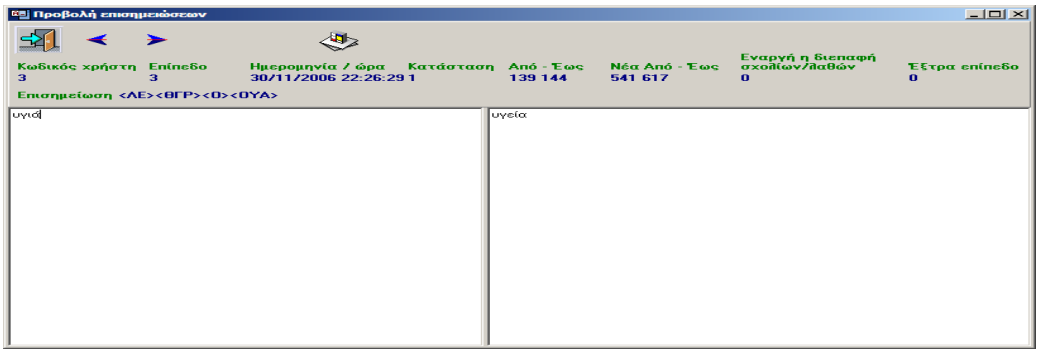

Figure 2: Tag management.

The tool is designed in such a way that the user is not forced to check the correctness of the annotation in the annotated text. The annotations are imported to the original text file through a visual interface. They are imported automatically when the user selects an annotation from the menu and without typing any text. The user types text only when he/she uses the program for error annotation (he/she may enter the correct part of text that substitutes the error in the original text file)

The process of annotation is as simple as follows:

- The user selects the text which he wants to annotate (sentence, word, part of word) (see figure 3)
- With the use of the right button of the mouse he activates the menus choices.

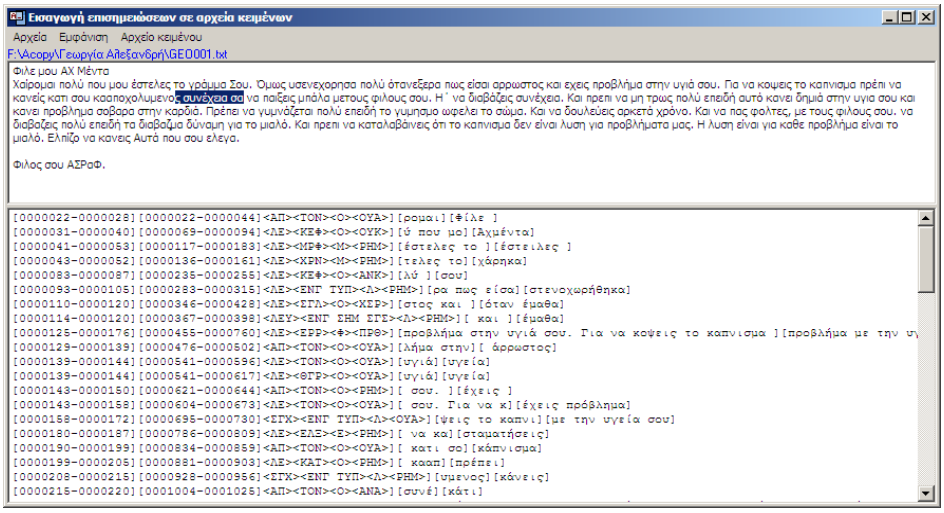

Figure 3: Insert a tag (step1 select text with error)

- From the menu choices he/she selects the choice for the import of the annotation (alternatively he/she can use the key Insert from the keyboard) (see figure 4).

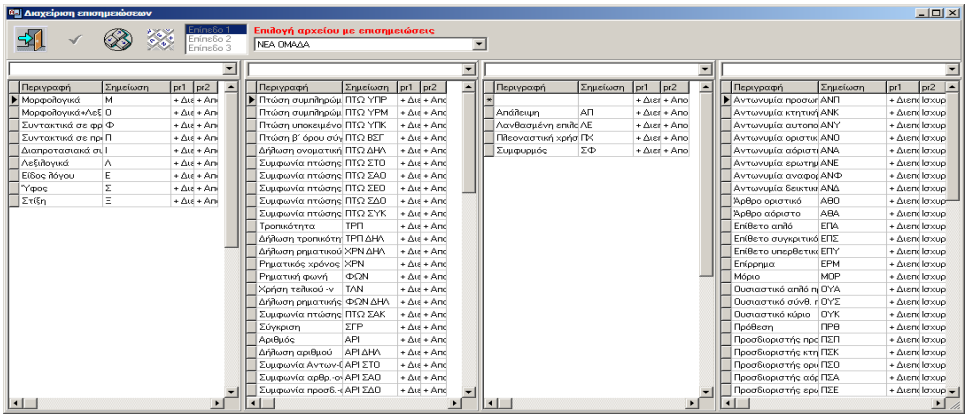

Figure 4: Insert a tag (step2 select tag)

- From the interface of the available annotations, the user chooses the one he wants by a double click of the left button of the mouse. In the case of error

annotation, the user will be presented with one more interface in order to store the correct text substituting the error (see figure 5).

Due to the fact that rules have been applied to each tag-set during the time the user-administrator created the tag-set and saved it in the program's database, the user's choices are limited to only the absolutely essential ones.

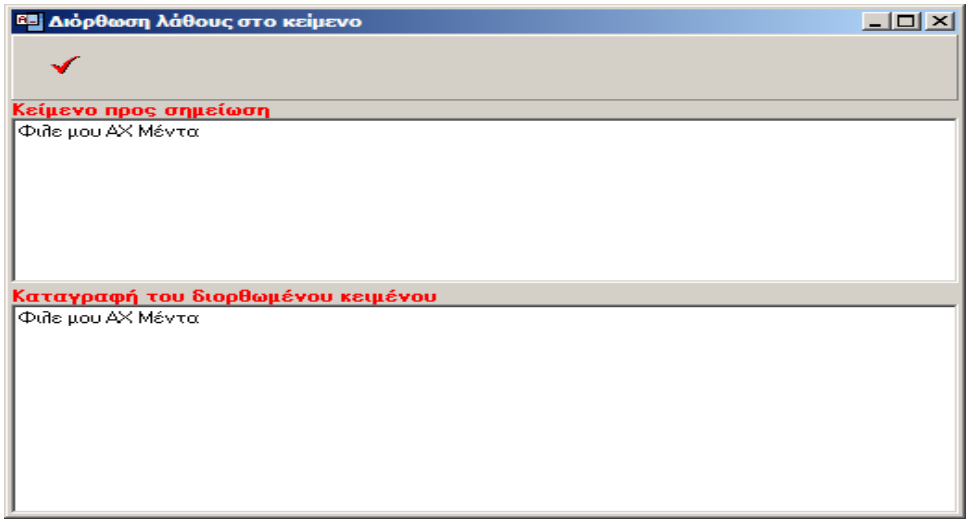

Figure 5: Insert a tag (step3 insert correct text)

3.3.3 Metadata files and handling

For each text file, which is imported for annotation, "Episimiotis" creates four more auxiliary files. These files, which are connected to the original text file, contain metadata of the annotated text file. All the metadata files carry particularly useful and important information for further processes on the annotations. Thus, for each original text file to be annotated, "Episimiotis" creates the following four metadata files:

1. The annotated file, i.e. the file that has all the active annotations.

2. The history-of-annotations file. This file has every annotation (together with its attributes) that was entered, altered or deleted in the original file. The attributes of every annotation are:

    - The user code (so that the user-administrator knows what each user has done).
    - The date and time that this annotation was entered.
    - The current state of the annotation: active or deleted.
    - A number which indicates the position of the first character of the annotated text in the original text file.
    - A number which indicates the position of the last character of the annotated text in the original text file.
    - A number which indicates the position of the first character of the annotated text in the annotated file.

- A number which indicates the position of the last character of the annotated text in the annotated file.
- A number which indicates the depth in the hierarchy of levels that this annotation is (first level, second level, third level).
- The description of the annotation (the tags used for it).
- The correct part of text which may be stored when the program is used for error annotating.
- The erroneous part of text (from the original text file) which may be stored when the program is used for error annotation.

3. The file with all the active annotations. It is a file with the same structure as the second one. The only difference is that this third file includes only those annotations that still remain in the annotated text, i.e. it doesn't include the annotations that have been deleted during the whole process of annotation.

4. The file with all the active annotations (as in the third file) delimited by tab. This function allows the user to open and process the file with other programs (e.g. Microsoft Excel) so that further statistical process can be done. In a hypothetical example and in the case that the program is used for error annotation, the user can search for the most frequent errors, the distribution of errors depending on the category etc.

3.3.4 Overlapping annotations

An important problem in text annotation is that of the overlapping annotations. Under the term "overlapping annotations" we mean problems which result from the structure in the recording of annotations. In a hypothetical example the structure is <Tag1><Tag2>_</Tag2></Tag1> acceptable, while the structure <Tag1><Tag2></Tag1></Tag2> is not acceptable.

In order to solve the problem of overlapping annotations, the multiple levels in the import of annotations is used (these levels have nothing in common with the three hierarchical plus the fourth independent level which were previously described) . In other words an additional attribute has been imported in each annotation. This attribute is the level of annotation in which each particular annotation belongs to. In this way, when a not acceptable structure of annotation (overlapping annotations) exists, the tool will recognise it without the intervention of the user and it will create automatically a new level of annotations where will also be stored in the annotation. After the import of a new level of annotations, the structure <Tag1><Tag2></Tag1></Tag2> will be acceptable since <Tag1></Tag1> and <Tag2></Tag2> are in different annotation levels.

4. Future objectives

In the next versions of "Episimiotis", a new unit will be included in the tool. By using this unit the user will perform statistical processes on the annotations. After performing these statistical processes, the user will be able to export useful conclusions that concern annotated files. Some of them could be:

1. Semi-automated (or automatic where it is feasible) importing of rules so that the mark-up language, which is used in the annotation of texts, follows the XML specifications.

2. Incorporation of ability to annotate syntactic structures (e.g. name phrases, subject, object etc.).

3. Incorporation of ability to annotate textual indicators (e.g. co-reference, name entities etc).

4. Creation of a system for automatic recognition of name entities.

Acknowledgments

The present study has been funded and published in the framework of the research programme **PYTHAGORAS I** which is co-funded by the European Social Fund (75%) and National Resources (25%) - Operational Program for Educational and Vocational Training II (EPEAEK II).

References

Granger, S. (2003) Error-tagged Learner Corpus and CALL: A Promising Synergy. CALICO Journal, 20 (3), 2–16.

James, C. (1998) Errors in Language Learning and Use. Exploring Error Analysis. London & New York: Longman.

Lüdeling Anke, Peter Adolphs, Emil Kroymann & Maik Walter (2005) Multi-level error annotation in learner corpora. *Corpus Linguistics 2005, Birmingham, England, 2005.* Available on-line from http://www.corpus.bham.ac.uk/PCLC/Falko-CL2006.doc