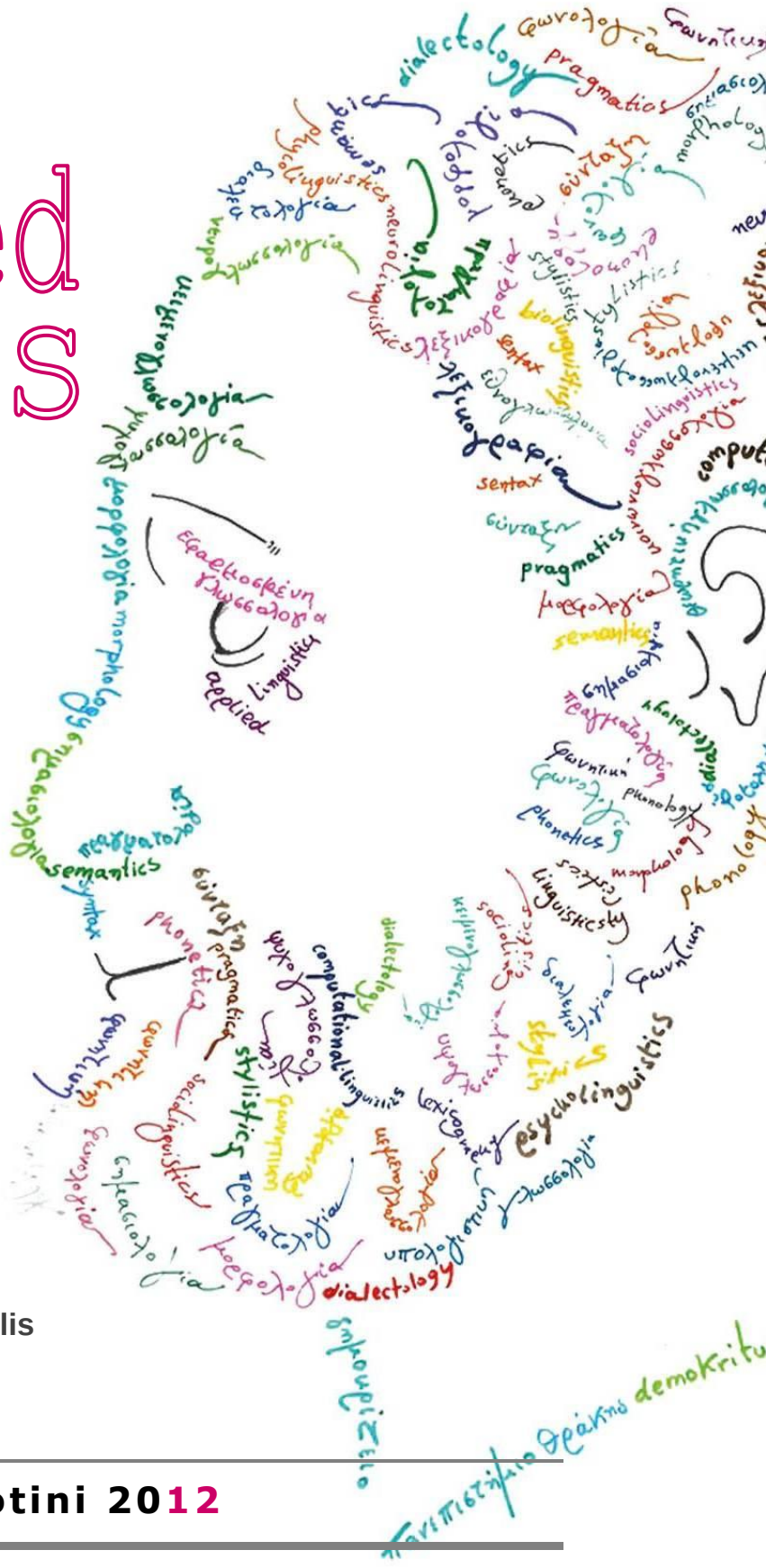
DEMOCRITUS UNIVERSITY of THRACE

selected papers of the 10th International Conference of Greek Linguistics



Edited by

Zoe Gavriilidou
Angeliki Efthymiou
Evangelia Thomadaki
Penelope Kambakis-Vougiouklis

Komotini 2012

ISBN 978-960-99486-7-8

Τυπογραφική επιμέλεια

Νίκος Μαθιουδάκης
Ελένη Παπαδοπούλου
Ελίνα Χατζηπαπά

Σχεδιασμός εξώφυλλου

Νίκος Μαθιουδάκης

A SUITE OF NATURAL LANGUAGE PROCESSING TOOLS FOR GREEK

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

Byron Georgantopoulos
Institute for Language and Speech Processing/Athena RIC, Athens, Greece
byron@ilsp.gr

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

ABSTRACT

Τα τεράστια κειμενικά δεδομένα που είναι διαθέσιμα σήμερα σε ηλεκτρονική μορφή απαιτούν εύρωστες τεχνολογίες επεξεργασίας φυσικής γλώσσας (ΕΦΓ). Η αλυσίδα αρθρωμάτων ΕΦΓ που έχει αναπτύξει το Ινστιτούτο Επεξεργασίας του Λόγου είναι μοναδική για την Ελληνική γλώσσα και μπορεί να χρησιμοποιηθεί τόσο για τη μελέτη διαφόρων γλωσσικών φαινομένων για ερευνητικούς σκοπούς όσο και για την αυτόματη ανάλυση κειμενικών συλλογών με στόχο την αποδοτικότερη δεικτοδότηση και χρήση τους. Τα εργαλεία που παρουσιάζονται σε αυτό το άρθρο στηρίζονται σε τεχνικές μηχανικής μάθησης αλλά και σε νομοθετικές προσεγγίσεις. Τα περισσότερα είναι ήδη διαθέσιμα ως διαδικτυακές υπηρεσίες από τη διεύθυνση http://nlp.ilsp.gr/ws/.

1. Introduction

The vast amount of electronically available textual data constitutes a wealth of information for both researchers and application developers. On the other hand, the overwhelmingly big datasets of today ask for robust and efficient processing tools. While a variety of relevant processors exist for well-resourced languages like English, it is often difficult to find similar tools for texts in less-spoken languages. In this paper we provide an overview of natural language technologies available from the Institute for Language and Speech Processing. This NLP suite is unique for the Greek language and comprises a series of processing units based on both machine learning algorithms and rule-based approaches. We report on updated versions of tools originally presented in Papageorgiou (2002) and, taking into account latest developments in this field, on new processors that we have implemented, together with the resources we created for their training and evaluation. Our infrastructure can be used by researchers interested in studying linguistic properties of the Greek language. At the same time, it can be employed in application scenarios involving fast processing of large document collections.

The paper is organized as follows. Section 2 discusses detection of paragraph, sentence and token boundaries in input text. Modules presented in Section 3 assign POS tags and lemmas to tokens. Section 4 presents a dependency treebank for training data-driven parsers. A term spotting algorithm is discussed in Section 5. Sections 0 and 7 focus on modules for sentence compression and text summarization. In Section 8, we discuss integration and use of the tools via standards-compliant web services.

In Z. Gavriilidou, A. Efthymiou, E. Thomadaki & P. Kambakis-Vougiouklis (eds), 2012, *Selected papers of the 10th ICGL*, pp. 511-519. Komotini/Greece: Democritus University of Thrace.

2. Paragraph, sentence and token segmentation

At the first stage of our processing architecture, input is channeled to a module that segments text into paragraphs, sentences and tokens. Input is read from locally stored text files or from documents collected from the Internet, stripped of their HTML markup (apart from paragraph tags) and stored as XML files.

When paragraph segmentation is available in the input as paragraph markup, this is taken into account. In the opposite case, a paragraph segmentor detects first whether input text has paragraphs broken across lines. The segmentor counts the relative frequency of non-empty lines that begin with a character that is not a capital letter or any kind of opening quote, dash, or opening bracket. If the relative frequency is less than 0.35, the tool assumes that end of lines constitute paragraphs. Otherwise, it assumes that input text contains line-broken paragraphs and extends paragraph boundaries until a set of constraints, including occurrence of empty or relatively short lines, is satisfied.

Sentence boundaries are detected inside paragraphs. The text of each paragraph is first segmented on obvious sentence-final punctuation marks (e.g. *.;!*), while a set of rules based on regular expressions takes care of not splitting strings like Internet URLs or currencies (e.g. *http://www.host.gr/quote?id=NBGr.AT*, *sftp://vls@ftp.ilsp.gr*, or*35.000*). Following this simplistic segmentation, a set of post-processing heuristics is used to join wrongly split text segments into sentences. As an example, these heuristics examine whether the sentence previous to the one scanned ends with an abbreviation. For a string to be classified as an abbreviation, the tool consults an abbreviation list containing approximately 2K entries. Alternatively, it checks whether the string matches a relevant regular expression. If the previous sentence ends in a non-breaking abbreviation like *άρθρ.*, *Δρ.* or *δηλ.*, the two sentences are joined into one. In the case of abbreviations that can occur in a sentence-final position like *Σ.τ.Ε.*, *A.E.* or *χλμ.*, the initial sentence split is maintained if the second sentence starts with a capital letter. Similar heuristics are used for correcting splits between initials and last names, or splits in texts with line-broken paragraphs.

The next process is tokenization, i.e. the recognition of word and punctuation boundaries inside the text of each sentence. This again involves an initial split at obvious points in the input text (spaces, punctuation marks, etc.), followed by some postprocessing. The latter includes cases like avoiding the separation of the relative indefinite pronoun *ό,τι*; splitting the 2nd and 3rd personal pronoun combination *σ'το* in two tokens; disambiguating between contracted forms like *'ρθει* and quote-token combinations like *'έρθει'* and recognizing one and three tokens, respectively; splitting off the period from the last word of the sentence, but remembering not to do it when the last word is an abbreviation like *O.H.E.*; detaching parentheses and hyphens but not in the case of enumerators like *2.1.1)* or of negative numbers like *-12,32*; etc. Each detected token is assigned a token type on the basis of the token itself and, in certain cases, the context of the token. The list of token types with some indicative examples is shown in Table 1.

Token type	Example	Token type	Example
DATE	*16/6/43*	ENUM (enumerator)	*2.1 i. a)*
PUNCT (punctuation)	*, - · (ano-teleia)*	DIG (digit)	*1.0009,1% - ⅜ ∛*
PTERM (terminal punct.)	*;!...*	INIT (initial)	*Τ. Χρ. Γερ.*
PTERM_P (potentially terminal punct.)	*. : ; !*	ABBR (abbreviation)	*δισ.ΟΓΑ ΣΥ.ΡΙ.ΖΑ*
OPUNCT (opening punct.)	*« " ([{*	NBABBR (non-breaking abbr.)	*π.χ. αναφ.*
CPUNCT (closing punctuation)	*» ")] }*	TOK (default)	Default type for all other tokens

Table 1 Token types

Sentences are also assigned a type attribute based on their capitalization. The list of values for sentence types includes `uppercase` for sentences typed in capital letters and `titlecase` for sentences where the first letter of every token is capitalized. An optional process involves normalization of `uppercase` sentences or sentences with regular capitalization, when no diacritics have been used by the author of the text. In this step, diacritics are restored to ease processing of other downstream processors like part of speech taggers and parsers. Diacritic restoration is performed as in Scannell (2011) by querying a lexicon of frequent words and, in the case of ambiguity (*δίκη/δική*), a table of bigram probabilities (e.g. `δίκη|–|πολιτική`) learned from large crawled corpora of Greek.

3. Part of speech tagging and lemmatization

After tokenization, we add morphosyntactic annotations to each token using a part of speech tagger called FBT. FBT is an adaptation of the Brill tagger (Brill, 1992) trained on a manually annotated corpus of Greek texts amounting to 455K tokens. During manual and automatic annotation, we use a tagset of 584 combinations of basic POS tags (Table 2) and morphosyntactic features, which capture the rich morphology of the Greek language[1].As an example, the full tag `AjBaMaSgNm` for a word like *ταραχώδης* denotes an adjective of basic degree, masculine gender, singular number and nominative case.

POS	Description	POS	Description
Ad	Adverb	PnIr	Interrogative pronoun
Aj	Adjective	PnPe	Personal pronoun
AsPpPa	Preposition + Article combination	PnPo	Possessive pronoun
AsPpSp	Preposition	PnRe	Relative pronoun
AtDf	Definite article	PnRi	Relative indefinite pronoun
AtId	Indefinite article	PtFu	Future particle
CjCo	Coordinating conjunction	PtNg	Negative particle
CjSb	Subordinating conjunction	PtOt	Other particle
PnDm	Demonstrative pronoun	PtSj	Subjunctive particle

Table 2 Common part of speech tags

For the construction of the corpus, linguists had to correct automatically assigned tags from an initial version of the tagger. We used interfaces that allow annotators to select between (features of) tags for ambiguous tokens. For example, in Figure 1, a user selects the `Nm` (nominative) value for the case feature to correct a wrongly assigned `Ac`(cussative) for the noun *στρατόπεδο*.

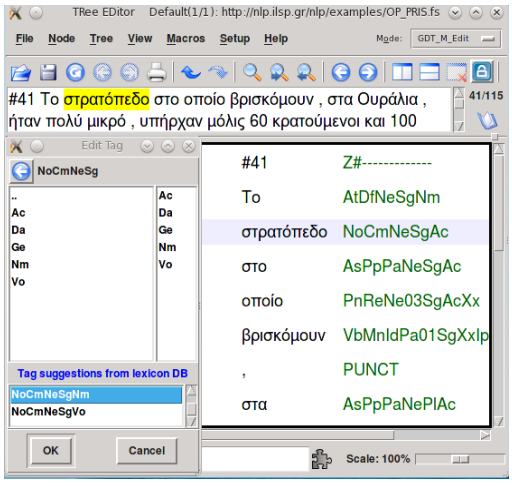


Figure 1 User interface for annotation of POS tags

Input	του μνημονιακού/AjBa**Ne**SgGe χειμώνα/NoCm**Ma**SgGe
Rule	AjBa**Ne**SgGe ->AjBa**Ma**SgGe NEXTTAG NoCm**Ma**SgGe
Output	του μνημονιακού/AjBa**Ma**SgGe χειμώνα/NoCmMaSgGe
Input	απαιτείται διαρκής/AjBaFeSg**Nm** επαγρύπνηση/NoCmFeSg**Ac**
Rule	NoCmFeSg**Ac** ->NoCmFeSg**Nm** PREVTAG AjBaFeSg**Nm**
Output	απαιτείται διαρκής/AjBaFeSgNm επαγρύπνηση/NoCmFeSg**Nm**

Table 3 Context rules correcting gender and case

During automatic processing, the tagger assigns to each token the most frequent tag in a lexicon compiled from the training corpus and augmented with entries from ILSP's Morphological Lexicon[2]. A lexicon of suffixes guides initial tagging of unknown words: for example, an entry like `νιακού-AjBaNeSgGe` would assign this specific tag to a word like *μνημονιακού*. After that, a set of about 800 contextual rules is applied to correct initial tags. The rules were automatically learned from the training corpus as detailed in Papageorgiou et al. (2000). When a token exists in the lexicon, rules are allowed

1 See http://nlp.ilsp.gr/nlp/tagset_examples/tagset_en/ for a full description of the tagset, including all morphosyntactic features and indicative examples.

2 http://www.ilsp.gr/en/services-products/langresources/item/32-ilektronikomorfologiko

to change its tag only if the resulting tag exists in the token's entry in the lexicon. As an example of rule application, the first rule in Table 3 would assign a masculine value for the gender feature of *μνημονιακού* in a context like *μνημονιακού χειμώνα*. FBT's accuracy has been tested against a 90K partition of the manually annotated corpus not used in training. 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 is 92.54%.

Following POS tagging, a lexicon-based lemmatizer retrieves lemmas from the Morphological Lexicon. This resource contains 66K lemmas, which in their expanded form extend the lexicon to approximately 2M different entries. When a token under examination is connected in the lexicon with two or more lemmas, the lemmatizer uses information from the POS tags assigned to disambiguate. For example, the token *ενοχλήσεις* will be assigned the lemma *ενοχλώ,* if tagged as a verb, and the lemma *ενόχληση*, if tagged as a noun.

4. Dependency parsing

One of the most prominent current paradigms in automatic syntactic analysis is dependency parsing. Dependency parsers create tree representations for each input sentence, where each word depends on a head word and is assigned a label depicting its relation to the head word. Treebanks with manually created annotations are used to train and evaluate data-driven dependency parsers. We have trained open source parsers on the Greek Dependency Treebank, a resource that comprises data annotated at several linguistic levels (Prokopidis et al., 2005). As of 2011, GDT contained 118+K tokens in 4948 sentences, while more annotated texts are being added[3]. Lemmas and POS tags for all tokens are manually validated. The texts include transcripts of European parliamentary sessions, articles from the Greek Wikipedia and web documents pertaining the politics, health, and travel domains.

Dep. Rel	Description	Dep. Rel.	Description
Pred	Main sentence predicate	Coord	A node governing coordination
Subj	Subject	Apos	A node governing apposition
Obj	Direct object	*_Co	A node governed by a Coord
IObj	Indirect object	*_Ap	A node governed by an Apos
Adv	Adverbial dependent	AuxC	Subord. conjunction node
Atr	Attribute	AuxP	Prepositional node
ExD	A node whose parent node is not present in the sentence (ellipsis)	AuxV	Particles or auxiliary verbs attached to a verb

Table 4 Common dependency relations in the Greek Dependency Treebank

The scheme used during manual annotation includes 25 main relations (Table 4) and is based on an adaptation of the guidelines for the Prague Dependency Treebank (Böhmová et al. 2003). The guidelines include indicative examples of several syntactic phenomena. For example, coordination structures (Figure 2) are headed by a conjunction assigned the label *Coord*, while each node headed by the conjunction is annotated with a label like *Obj_Co*. These labels denote both the node's function in the sentence and the fact that it participates in a coordination structure.

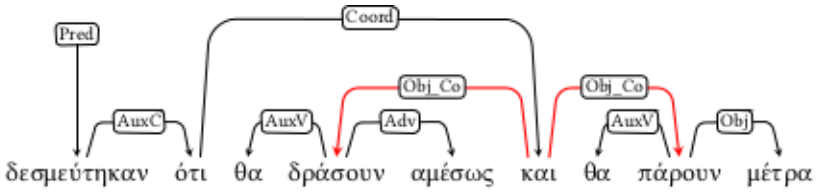


Figure 2 Representation of coordination structures

The scheme allows for simple and intuitive descriptions of structures common in languages which, like Greek, exhibit a flexible word order. Since dependency relations are directly encoded, without the

3 Updated information on the GDT can be found at http://gdt.ilsp.gr/.

presupposition of any default constituent structure from which all others are derived, representation for the main relations in a sentence is straightforward. In an OVS example like *την έγκρισή τους έδωσαν οι υπουργοί*, the verb *έδωσαν* heads the sentence as the main predicate, while two words, *έγκριση* and *υπουργοί*, are annotated as object and subject dependents of the predicate respectively.

Non-projective structures are also allowed in the scheme. As an example, subjects or objects extracted from an embedded clause can be linked to their head without the use of co-indexation with a trace. This is illustrated in the non–projective tree of Figure 3, where the relative pronoun *οποία* directly depends as a subject to its head *έλειπαν*, thus crossing the link of the verb heading the relative structure to the antecedent.

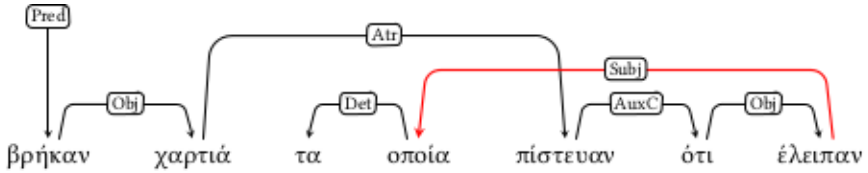


Figure 3 Non-projective relation

In n-fold experiments with the MaltParser system for dependency parsing (Nivre, 2007), we have trained models on the GDT that showed an overall labeled attachment score (i.e. the proportion of tokens attached to the correct head and assigned the correct dependency relation) of 74.83% and an overall unlabeled attachment score of 81.04%. Precision and recall for the subject relation reached 83.49% and 89.46% respectively.

5. Term extraction

We can view terms as linguistic realizations of domain specific concepts, usually lexicalized in the form of noun phrases. For terminology recognition we have implemented a hybrid methodology: we initially construct a candidate term set using a term grammar and then filter this set through statistical techniques. The module operates on input with lemmas and part-of-speech tags assigned to each word. First, the following term pattern grammar recognizes single and multi-word (up to 4-word) candidate terms:

```
((Adj|Noun)*(Prep|Det)?) (Adj|Noun)* Noun
```

Then, a statistical filter following the *tf-idf* paradigm is applied to the list of grammar-extracted terms in order to rank them according to statistical evidence. The reference corpus used in the *idf* calculation is the Hellenic National Corpus (HNC, http://hnc.ilsp.gr), a 47M words tagged and lemmatized corpus covering a wide range of topics including, among others, news, literature, science and business. The following formula calculates the confidence score for a term:

$$Score(candidate) = tf \cdot log(idf)$$

In the case of 2-word terms we use contingency table statistics (Daille, 1995). For a given pair w_i + w_j (as, for example, in the case of noun + noun), the contingency table is defined as in the following table:

	w_i	w_i, $j \neq j'$
w_i	a	b
w_i, $i \neq i'$	c	d

Table 5 Contingency table for 2-word terms

where *a* stands for the frequency of pairs involving both w_i and w_j (number of occurrences of a pair); *b* stands for the frequency of pairs involving w_i and $w_{j'}$ (number of occurrences of pairs where a given word appears as the first element of the pair); *c* stands for the frequency of pairs involving $w_{i'}$ and w_j (number of occurrences of pairs where a given word appears as the second element of the pair); and *d* stands for the frequency of pairs involving $w_{i'}$ and $w_{j'}$ and has a constant value calculated from the HNC

(total number of occurrences of all the pairs in the reference corpus). The score formula is based on log-likelihood:

$$Score(2-word) = a \cdot log(a) + b \cdot log(b) + c \cdot log(c) + d \cdot log(d) \\ -(a+b) \cdot \log(a+b) - (a+c) \cdot \log(a+c) \\ -(b+d) \cdot \log(b+d) - (c+d) \cdot \log(c+d) \\ +(a+b+c+d) \cdot \log(a+b+c+d)$$

A couple of factors were taken into consideration in order to smooth the confidence scores across candidate terms with (1) the same number of words and (2) with different number of words. Regarding the former factor, the top-scoring term of each set of terms with the same number of words is assigned a score of 1 (the maximum) and all the others are analogically calibrated from 0 to 1. Regarding the latter factor, in order to account for the fact that *idf* statistics are getting sparser as the number of words increases, we weight the score of a candidate term with the number of words it maintains, in a logarithmic fashion:

$$Score(n-word\ term) = OriginalScore \cdot \big(1 + log(n)\big)$$

Figure 4 displays an example of terms extracted from the sentence: *Σε κινητοποίηση κατεβαίνουν την Τετάρτη και την Πέμπτη οι εργαζόμενοι της Wind και της Vodafone για τις ελαστικές σχέσεις εργασίας (ακόμα και ενοικίαση ή πώληση εργαζομένων!!) αλλά και για τις παράνομες απολύσεις.*

```
<Term conf="0.784" end="#w2" start="#w2" text="κινητοποίηση"/>
<Term conf="0.915" end="#w10" start="#w10" text="εργαζόμενοι"/>
<Term conf="1" end="#w20" start="#w18" text="ελαστικές σχέσεις εργασίας"/>
<Term conf="0.596" end="#w24" start="#w24" text="ενοικίαση"/>
<Term conf="1" end="#w27" start="#w26" text="πώληση εργαζομένων"/>
<Term conf="1" end="#w36" t="#w35" text="παράνομες απολύσεις"/>
```

Figure 4 Example output from the term extractor

6. Sentence compression

Sentence compression is used as a building block in, among others, text simplification and automatic summarization applications. Our sentence compression tool (Prokopidis et al., 2008) processes syntactically analyzed input by a) replacing words with paraphrases shorter in length and b) deleting elements carrying relatively small semantic information.

We used a thesaurus of synonyms and antonyms (Ιορδανίδου, 2005) to manually construct an initial seed of paraphrase lemmas. Paraphrases that were too domain- or register-specific were filtered-out. We then evaluated the seed against the HNC, checking for paraphrase interchangeability and applicability in different linguistic contexts. When all morphological variants of each lemma were automatically generated, we came up with a table of 9860 paraphrase entries consisting of types and morphological features shared by types (Figure 5). Since input is expected to be automatically annotated for the same features, this information guides the paraphrase module into making correct substitutions for homographic source types that may correspond to more than one target types. Thus, if input text contains the noun *θιασώτες*, the module will choose between *οπαδοί* and *οπαδούς* based on the case feature automatically assigned to the source noun.

```
<Paraphrase source="θιασώτες" stag="NoCmMaPlAc" target="οπαδούς" />
<Paraphrase source="θιασώτες" stag="NoCmMaPlNm" target="οπαδοί" />
<Paraphrase source="αγαθοεργίες" stag="NoCmFePlAc" target="ευεργεσίες"/>
```

Figure 5 Paraphrases sharing the same morphological features

A set of deletion rules operates on the output of the paraphrase module. Each deletion rules traverses the nodes of the dependency tree, checking whether specific morphosyntactic constraints apply for the node currently examined. When the constraints match, the node and the subtree that is headed by this node are marked as deletables. Constraints may focus on the node's (or children or

parent nodes') dependency relations, their POS tag, etc. The most frequent actions involve deletions of adjectives (*delAdjs*), adverbs (*delAdvs*, Figure 6) and preposition-headed adverbials (*delPPs*). As an example, *delAdjs* selects as deletion candidates adjectives which a) are not the heads of other nodes (e.g. *ο καλύτερος όλων*) and b) are not headed by a copula verb (e.g. *είναι μόνος*). Subtrees marked to be deleted are ranked according to their *relevance*, which is estimated as in Daelemans et al. (2004) on the basis of the log-likelihood of the frequencies of the subtree words, as these frequencies were observed in a 70M words Greek corpus. Using this information, the deletion of less significant subtrees, which is expected not to seriously affect sentence meaning, precedes elimination of more important subtrees.

```
Orig: Τις τελευταίες δεκαετίες της τις πέρασε στο Παρίσι, όπου σκόρπισε
αφειδώς τα χρήματά της σε αγαθοεργίες.
Paraphrase 7_1: αγαθοεργίες -> ευεργεσίες
Alt: Τις τελευταίες δεκαετίες της τις πέρασε στο Παρίσι, όπου σκόρπισε
αφειδώς τα χρήματά της σε ευεργεσίες.
Deletion 7_2: (relevance =13.38): αφειδώς
Alt: Τις τελευταίες δεκαετίες της τις πέρασε στο Παρίσι, όπου σκόρπισε τα
χρήματά της σε ευεργεσίες.
```

Figure 6 Reducing sentence length via paraphrase application and subtree deletion

7. Text summarization

Recent work on text summarization has mainly focused on producing extracts rather than abstracts, reflecting the difficulty in tackling complex NLP problems such as anaphora, polysemy, world knowledge, etc. Our summarizer provides extract-based, single document summaries. For each sentence a score, indicative of its salience, is calculated as a weighted sum of several summary-worthy features. The summarization process requires an input with terms and named entities recognized. Currently used features for each sentence include *sentence location*: sentences closer to the beginning of a document are favored; *sentence length*: sentences shorter than *n* (currently 5) content words are discarded; and *term and named entity occurrence*: inclusion and weight of terms and named entities in a sentence increases the sentence's importance. The scoring formula for all sentences with *length* ≥ *n* is the following:

$$Score(s) = w_L \log\frac{|D| - Sp + 1}{|D|} + \frac{w_T S_T + w_N S_N}{length(s)}$$

where $|D|$ is the total number of sentences in the document, S_P is the position of the sentence ($1 \ldots |D|$), S_T is the sum of confidence scores for each term in the sentence, S_N is the sum of confidence scores for each named entity in the sentence and *length(s)* is the number of content words in the sentence. The respective feature weights are $\{w_L, w_T, w_N\} = \{1, 4, 4\}$.

The final extract is built from top-scoring sentences selected in their original order in the text. The number of extracted sentences is determined by a compression factor currently set to 10% of the original text. The following figure displays a document with the top-selected extract sentence highlighted.

Ψήφο υπέρ μιας Ευρώπης που θα «μετρά» ως παγκόσμια δύναμη ζητά ο Σιράκ. Ο Σιράκ απέκλεισε το ενδεχόμενο να παραιτηθεί, εάν τελικά οι Γάλλοι καταψηφίσουν το Ευρωσύνταγμα. **Το «όχι» στο δημοψήφισμα για το Ευρωσύνταγμα θα καθυστερήσει την ευρωπαϊκή ολοκλήρωση, προειδοποίησε τους Γάλλους πολίτες ο Ζακ Σιράκ, εγκαινιάζοντας δυναμικά την εκστρατεία υπέρ του «ναι» με τηλεοπτική του εμφάνιση.** Ο Σιράκ κάλεσε τους συμπατριώτες του να ψηφίσουν «ναι» στο κρίσιμο δημοψήφισμα της 29ης Μαΐου, προκειμένου να συμβάλουν στην οικοδόμηση «μιας Ευρώπης, που θα 'μετρά' ως δύναμη στον κόσμο του αύριο». Ταυτόχρονα, απέκλεισε το ενδεχόμενο παραίτησής του, εάν τελικά ψηφίσουν «όχι» στο Ευρωσύνταγμα.

Figure 7 Top-selected sentence for an extract-based summary

8. Integrating and accessing the tools

Integrating the tools mentioned above into a robust and efficient pipeline capable of analyzing the enormous amounts of texts available online today is not a trivial task. To accomplish this goal, we have wrapped all tools as UIMA (http://uima.apache.org/) modules. UIMA is an open source framework for developing analyzers of unstructured data. The framework caters for separation of algorithmic design from input and output requirements and allows NLP engineers to predefine the annotation type system to use. The framework also uses the stand-off annotation practice, where automatic and manual annotations compatible to the type system are separated from primary data.

Our team has been actively involved in national and European projects aiming at automating the stages involved in the acquisition, production, updating and maintenance of language resources and tools. Given the large number of linguistic services and tools already developed by various organizations throughout Europe, the need for building interoperable infrastructures surpassing different underlying technologies becomes apparent. To this end, we have already made available most of the tools described above as web services that can be accessed and tested by linguists or other interested end-users from http://nlp.ilsp.gr/ws/. Since these services are standards-compliant, they can be combined with services provided by other teams and organizations in larger processing workflows.

9. Conclusions and future work

We presented a suite of robust processing tools for the analysis of Greek texts that can be used in research and application settings. The tools are developed and evaluated on the basis of several manually annotated resources. We plan to augment this battery of language resources and tools in the hope that this effort will provide valuable support to both theoretical linguists and language engineers. Our current research focuses on the development of tools for coreference resolution and spatiotemporal anchoring of events.

10. Acknowledgements

This paper was supported by PANACEA, a 7th Framework Research Program of the European Union, contract 7FP-ITC-248064.

References

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

Böhmová A., J. Hajič, E. Hajičová, and B. Hladká. “The Prague Dependency Treebank: A Three-Level Annotation Scenario”. In *Treebanks: Building and Using Parsed Corpora*. Kluwer, 2003.

Boutsis S., P. Prokopidis, V. Giouli and S. Piperidis. 2000. “A Robust Parser for Unrestricted Greek Text”. In *Proceedings of the 2nd Conference on Language Resources and Evaluation*. Athens.

Brill, E. 1992. “A simple rule-based part of speech tagger”. In *Proceedings of the Workshop on Speech and Natural Language*, 112-116.

Daille, B. 1995. “Combined approach for Terminology Extraction: Lexical statistics and linguistic filtering”. In *TALANA*, Universite Paris 7.

Daelemans, W., A. Höthker and E.Tjong. 2004. “Automatic Sentence Simplification for Subtitling in Dutch and English”. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*. Lisbon.

Frantzi, K.T., S. Ananiadou and H. Mima. 2000. “Automatic recognition of multi-word terms: the C-value/NC-value method”. *International Journal on Digital Libraries*, 3(2): 115-130.

Georgantopoulos B., T. Goedeme, S. Lounis, H. Papageorgiou, T. Tuytelaars and L. Van Gool. 2006.“Cross-media Summarization in a retrieval setting”. In *Proceedings of the 5th Conference on Language Resources and Evaluation*. Genova.

Georgantopoulos, B. and S. Piperidis. 2000. “A Hybrid Technique for Automatic Term Extraction”. In *Proceedings of International Conference on Artificial and Computational Intelligence for Decision, Control and Automation in Engineering and Industrial Applications- ACIDCA'2000*, 124-128.

Ιορδανίδου, Α. 2005. *Θησαυρός Συνωνύμων και Αντιθέτων της Νέας Ελληνικής*. Εκδόσεις Πατάκης, Αθήνα.

Nivre, J., J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov and E. Marsi. 2007. “MaltParser: A language-independent system for data-driven dependency parsing”. In *Natural Language Engineering*, 13(2): 95-135.

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

Papageorgiou H., P. Prokopidis, I. Demiros, V. Giouli, A. Konstantinidis and S. Piperidis.2002. "Multi-level XML-based Corpus Annotation". In *Proceedings of the 3rd Language Resources and Evaluation Conference*. Las Palmas.

Prokopidis, P., E. Desipri, M. Koutsombogera, H. Papageorgiou and S. 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*, 149-160. Barcelona.

Prokopidis P., V. Karra, A. Papagianopoulou and S. Piperidis. 2008. "Condensing sentences for subtitle generation". In *Proceedings of the 6th Language Resources and Evaluation Conference*. Marrakech.

Radev D., T. Allison, S. Blair-Goldensohn, J. Blitzer, A. Celebi, S. Dimitrov, E. Drabek, A. Hakim, W. Lam, D. Liu, J. Otterbacher, H. Qi, H. Saggion, S. Teufel, M. Topper, A. Winkel, Z. Zhang. 2004. "MEAD - a platform for multidocument multilingual text summarization". In *Proceedings of the 4th Conference on Language Resources and Evaluation*. Lisbon.

Scannell, K. P. 2011. "Statistical unicodification of African languages".In *Language Resources and Evaluation* 45(3): 375-386