DSV
Master Thesis[1]
George Pachantouris
2004-2005

GreekSum
A Greek Text Summarizer

Thesis advisor: Dr. Hercules Dalianis

[1] This thesis corresponds to 20 weeks of full-time work

Abstract
An automatic text summarizer is a computer program that summarizes a text. The summarizer removes redundant information from the input text and produces a shorter non-redundant output text. The output text is an extract from the original text.
This master thesis describes the construction and evaluation of the first automatic text summarizer for Greek news text, named GreekSum.

GreekSum is based upon the well known SweSum summarization engine from KTH/Stockholm University and a Greek key word dictionary provided by NCSR Demokritos, Athens. The SweSum family of summarizers supports several languages, namely: Swedish, Norwegian, Danish, Spanish, French, English, German, and Persian (Farsi). There is also one language independent version of SweSum called Generic. Generic uses no key word dictionary and can be used to summarize almost every language. In this thesis is also a small evaluation of GreekSum carried out were we found that using the Greek keyword dictionary in GreekSum made the summarizer 16 percent better that not using a dictionary.

DSV Στοκχόλμη
Διπλωματική εργασία
Γιώργος Παχαντούρης
2004-2005

GreekSum
Αυτόματη Ελληνική Περιληπτική Μηχανή

Επόπτης καθηγητής: Δρ. Ηρακλής Νταλιάνης

Περίληψη

Αυτόματη περιληπτική μηχανή είναι πρόγραμμα που κάνει περιλήψεις. Η εργασία που επιτελεί είναι να αφαιρεί επαναλαμβανόμενη πληροφορία και το αποτέλεσμα είναι ενα μικρότερο κείμενο, το οποίο περιλαμβάνει ολόκληρες φράσεις από το αρχικό.

Αυτή η διπλωματική εργασία περιγράφει την κατασκευή και αξιολόγηση της πρώτης αυτόματης περιληπτικής μηχανής για κείμενα εφημερίδας στην ελληνική γλώσσα.

Η μηχανή βασίζεται στην αντίστοιχη σουηδική από το Πολυτεχνείο της Στοκχόλμης, Σουηδία (SweSum) και το ελληνικό μορφολογικό λεξικό από το ΕΚΕΦΕ Δημόκριτος. Η οικογένεια περιλαμβάνει τις εξής γλώσσες: Σουηδικά, Νορβηγικά, Δανέζικα, Ισπανικά, Γαλλικά, Αγγλικά, Γερμανικά και Περσικά. Υπάρχει και μια μέθοδος ανεξαρτήτου γλώσσας που καλείται Generic. Η τελευταία δεν χρησιμοποιεί κανενός είδους λεξικό και μπορεί να χρησιμοποιηθεί για όλες τις γλώσσες.

Στο τέλος της εργασίας βρίσκεται μια σύντομη αξιολόγηση που έδειξε ότι η ελληνική περιληπτική μηχανή έδωσε κατά 16 τις εκατό καλύτερα αποτελέσματα από την επιλογή Generic.

Acknowledgements

I would like to thank mostly my advisor Dr. Hercules Dalianis for his encouragement and enthusiasm about this project. Additionally I would like to thank Martin Hassel, Nima Mazdak and Adam Blomberg for their precious technical help. I could not forget Stergo Afanteno and Dr.Vangeli Karkaletsi from the Democritus Institute of Athens, for the kind offer of the Greek lexicon. Finally I feel the obligation to thank my Greek friends in the student residence PAX, Stockholm who gave their time for the evaluation and valuable comments.

Ευχαριστίες

Θα ήθελα πρώτα απ'όλα να ευχαριστήσω τον επόπτη μου Δρ. Ηρακλή Νταλιάνη, χωρίς την ενθάρρυνση και τον ενθουσιασμό του οποίου αυτή η εργασία θα ήταν αδύνατη. Επίσης τους Μαρτιν Χασελ,Νιμα Μαζντακ και Αδαμ Μπλουμπεργκ για την πολύτιμη τεχνική υποστήριξη . Δεν μπορώ να ξεχάσω τον Στεργο Αφαντενο και Δρ.Βαγγέλη Καρκαλετση από τον «Δημόκριτο» της Αθηνάς για την ευγενική παραχώρηση του μορφολογικού λεξικού. Τέλος τους Έλληνες φοιτητές, που μοιραστήκαμε υπέροχες στιγμές στις εστίες της Στοκχόλμης, για το χρόνο που αφιέρωσαν στην αξιολόγηση του προγράμματος και την εποικοδομητική κριτική.

Το υπόλοιπο κείμενο της εργασίας είναι στα Αγγλικά.

Table of Contents

1. Introduction

In the middle ages an average human in his whole life had to process as much information as there is today in one copy of the "Sunday Times". Information is all around us. Summarization is essential in order to be able to keep up with what is happening in the world. Some examples from everyday life are:

- Headlines of the news
- Table of contents of a magazine
- Preview of a movie
- Abstract summary of a scientific paper
- Review of a book
- Highlights of a meeting

Summary is the process of extracting the most important information from a source or sources to produce an abridged version for a specific user or users (Maybury & Mani, 1999)

1.1 Purpose

The purpose of this thesis work is to build an automatic text summarizer for the Greek language. We named it GreekSum and it is based on the algorithms developed and used for the SweSum (Dalianis, 2000), text summarizer for Swedish, created by Hercules Dalianis and Martin Hassel at KTH, Stockholm. Several changes needed to be made to support the differences of the Greek language from the Swedish already implemented in SweSum.

Apart from constructing the project, part of the required work was to evaluate it. How well does it perform without the language specific files as the Greek key word dictionary? Description of the problems encountered on the way and found solutions are recorded in this paper.

1.2 Method

GreekSum consists of a language independent summarization engine and several dictionaries. The programming language used is Perl and the server is Apache, both freely available on the internet. The Greek keyword dictionary was kindly provided by the "NCSR DEMOKRITOS"[2]. Some more that was needed and described later on, was found on the Internet.

When GreekSum was "up and running" evaluation took place. This was done by Greek native speakers in two steps. Primarily, using the Greek dictionaries and secondly by using the SweSum's built in function of Generic summarization.

[2] National Centre of Scientific Research "Demokritos"

2. Background

Text summarization is a field that has been under development for years with special algorithms and methods being available. The significance of such a tool is even more important nowadays that the Internet has become part of our everyday life. Some applications that text summarization can support are:

- News summarization down to SMS[3] or WAP[4] for Mobile phones.
- Make computers read summarized text. Written text can be too long to listen too.
- For search engines to present short description of matching text.
- In a foreign language, to obtain short translated text of a summarized text.

The two types of creating a summary, taking the most important parts of the original text are: **Abstract** and **extract**

Abstract Summarization
The result of this summary is an interpretation of the original text. The result is a smaller text where word concepts are transformed into shorter ones. For example: "They went to Kos, Rhodos, Cyprus" is becoming "They went to some islands". This kind of summarization requires symbolic word knowledge which makes it hard in order to provide a good summary.

Extract Summarization
This method uses statistical, linguistical and heuristic methods, or a combination of all to provide a summary. The result is not syntactically or content wise altered. The creation SweSum, the Swedish text summarizer, is based on this method

There are many different approaches in automatic text summarization. (Luhn, 1959) introduced word-frequency-based rules to identify sentences to use in summary, based on the feeling that most important words on a text represent the most important concepts. (Edmundson, 1969) introduced new methods such as cue phrases, title/ heading words, and sentence location in addition to Luhn's work.

2.1 Process

For performing an automatic text summarization, (Lin and Hovy, 1997) have identified three steps: topic identification, interpretation and generation, which are explained further on.

2.1.1 Topic Identification

This first step includes the identification of the essence of the text, what is it about. There are various techniques to do that, some of which are shown below. SweSum, that is mainly designed for newspaper text, has implemented the first one.

[3] SMS:Short Message Service
[4] WAP: Wireless Application Protocol

- In some text types, certain parts of it hold an important topic. The title is always important, the first sentence, the last etc.
- Some words or phrases indicate where the essence of the text is, ex. “in summary”, “in conclusion”, “to sum up”, “this paper” etc
- Some words, depending on the content of a text, tend to appear more often and this can determine the topic of it. (Word frequency)
- Some topics are identified by counting concepts instead of words (Concept frequency).

2.1.2 Interpretation

Previously we distinguished summaries into two categories. Extract and abstract. In extract, the above methods are used. However in abstract, interpretation is performed. This includes merging similar topics to one, removing redundancies etc.

Example: She **entered the plane, sat, took off and landed**. This can become: She **flew**
This technique is a lot harder to implement, but leads to very good results. This is also called unbounded lexical aggregation (Dalianis, 1999).

2.1.3 Generation

Final step of an automatic text summarization is the generation of final output. It consists of phrase merging, word or phrase printing and sentence generation. Some of the four methods following may be used (Hovy & Lin 1997):
Extraction: After the process of summarization is over, the resulting sentences and phrases are printed on the output.
Topic lists: The most frequent keywords or interpreted fuser concepts are printed on the output.
Phrase concatenation: Two or more similar phrases are merged together.
Sentence generation: The result of sentence generator is new sentences. The input to that is a list of fuser concepts and their related topics.

2.2 Method and Algorithms of summarization

In previous paragraphs we described two methods of creating summaries, abstract and extract. The abstract method requires techniques like NLP[5], semantic parsing etc., which are still under development. Most of the current automated text summarization systems use extract methods to produce summaries. The output is a collection of some important sentences of the text, reproduced word-by-word (Lin, 1999). A method, called Local Salience Method, developed by (Boguraev et al., 2001) extracts phrases rather than sentences and paragraphs.

Although extract summarization is easy to implement, there are three major difficulties:

- Finding out which are the most important sentences to use on the summary.

[5] NLP stands for Natural Language Processing or Natural Language Parsing. It is the process used by a computer to understand and produce a language that is understood by humans. In this way people can communicate with machines as communicating with other humans.

- How to generate a coherent summary
- Remove all redundancies in the summary

The scientific society has proposed some solutions to these problems. To handle the sentence choosing, a method of scoring the candidate sentences has been developed. Among those the ones with highest scores are chosen for the final summary. Some rules according to which sentences are rated are presented below (Lin, 1999):

Baseline: This is a scoring system according to which sentences take their marks depending on their place on the text. For newspaper texts, the first sentence of the text gets the highest ranking, while the last get the lowest.

First sentence: Similarly to the previous condition, the first sentence of each paragraph of the text is considered to be very important.

Title: The words included in the title along with the following sentences get a high score.

Word Frequency: Words, called open class words, which are frequent in the text, are more important than less frequent. The sentences including such keywords that are most often used in the passage usually represent the topic of it.

Indicative Phrases: Sentences containing phrases like "…this document…"

Position Score: There is a theory that certain types of documents have their key meaning in certain parts of it. For example in the newspaper text, the first four paragraphs are the most important, while in technical papers the conclusion section is the most important part.

Sentence Length: The score given to a sentence reflects the number of words in a sentence, normalized by the length of the longest text in the passage.

Proper Name: Sentences which contain proper nouns get a higher scoring.

Average Lexical Connectivity: The sentences that share more terms with other sentences are scored higher.

Numerical Data: The sentences that contain any sort of numerical data are scored higher than those that do not contain.

Proper Name: Certain types of nouns, like people's names, cities, places etc are important in newspaper texts and sentences containing them are scored higher.

Pronoun: Sentences containing a pronoun (reflecting co-reference connectivity) are scored higher than those that do not contain.

Weekdays and Months: Sentences containing names of weekdays or months are scored higher.

Quotation: Sentences containing quotations may be important for some sort of questions, input by the user.

Query signature: When a user requires a summary he or she usually has a certain topic on his/her mind. The query of the user affects the summary in that the extracted text will be compelled to contain these words. Normalized score is given to sentences depending on the number of query words they contain.

The methods described above are very important to text summarization, but alone are not enough to produce a good quality summary. Additionally several word-level techniques, such as word frequency have been criticized for the following reasons:

- **Synonymy:** There may be more than one word to express the same thing. For example underground and metro mean exactly the same thing.
- **Polysemy:** One word may have many different meaning. For example iron can mean either the metal, or the act of making the clothes straight after the washing procedure.
- **Phrases:** A phrase can have different meanings depending on the words it contains. For example: The periodic table has nothing to do with the furniture (Lin and Hovy, 1997).

By extracting sentences from a passage using just statistical keyword approach often causes a lack of cohesion on the final summary. To make them even better, some algorithms and methods presented below which work supplementary to the ones described.

2.2.1 Sentence Selection Function for Extraction

In order to score each sentence, the summarization uses several modules. These scores are combined in order to create a single one for each sentence. However, it is not quite clear how to combine these several different scores. From the several approaches that have been described on the literature, the most common point of them is that coefficients assign various weights to the individual scores. Those are summed up. It is important to mention that those coefficients are depended on the language of the text.

Simple Combination Function: This is a linear combination in which the parameters are specified manually by experimentation. These coefficients can be, as described before first sentence, numerical data etc. The score is calculated according to the following mathematical type:

$Sentence\ score = \sum_j C_j P_j$, where **C**oefficient, **P**arameter, j=1,…,n , n=number of parameters

SweSum is using a **simple combination function** to evaluate sentence scores. It assigns coefficients manually and for example 1000 is assigned to the baseline parameter.

2.2.2 Knowledge-Based Concept Counting

In 1995 Chin-Yew Lin introduced a new method that was able to identify the central ideas in a text, based on a knowledge-based concept counting paradigm. According to him, word frequency methods described before understand only the literal word forms and ignore

conceptual generalizations. As an example, the topic in the sentence **"He was observing the beautiful stars, galaxies, planets …"** is **universe** which by any mean cannot result from word counting or other methods. It is impossible to relate in deeper level of semantics, that is required, the words stars, galaxies etc with universe.
Concepts are generalized using concept generalization taxonomy (WordNet). Figure 1 illustrates a hierarchy for the concept Car Company. Following this hierarchy we can understand that if we see BMW, Toyota, and Fiat in the passage, we can realize it is about Car industries. Furthermore, if the passage also mentions Volvo, Scania and MAN, which are truck companies, it makes sense to say that text has relation with vehicle companies.

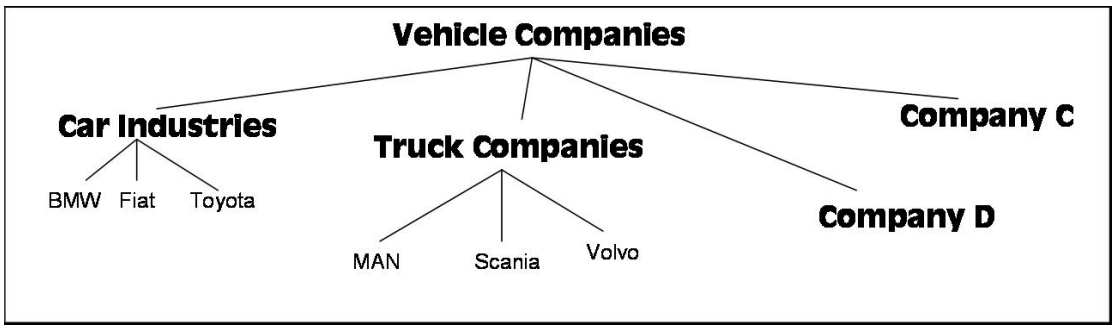

Figure 1: Hierarchy for the concept "Car Company"

The primary problem arising by using this method is how to find the most appropriate generalization in the taxonomy hierarchy. The nodes it the middle of the taxonomy are the most appropriate, since the top is a thing (everything is a thing). Using the leaf concepts gives us no power for generalization. Ratio (R) is a way to find the degree of summarization. The higher this number, the more it reflects only one child. R is given by the following mathematical formula:

R=MAX (W)/SUM (W)
, where W=weight of all the direct children of a concept.

The weight of the parent concept is defined as the frequency of occurrence of a concept C and its sub concepts in a text. In Figure 2 the Ratio is 5/ (5+2+1) = **0.625**
Another term, the branch ratio threshold (Rt) serves as a cutoff point for interestingness. It determines the degree of generalization. If Ratio is less that branch ratio threshold, it is an interesting concept. In the example, if the Rt = 0.5 we should choose BMWas the topic instead of its parent because we have R>Rt.

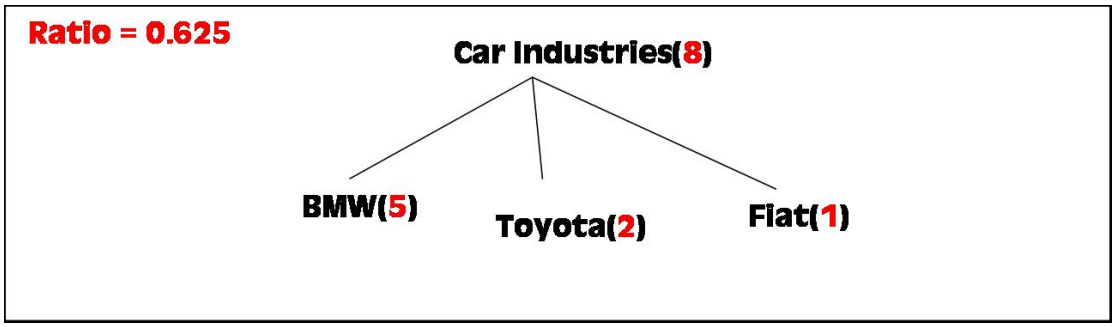

Figure 2: Ratio example

2.2.3 Lexical Chain Methods

Although word frequency is a very good way to measure importance of text content, it does not take into consideration the relation between the different parts of a text, and therefore sentence extraction with this method often lacks cohesion. Methods like lexical chains, based on sentence or paragraph extraction, have been proposed to address the problem. According to (Brunn et al., 2002), a lexical chain is a set of words in a text that are related to each other. This relation between the words can be found using lexical lists such as thesaurus or computer based lexicons, for example WordNet. With this method, the most important concepts can be found statistically, by looking the structure of the passage rather than deep semantic meaning. To calculate this, all required is a generic knowledge base containing nouns and their associations.
The general algorithm for computing word lexical chain is the following:

- Make a list of candidate words from the passage
- For each of the candidate words, find an appropriate lexical chain to get a candidate word, relying on the relatedness criterion between members of the lexical chain and the candidate words.
- If such a chain is found, insert the candidate word in the lexical chain and update it accordingly or else create a new chain.

Chains are scored depending on a number of heuristics, some of which are their length, the kind of relation between their words, the position they hold in the passage, etc. The ones that are the mostly connected to lexical chains are those that are being extracted.

The major drawback of the lexical chains is that they are insensitive to the non-lexical structure of passages, such as their rhetorical, argumentative or document structure. To give an example, they do not take into account the position of elements of a chain within the argumentative line of the discourse, often not within the layout determined structure of the document. So, the relevance of chain elements is calculated irrespective of other discourse information.

2.2.4 Latent Semantic Analysis (LSA)

This is a statistical, corpus-based text comparison mechanism. In the beginning it was launched as a task for information retrieval, but in the preceding years it showed remarkably human-like abilities in several language tasks (Wiemer, 1999). It can be used in sentence extraction methods and it helps reducing the summary. Although anti-redundancy was not accounted for in early summarizing systems, it holds an essential place in modern ones. Anti-redundancy scoring is computed dynamically as the sentences are included in the summary, to make sure that there is no information being repeated on the final summarized text.

Comparison of sentence extraction using both LSA and relevance method, was made by (Gong and Liu, 2001). In the last one, the passage is decomposed into sentences and furthermore, each one of them is presented as a vector of words that it consists of. Therefore the whole text is presented as a vector of word frequencies. These frequencies are weighted by local and global word weights and those are used to determine the relevance. A sentence that has the higher relevance is picked from the passage, included in the summary and all the words contained in it are removed from the document vector. This process continues until the

number of sentences in the summary has come to a pre-defined value. The one that conveys the most information is the one more relevant to the document vector. In this way, every time sentences with maximum information are selected according to the sentence vector of that time. This process of removing words makes sure that redundant information is not included in the final summary. The latent semantic analysis approach uses a matrix decomposition mechanism to generate an index matrix (Landauer et al., 1998). This mechanism is called Singular Value Decomposition (SVD). This index matrix is used to select the number of sentences to be included in the final summary.

2.2.5 Vector-Based Semantic Analysis using Random Indexing

This is a technique to extract, from a text, semantically similar terms by observing the distribution and collection of terms inside the text (Karlgren and Sahlgren, 2001). The result of running a vector-based semantic analysis on a text is a thesaurus: an associative model of term meaning.

Random Indexing (RI) uses sparse, high-dimensional random index vectors to represent documents. Based on the hypothesis that any document has assigned a random index vector the term similarities can be calculated by computation of terms-by-contexts co-occurrence matrix. Each of the rows of it represent a term, and the term vectors are of the same dimensionality as are the random vectors assigned to texts. Every time a term is found in a text, that text's random index vector is being added to the row for the term in question. With this method, terms are represented in the matrix be high-dimensional semantic context vectors that contain traces of every context the specific has been observed in. The assumption behind this theory is that semantically similar terms will show up in similar contexts and therefore their context vectors will be quite similar. In this way, by calculating similarities between context vectors, it should be possible to calculate the semantic similarity between any given terms. This similarity measure will reflect the distributional or contextual similarity among different terms.

2.2.6 Pronoun Resolution

When performing an automatic summary no deeper linguistic analysis is conducted. Therefore the resulting text can often result in broken anaphoric references. For example in the following example we assume that the summarization algorithm selects the second sentence. If "Maria" and "Athens" are mentioned nowhere in the previous text is impossible to understand that "She" refers to "Maria" and "there" refers to "Athens".

Maria moved to **Athens**. **She** has lived **there** for 3 months

There have been several methods proposed to resolve different types of pronouns. One, which resolves some types of pronouns in Swedish, the Pronominal Resolution Module (PRM), has been implemented as a text pre-processor, written in Perl language (Hassel, 2000).

Pronominal Resolution Model uses lists of focus applicants, and it is being used as a preprocessor in SweSum. Focus means persons or items that are most prominent at a specific point in the discourse. These are lists that can be seen as stacks of focus applicants who are pushed upon appropriate stack when revealed. In this way, when a nominal phrase is identified it is categorized and placed to a list for that category.

For the time being there are only two focus applicant lists, one for each gender. The choice of an applicant for an anaphor based on salience that's represented by the antecedents position in a list) and semantic likelihood (based on what list the antecedent is to be found in). The second one is determined by using semantic information from a noun lexicon.

PRM is using a noun lexicon, which contains information for every entry's natural or grammatical gender. The noun lexicon used currently contains over 1500 gender specified first names.
The algorithm implemented consists of three different steps acting on three different levels: discourse, sentence level and word level

For every discourse:

- Identify and annotate abbreviations. This step is essential for sentence segmentation.
- Identify and segment sentences.

For every sentence:

- Use special designed templates to identify cases like active or passive phrases.
- If more than one pronoun or anaphoric expressions are found, annotate them using appropriate focus applicant lists.
- Identify and segment words.

For every word:

- Search for a pronoun or anaphoric expression. If one is found then annotate it in AHTML[6] with the most likely antecedent and sentence number found in the corresponding focus applicant list (based on gender and any other given semantic information). Pronouns then are marked with a tag pair
  <! ANAPHOR REF="Referent" LINE="Line number"> and </! ANAPHOR>.
  For example
  <! ANAPHOR REF="Michael" LINE="213">he</! ANAPHOR> shows the antecedent Michael found in line 213.
- Search and compare with the lexicon to see whether the noun exists.
- If the noun is found, place it first in relevant focus applicant list (depending on category). With it include information about the sentence it was found in.

2.2.7 Machine Learning Techniques

This technique is based on a set of texts and their extractive summaries. The process is then modeled as a classification problem where sentences are categorized as summary and non-summary ones, based on the features they posses (Neto et al. 2002). The classification probabilities arise statistically from the training data, using Baye's formula:

$$P(s \in <S \mid F_1, F_2, ..., F_n) = P(F_1, F_2, ..., F_n \mid s \in S) \times P(s \in S) / P(F_1, F_2, ..., F_n)$$

[6] It the HTML standard which Hassel designed specifically to mark anaphors.

Where, s is sentences from the text collection, $F_1, F_2, ..., F_n$ are characteristics that have been used in the classification and P(s $\in$ <S | $F_1, F_2, ..., F_n$) is the possibility that a sentence s, will be picked to form the summary S given that it holds the $F_1, F_2, ..., F_n$ features.

A trainable summarization program that is grounded in a sound statistical framework was developed by (Kupiek et al., 1995). For summaries, 25 percent of the size of the average test document, it selected 84 percent of the sentences chosen by the professionals.

2.2.8 SUMMARIST

This project was developed at the University of Southern California, Institute of Information Sciences (ISI) (Lin and Hovy, 1997). Eduard Hovy was Dalianis supervisor at the same university and the main inspirer for SweSum. The prime target of the SUMMARIST is to provide a good summarization, based on the following formula:
Summarization = topic identification + interpretation + generation
Both extract and abstract summaries can be generated by this system.
Each step contains modules trained on large corpora of text. The first stage filters the given text to determine the most important topics.

This technique combines NLP methods using statistical techniques (extract) with symbolic word knowledge (abstract) provided by dictionaries, WordNet and other resources.

3. SweSum

SweSum (Dalianis, 2000) is a web-based text summarizer, developed at KTH (Kungliga Teckniska Högskolan), Stockholm. Extraction methods are being used based on statistical, linguistic and heuristic methods. It is all implemented in Perl language and the domain is HTML-tagged newspaper text.

Besides Swedish, latest versions include Norwegian, Danish, Spanish, French, German and a version for Farsi language (Iranian). It has been evaluated for Swedish, Danish, Norwegian, French and Farsi with the results being very encouraging. The French, German and Spanish versions are in prototype states (Hassel, 2004).

In the current implementation some of the techniques described previously are being used. Those that require linguistic resources are not available yet. Because only extract summaries algorithms are used, the methods in the SUMMARIST project for generating abstract summaries are not implemented. There are thoughts to include Latent Semantic Analysis (LSA) or Random Indexing (RI) techniques in next versions.

3.1 Architecture

Currently SweSum is implemented as a client/server. On the one side there is an Apache server, where the whole processing of summarization takes place. The user interacts through a browser to send data on the server, where the processing takes place. The results are given as a different page on the browser. Figure 3 shows diagrammatically the process.

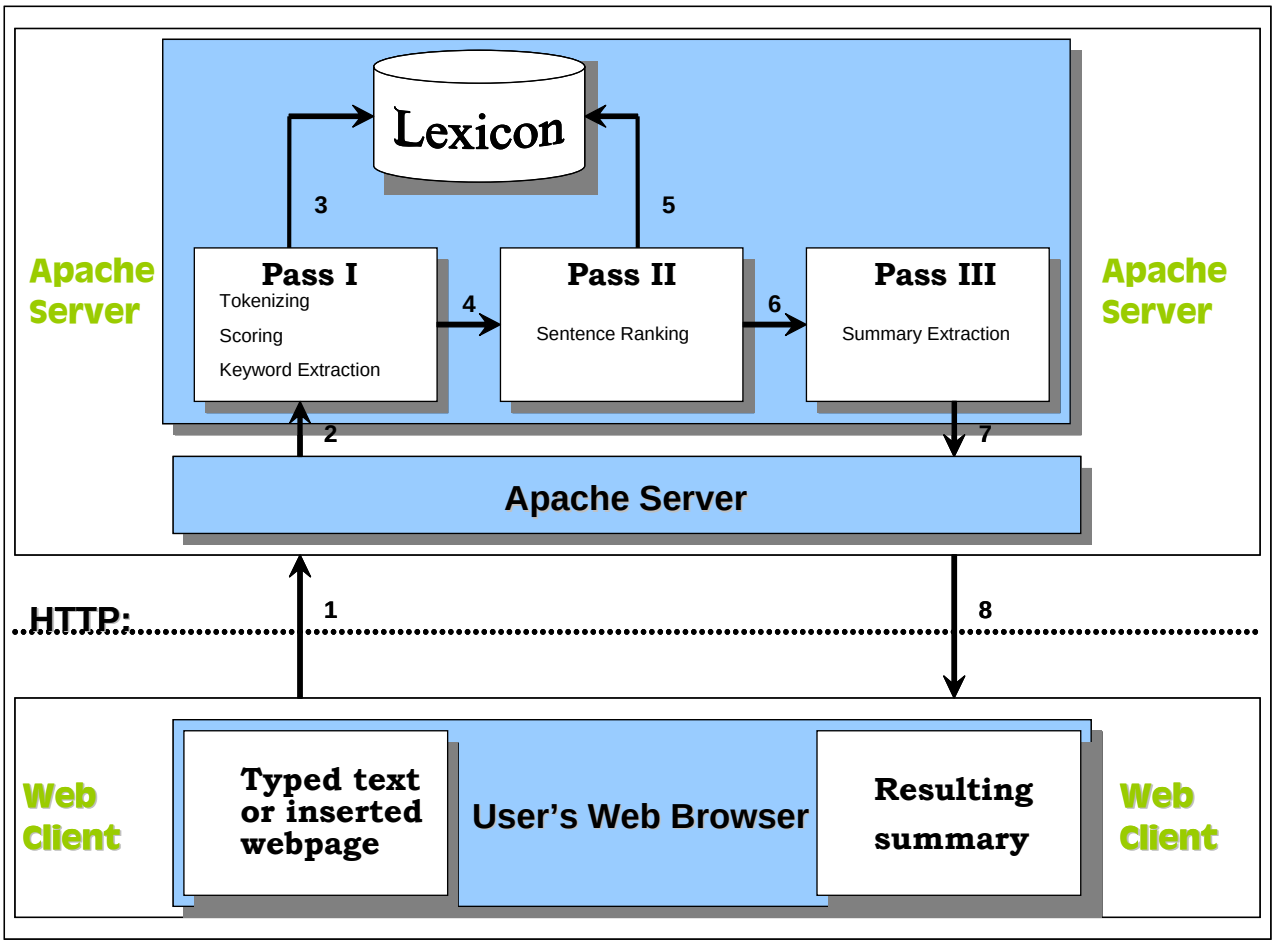

Figure 3: SweSum architecture

According to figure 3 (Mazdak 2004) the summarization process starts when the user types on the special form a text or inserts a URL. In detail the steps are:

1. The users browser sends a request to the Web Server, with the required text or URL to be summarized **(1)**
2. SweSum receives the text **(2)**
3. In several steps the text gets through the program and is being summarized **(3-6)**
4. After the process of summarization is over, the resulting text is returned to the HTTP server **(7)** and from there to the users screen in a new webpage **(8)**

3.2 HTTP

Hyper**T**ext **T**ransfer **P**rotocol is a worldwide protocol that is being used in all computers to enter the World Wide Web. It is used for a variety of actions, for example to view HTML pages, picture files, view videos, get CGI script outputs, etc. Designing websites that interact intelligently with users input is still very hard to implement. It is important to mention that despite its universal use, it is not considered to be a very secure protocol.

3.3 Web Client

The Web Client is the users end and window to the World Wide Web. There are several programs that do this work like Microsoft Internet Explorer, Netscape Navigator, Mozilla, Opera etc.

3.3.1 HTML

HTML is the acronym for **H**yper**T**ext **M**arkup **L**anguage and it is used for all documents on the Internet. The browser interprets and renders the tag-based notation language used to create documents. It is an application of Standard Generalized Markup Language (SGML) and uses tags to mark various elements like text, pictures etc and show how they should be presented on the browser. HTML is also responsible to respond to users actions like mouse click or the press of a button. Currently there are various versions of HTML and World Wide Web Consortium (W3C) works on future developments.

3.3.2 Typed text or inserted webpage

The user has three different options of inserting text for summarization

1. There is a special form field, where the text can be typed in.
2. Any text file or HTML page saved on the user's hard disk or local network can be uploaded and forwarded for summarization
3. A link from the WWW can be typed and the summarization will take place with whatever this link includes.

3.4 Summarizer

The summarization engine that does all the summarization is set up on a Web Server. The process starts as soon as the original text is send in. The whole process is done in three steps, which are called "passes" (Dalianis, 2000).

3.4.1 The Lexicon

The lexicon, used in SweSum, is a data structure for storing key pairs root table. It is a two column file, containing words where the key is the inflected word and the value is the root of the word. The Swedish version contains around 40.000 words and 700.000 different inflections. Also in GreekSum, a Greek root table is being used.

3.4.2 First pass

3.4.2.1 Tokenizing

In this first step the tokenizer goes through the input text, and output is the tokenized text. In more detail it does the following:

- With the "\n" command of Perl, it removes all new line characters.
- All the abbreviations, that are included in a special hashed table, are marked with the tag <! ABBRV>. The abbreviation ex. is marked as <! ABBRV>ex. </! ABBRV>. This process takes place if only there is an available abbreviation table for the given language.
- At an experimental basis currently, it invokes Pronominal Resolution.
- It searches for symbols that mark the sentence boundaries like "**.**" , "**,**" , "**!**" , "**?**" , "**<**" , "**>**" , "**:**" . The symbol for HTML's new line,
 is also taken into consideration. The "\n" is inserted when such boundaries are identified.

After this whole process is over, the result is the inputted text, but in a different form. It includes new line markers after sentence boundaries. To do this the "\n" is being used. Every line is put in a hash table, named text table. The key in the table is the number of the line and the value is what the line contains. An example of it is shown below:

Sentence	Line
<html>	1
<title> Kerry-Bush 1-0 </title>	2
<body>	3
Bush and Kerry took the stage smiling and shook hands -- the only opportunity they are allowed to approach each other, according to the rules of the debate…	4
.....	……
</body>	n-1
</html>	n

Table 1: Text table

3.4.2.2 Keyword Extraction

They are essential for text summarization, but there is a variety of different types used by the research community. SweSum uses the following as keywords: nouns, adjectives and adverbs. In order to extract the keywords two different methods have been developed.

- Tagging and parsing of text
- Using a lexicon

Using a lexicon has been proven to be a lot easier and much faster. One major drawback is that it requires constant update, as the languages develop constantly new words that need to be added. In SweSum, keyword frequency counting is based on the types of keywords described previously, by using a static lexicon. When the program is asking the lexicon for a word, the second returns the lemma of the word that is stored in word frequency hash table. The same words with different inflections are not being counted as different words.

Word	Frequency
Bush	5
Kerry	8
Hands	15
Debate	7
……	…

Table 2: Word frequency table

3.4.2.3 Scoring

In order to determine the importance of every sentence on the text, a special scoring system has to be introduced. The sentences that contain no text are not summarized and are labeled "not text. Those that contain the text to be summarized are labeled with the value "text".

Sentence	Line	Value
<html>	1	Not Text
<title> Kerry-Bush 1-0 </title>	2	Not Text
<body>	3	Not Text
Bush and Kerry took the stage smiling and shook hands -- the only opportunity they are allowed to approach each other, according to the rules of the debate…	4	Text
.....	……	Text
</body>	n-1	Not Text
</html>	n	Not Text

Table 3: Text Table Value

The lines that are marked as “text” are put in a data structure for storing key/value, which is named Text Table Value. Line content is the key to the table and the line number and value is the score of the line. This one depends on the position the line has and how the words in this line are scored. In the final summary, only those high scored are contained.

In SweSum the following is used for calculating the line score:

- **First line:** It should always be included in the summary. This is done by assigning it a very high score, 1000 in SweSum
- **Position:** Because SweSum supports two kinds of text, Newspaper and Report, the position score depends on the text to be summarized. As mentioned before, in newspaper text the first line is the most important and gets the highest score, followed by the others. There is a mathematical formula being used for defining the position score

$$\textbf{Position score} = (1/\text{line number}) \times 10$$

- **Numerical values:** Whenever a number is identified in a text, the line that includes it gets one additional point.
- **Bold Text:** By identifying the <B> symbol in the HTML code, SweSum assigns the score “100” to lines containing bold text. This is because it sometimes shows the beginning of a paragraph or the first sentence of it.
- **Keywords:** They are automatically identified as the most frequent words on the passage. The sentences that contain more keywords take a higher score than those that contain fewer or none.
- **User Keywords:** The play similar role as the keywords described above, but the user defines which words should be used as keywords.
- **Simple combination function:** The different methods described above are combined and used with predefined weights to calculate the score of each sentence.

3.4.3 Second Pass

In this step the score of every word is being identified and added to the sentence score in the text table value.

3.4.3.1 Sentence scoring

$Word\ score = (\text{word frequency}) \times (\text{keyword constant})$

$Sentence\ Score = \sum WordScore$

, where (keyword constant) has the default value of 0,333.

Example: Bush and Kerry took the stage smiling and shook hands –
Word score (Bush) = $5 \times 0.333 = 1.665$
Word score (Kerry) = $8 \times 0.333 = 2.664$
Word score (hand) = $15 \times 0.333 = 4.995$
Sentence score = Word score (hand) + **Word score** (Kerry) + **Word score** (Bush) = 9.324

3.4.3.2 Average Sentence Length

There is always a risk that long sentences will be ranked higher. In order to avoid such a phenomenon the sentence score is multiplied by the Average Sentence Length (ASL) and later divided by number of words in the sentence, for normalization.

WordCount = number of words in the text
LineCount = number of lines in text
Average Sentence Length (ASL) = WordCount / LineCount
Sentence Score = (**ASL**×**Sentence Score**)/ (number of words in sentence)

An example is given below where we assume that the passage contains 20 sentences and 200 words. The Sentence Score for line 5 is calculated.

WordCount = 40 (number of words in lines 1-5)
LineCount = 5
Number of words in sentence = 10
Normal Sentence Score = 12.758
Average Sentence Length (ASL) = WordCount/ LineCount = 40/5 = 8
Sentence Score = (8×12.785)/10 = 10.228

3.4.3.3 Cutoff size and unit

Along with text insertion, the user has the option to insert two kinds of values, cutoff size and unit. The first one is a numerical value and unit is either percent, number of words or characters.

User Input	Summary Result
Cutoff size = 20, unit = percent	Keeps 20% of the words of the original text
Cutoff size = 200, unit = words	Summary contains 200 words

Table 4: Cutoff size and unit example

3.4.3.4 Sorted Text Value

A new array, named sorted text value, is made and contains high value sentences from text table value. The numbers of lines to be used in the final summary are calculated depending to the users input in cutoff size and unit fields.

3.4.4 Third pass

In this third and final pass, the final summary file is created. This is what the user sees and it contains:

- All non HTML lines.
- From the sorted text value, all the lines that have been ranked high.
- Some statistical information like the percentage of the summary, the keywords, number of lines, words etc.

3.5 Evaluation

3.5.1 General Evaluation Rules

Evaluation of an automatic text summarizer is a very difficult process and there are two ways to do it.

- **Manually:** People, who have a good knowledge of the subject language, compare summaries and decide on the best one. This method is depending only on every individual's personal judgment, so it is not very reliable. In an experiment, (Hassel, 2003) found that the agreement between summaries created by different persons was 70 percent in the best case. Another problem with this way of evaluation is that it very time-consuming.

- **Automatic:** As the name indicates, it is an evaluation where the whole process is automatized. The field is yet not well developed and it is a research topic. One method is by using the golden standards. The gold standard summary is made out of the most frequent sentences in a text that is produced by majority votes over of manually created extracts. (Dalianis et al., 2003). This gold standard is used to be compared with summaries created by the automatic summarizer.

  It is assumed that the golden standard is the best possible summary of a text. However this presupposes that there is only one "best summary". The truth is that in summaries there is not "one truth" as it is humans that produce the golden standard.

3.5.2 Evaluating SweSum

SweSum was evaluated in two different ways:

1. **Manual Evaluation**

   This evaluation took place in 2000, within the context of a 4 credit course at NADA/KTH (Hassel, 2004). The experiment took place among nine students who had the task to summarize ten texts, using SweSum. The size of the result summary was constantly being reduced, until coherence was broken and important information started missing. The goal

was to find out how much can a text be summarized without loosing important information.

The result showed that at 30 percent of summary there was still good coherence and at 24 percent there was still good content.

A different manual evaluation was carried out by Fallahi (2003). He conducted his work by comparing 334 texts taken from the newspaper Sydsvenska Dagbladet and summarized in SweSum, with the performance of human editors.

He came to the result that SweSum performed really well, with minor problems that can be focused on the following:

- It sometimes cut sentences by mistake in boundary detection.
- A problem, that has now been fixed, was that sentences of unformatted text were put together in a single paragraph.
- Although the first sentence of one paragraph is ranked higher and always kept for its important information, in some cases it was the second or third sentences to being used. Of course the quality of the summarized text is affected.
- Despite the very small size of an SMS (160 characters) SweSum performed very well when trying to summarize text to such small size.

2. Evaluation Using Extract Corpus

This evaluation was conducted individually by Martin Hassel, (Hassel, 2004). For this purpose he created an extract corpus, which he named KTH eXtract Corpus. This contains original texts and some assets of their manually made extracts. Those extracts are made by several individuals, who were asked to pick a certain number of sentences out of a text, which considered being important.

The KTH extract tool gathers statistics on how many times an important sentence has been included on a number of different summaries. Next step is to generate gold standard summary which is produced by majority votes, containing the most frequent chosen sentences. The SweSum extract summaries were compared sentence-by-sentence with the gold standard extracts.
The result showed that 57.2 percent of the sentences included in the SweSum summaries, were common with the gold standard.

3.6 Special Notes on SweSum

Text's Central Topic
The central topic in the text is identified by using a statistical keyword approach method (keyword frequency), described before. This method is very efficient, but cannot find synonym words in the text. This problem can be solved by using Latent Semantic Analysis or Random Indexing. The method is currently under development.

Cohesion
In SweSum summarization takes place in three steps:

- The given text is broken up to sentences.
- Each sentence is being scored.
- The sentences that get the higher score are those that are being used in the final summary.

Because this technique is mostly statistical, the sentences may not relate to each other and the final text has no meaning.

Redundancy
Some high keyword ranking, in the summarization process, can lead to redundancies in the final text and the summary to be concentrated on only one topic. Latent Semantic Analysis can lead to the reduction of the redundancies.

4. GreekSum

We named GreekSum the version based on SweSum which summarizes Greek newspaper text. The basic principles are the same with SweSum, but some adjustments had to be made in order to support Greek characters.

4.1 History of Modern Greek

Modern Greek is the language spoken present by the Greek people. It is a development of the ancient Greek language, but it differs in several aspects. It is a member of the Indo-European languages and the only member of the Greek subfamily.
Forms of Greek where spoken even before the era of recorded history. Prehistoric tribes moving from central and northern Asia settled in more fertile areas of the Greek peninsula spoke one language, four major dialects of which were: Arcado-Cyprian, Doric, Aeolic and Ionic. The Attic, a standard form of classical Greek, developed from the Ionic dialect. It was spoken in Athens and the surrounding district. Because in 5th century B.C. Athens was dominating with its Art, drama etc. this dialect superseded all others. When Alexander the Great conquered the Middle-East, Greek became the common language to this area. Inevitably, mixtures took place and a new language, named Koine, was formed and being used in all the Greek colonies of the Hellenistic period. Throughout Byzantine and the Ottoman occupation, the language did not develop and only theological works can be found.

After the liberation from the Ottoman Empire in 1821, there was little concern about the language as the nation addressed more important problems. By the end of the 19th century Greek scholars started considering the systemization of a popular tongue for every-day use, called Demotiki. Opposed to them where those who wanted to arise the ancient cultural heritage, and users of the Katharevousa version of Greek.

Katharevousa was created by Adamantios Korais in the early 19th century. This language was something in between Ancient and Modern Greek (Demotiki). During this period there was a lot of conversation around the two forms of Greek language, Archaic and Modern. Katharevousa's main purpose was to fill the gap between these two forms. As the name implies "the clean one" it is free from all the words borrowed from other languages, especially the Turkish one.

In 1976 by a law of the Greek Parliament, Demotiki became the official language, which is until the present day.

In the last few years, a new form of Greek is being used from people all around the world. Because of the difficulties and incompatibilities of Greek fonts, the Internet users started communicating in an easier form, called “Greeklish”. Roman characters are being used to the equivalent Greek which makes communication easier. There are no strict grammatical rules, not even exact matching of non-common characters, as every user does as he pleases.

4.2 Greek Language

4.2.1 Greek Letters

The Greek letters are unique and used only for Greek. Modern and Ancient Greek share the exact same alphabet, which is taken from the Phoenicians and it contains 24 letters. All the letters of the Latin alphabet derive from those of the Greek one, and some are still similar. In detail the Greek letters are:

Upper case	Lower case	English name	Upper case	Lower case	English name
Α	α	alfa	Ν	ν	ni
Β	β	beta	Ξ	ξ	xi
Γ	ɣ	gamma	Ο	ο	omicron
Δ	δ	delta	Π	π	pi
Ε	ε	epsilon	Ρ	ρ	rho
Ζ	ζ	zeta	Σ	σ	sigma
Η	η	Etta	Τ	τ	taf
Θ	Θ	theta	Υ	υ	ypsilon
Ι	ι	iota	Φ	φ	phi
Κ	κ	kappa	Χ	χ	chi
Λ	λ	lambda	Ψ	ψ	psi
Μ	μ	mi	Ω	ω	omega

Figure 4: Greek Letters

Over all the vowels, when needed take an accent, and all the words need to have an accent. Dieresis is another symbol called in more rare cases, sometimes in combination with the accent.

The ancient Greeks were using the same letters as their numbers. It was a very complicated and difficult system to understand. This is the reason why in ancient Greece, geometry was so advanced and not algebra. Thankfully, today the same ten digits as the rest of the world are being in use.

4.2.2 Greek and Unicode

In Unicode, there are two blocks of Greek characters. One is "Greek and Coptic", which is based on ISO 8859-7 and is all that's used in Modern Greek. The second called "Greek Extended" includes the polytonic system used for Ancient Greek and Katharevousa. Microsoft has developed its own encoding, which is named Windows-1253.

Part of the ISO 8859-7 table can be viewed in figure 5:

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
Bx	°	±	²	³	΄	΅	Ά	·	Έ	Ή	Ί	»	Ό	½	Ύ	Ώ
Cx	ΐ	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
Dx	Π	Ρ		Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Ϊ	Ϋ	ά	έ	ή	ί
Ex	ΰ	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
Fx	π	ρ	ς	σ	τ	υ	φ	χ	ψ	ω	ϊ	ϋ	ό	ύ	ώ	

Figure 5: ISO 8859-7

4.2.3 Word and Sentence boundaries

- **Sentence:** In Greek the punctuations that mark the sentence boundaries are almost similar to the rest of the European languages. One big difference is that the sign "?" has absolutely no use in the language and what stands for a question mark is ";". There is a symbol called upper-dot "·" and is used similar to ";" in Latin-based languages. There is no character coding in ISO for the upper-dot character so the "middle-dot" is used instead. However in Unicode, character U+0387, GREEK ANO TELEIA is being used.

- **Words:** Like every language the blank character (space) shows the end of one word. Some long words are been cut in written form, with the symbol "/", and therefore this symbol should not determine a word boundary. The usual parenthesis, brackets, quotes, all kinds of marks, are being used to show a word boundary.

4.3 Component files used for the GreekSum

For the implementation of GreekSum, we were lucky enough to find all the components required by the original SweSum algorithm. These are: the Greek keyword dictionary, the Greek abbreviations table and a Greek proper nouns file. All of them, before being used by GreekSum have to be compiled in a Perl hash table.

4.3.1 Greek Root table

δημοκρίτειων	δημοκρίτειος
δημοκρατία	δημοκρατία
δημοκρατίας	δημοκρατία
δημοκρατίες	δημοκρατία
δημοκρατίζαμε	δημοκρατίζω
δημοκρατίζανε	δημοκρατίζω
δημοκρατίζατε	δημοκρατίζω
δημοκρατίζει	δημοκρατίζω
δημοκρατίζεις	δημοκρατίζω
δημοκρατίζετε	δημοκρατίζω
δημοκρατίζομε	δημοκρατίζω
δημοκρατίζοντας	δημοκρατίζω
δημοκρατίζουμε	δημοκρατίζω

Figure 6: Greek root table

This is a file that contains pairs of words. On the first column is the inflected form of the word and on the second the root of it. The Greek version used for GreekSum contains 549.026 different words all with their corresponding roots. In the second column the corresponding roots are 52.875.
This picture is part of the Greek root table, shows how the file is formed. The specific part contains 8 different forms of the word "δημοκρατία", which stands for the word democracy.

4.3.2 Greek Abbreviations Table

This provides a small part of the Greek abbreviation table. It consists of two columns, where the first one shows the abbreviation and the second the word or phrase it stands for. It contains as much as 400 different abbreviations. As described before the abbreviation table is used by the tokenizer, where every word included in this table is being marked.

μετεπιθ\.	μετεπιθετικός
μεταρ\.	μεταρηματικός
μεταπλ\.	μεταπλασμός
μετακ\.	μετακίνηση
μετάθ\.	μετάθεση
μεσοφ\.	μεσοφωνηεντικός
μειωτ\.	μειωτικός
μεε\.	μετοχή ενεργητικού ενεστώτα
μεγεθ\.	μεγεθυντικός
μαθημ\.	μαθηματικά
μαγειρ\.	μαγειρική

Figure 7: Greek abbreviations table

4.3.3 Greek Proper Nouns

Λουκιανός	m
Λυδία	f
Λύδια	f
Μάγδα	f
Μάκης	m
Μάνος	m
Μάξιμος	m
Μάρθα	f
Μάριος	m
Μάρκος	m
Μίλτος	m
Μίτσος	m
Μαγδαληνή	f

Figure 8: Greek Noun Table

This file contains Greek first names and corresponding sex. It has as much as 368 male and female ones and a small part of it is given on the table on the left.

4.4 Implementation of GreekSum

As the SweSum, the GreekSum is implemented in Perl language and installed on an Apache Server. We were lucky enough to be able to find the tables mentioned above (Root-table, Abbreviations and Proper Nouns). Several changes and modifications had to be made to support the Greek language, both in the Perl code and the Apache server.

4.4.1 Apache Server

Apache server is a project started in 1995. It is Open Source code and totally free. The idea is that a core of few people started working on it, but the source is available to anyone. Users from around the world can send their ideas and improvements, which are taken under consideration. If the contribution is worthy, the new release contains it. These users are called the Apache Group, and there are hundreds of them who actively participate in the project.

The Apache version used for GreekSum is 2.0.50. It can be downloaded from the www.apache.org website. It is very easy, even for an inexperienced user to install it and directly start working with it.

Every modification considering the server can be done by one simple text file: httpd.conf. It is written in English language and is very easy to adjust. The necessary change that needed to be made was to set the path which contains the Perl code. Any folder can be used, but Apache has one called /cgi-bin, that is made specially to be used for scripts. This is the one we used.

The web-interface of GreekSum is in Greek; therefore the server had to be modified to support this. A simple command “AddDefaultCharset” had to be modified, by setting the value to ISO 8859-7, so it became “AddDefaultCharset ISO 8859-7”.

4.4.2 Perl

Perl is a programming language launched in 1987 by Larry Wall (Wall et al. 2000). Its main feature is that it supports both procedural and object oriented programming. As mentioned in the www.perl.org website: "Perl takes the best features from other languages, such as C, awk, sed, sh, and BASIC, among others". There is a web-based library named Comprehensive Perl Archive Network (CPAN) that contains many, free, modules which make Perl very extensible. Finally what was important for the implementation of GreekSum and FarsiSum, is that it supports Unicode.

For the framework of the Master Thesis the most work had to be done in adjusting the SweSum code so it can make use of the Greek lexicon and actually work for the Greek language. The major problem we had to solve was that with the current implementation of SweSum no keywords where visible in the final summary. This was solved by converting all the files to be used in extended ASCII format, which is the only format that SweSum and Perl recognizes.

4.4.3 Web Interface

The interface, which is the front-page of GreekSum, consists of two pages. The user can choose between typing the text himself, just inserting a URL for summarization or uploading a text file saved in the hard disk. Optional functions of setting keywords, defining summarization percentage etc. are also provided.

Because the page uses ISO 8859-7 encoding, the user should have the proper encoding on his/hers computer to view it. The HTML code has been modified so the encoding automatically changes to the proper one. In order to do this, the following line had to be added to the header:

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-7">

Screenshots of the interface can be found on Appendix A.

4.5 Other Greek Summarization Applications

By conducting a search on the Web, one can find many applications that conduct summaries. Some worth mentioning are: the one built in the latest versions of Microsoft Word. Copernic, Inxight, Open Text Summarizer and many others. Most of these products are commercial and have several functionalities, like summarizing web pages, highlighting key sentences etc. None of the products mentioned above had special functions for Greek, but some claim that have full UTF-8 encoding support. It was out of the scope of this thesis to check how each ones perform for the Greek language.
One product we found that clearly stated that supports the Greek language is "Subject Search Summarizer™" a commercial product, which was developed by "Kryloff Technologies". From a brief test conducted, it worked well but had some problems with the Greek encoding.

The most famous of above products, Microsoft Office Word, supports summaries only in the following languages: Chinese (traditional and simplified), English, French, German, Italian, Japanese, Korean, Portuguese (Brazil), Spanish and Swedish.

5. Evaluation

5.1 The Method

In order to evaluate the GreekSum we used eight different Greek texts, of various contents, and compare two different summaries. The first one is what is given by the GreekSum and the other one is the result of the text put in the Generic mode of SweSum. This was done by four Greek native speaker students, who were separated in two groups, each one evaluating four texts. It is important to mention that on the Generic mode of SweSum the keywords are not visible for the Greek texts.

The texts used were taken from the Greek news website: http://www.in.gr The content was primarily Greek domestic and science news. The longest text had educational-mathematical content.

The two results are compared according to the following four points:

1. Which summary is better?
2. In which one the most important information is being kept?
3. Which summary is more coherent?
4. Out of a scale from 1-5, where 5 is the best, what score would you assign to each summary?

In the next table some information about the text appears:

	No. of Words in the original text	% of Summary tested
Text 1(T1)	3,469	10
Text 2(T2)	266	30
Text 3(T3)	723	30
Text 4(T4)	287	30
Text 5(T5)	259	40
Text 6(T6)	605	20
Text 7(T7)	859	15
Text 8(T8)	1,435	10

Table 5: Evaluation text information

We call the two different methods M1 and M2 where:
M1: Usage of the GreekSum
M2: Usage of Generic

5.2 The Results

In the following section there are some statistics on what the evaluators thought of the two different kinds of summarizations. These are based on the analytical results, which can be found on Appendix B.

1. Which summary is betterr?

The results show, that at total of 93.75 percent the GreekSum gave better results than the Generic mode of SweSum. In the small summaries it performed really well but when it comes to the 3,469 words text, the gap result is not that good.

	M1 GreekSum	**M2 Generic**
T1	50%	50%
T2	100%	0%
T3	100%	0%
T4	100%	0%
T5	100%	0%
T6	100%	0%
T7	100%	0%
T8	100%	0%
AVG	**93.75%**	6.25%

Table 6: Best method result table

2. In which one the most important information is being kept?

In this case GreekSum performed better by a 2.15 percent from the Generic mode. In all texts evaluators judged that the two machines performed almost the same.

	M1 GreekSum	**M2 Generic**
T1	50%	50%
T2	50%	50%
T3	50%	50%
T4	50%	50%
T5	50%	50%
T6	50%	50%
T7	50%	50%
T8	66.6%	33.3%
AVG	**52.075%**	49.925%

Table 7: Information preserved result table

3. Which summary is more coherent?

Like in the previous case, the GreekSum gave similar results to the Generic mode. There is a slight aberration 8.3 percent in the total result.

	M1 GreekSum	M2 Generic
T1	50%	50%
T2	50%	50%
T3	50%	50%
T4	66.6%	33.3%
T5	50%	50%
T6	50%	50%
T7	50%	50%
T8	66.6%	33.3%
AVG	**54.15%**	45.85%

Table 8: Coherent information result table

4. Out of a scale from 1-5, where 5 is the best, what score would you assign to each summary?
   In this table the results from the users are turned into statistics, according to the added score of the two results and compared on a scale of 100. With this method, GreekSum granted an overall 58 percent.

	M1 GreekSum	M2 Generic
T1	58.3%	41.7%
T2	55.5%	44.5%
T3	58.8%	41.2%
T4	58.8%	41.2%
T5	56.25%	43.75%
T6	64.3%	35.7%
T7	56.25%	41.2%
T8	56.25%	41.2%
AVG	**58%**	42%

Table 9: 1-5 rating result table

The overall results show that GreekSum gave better summaries than the Generic mode. The keyword function, that is available in all the versions of SweSum, but not in Generic mode for Greek language, seemed to have played a crucial role in the judgment of the result. Otherwise, it is important to mention that the content of the two summaries in most cases was identical

Limitations of method.

The method followed is similar to the one (Mazdak, 2004) followed for the evaluation of FarsiSum. Additionally there is no availability of Greek corpus and it is not possible to gather

many Greek native speakers in Sweden to evaluate the system for the limited time of this Master thesis.

The results are totally subjective and according to every user personal judgment. An evaluation similar to the one (Dalianis and Hassel, 2001) conducted for SweSum can be part of future work.

6. Conclusions and Future Work

6.1 Conclusions

As mentioned in the previous paragraph much can be done in the area of evaluating the text summarizer, which there was no time to do during the period of the Master thesis. A selection of Greek corpus text and a bigger number of native Greek speakers can be used to evaluate GreekSum using a method similar to the one Dalianis and Hassel (2001) used to evaluate SweSum.

The initial step for this thesis work was to read several papers published on the field, which was enjoyable and gained lot of knowledge. Most of this work was done by using the Internet. When given the code, some experiments have to be conducted using the (SweSum, 2005) with English and Greek texts so we get an understanding of how the system works. This was done partially by the above method and by reading and understanding the Perl code. The task was hard as there was not a relevant programming language background. The biggest problem we encountered and took the most of the time was to insert the lexicons in the program. Through a series of meetings, we understood that the only format that is recognized internally in the SweSum engine is ASCII and therefore we had to use this format to insert the several components. The specific encoding of extended ASCII we used in our case was ISO 8859-7. Final step was the evaluation which was pleasantly done by 4 students. During this period several comments were given which helped understanding some problems from another perspective. Results showed that GreekSum performs well in texts that are 20-30 percent summarized, just as the SweSum does.

Apart from the GreekSum, more languages can be included in the SweSum family. The algorithm used is basically similar and can be easily adjusted to serve the needs of different languages. In order to better handle the different encodings of the several languages, a big improvement would be to rewrite the SweSum in another, object oriented, programming language like Java.

Every member of the SweSum family lacks the ability to summarize web-pages when there are pictures, links and other irrelevant to the text elements. A solution to that may be a program that would understand the layers and frames on a webpage, find the one that contains the text, summarize it and return the original page with the summarized text in the proper frame.

For the Greek version it is important to mention that we do not make any use of the proper nouns file after all. It did not affect much the quality of the resulting summaries and additionally from a study it was found that named entity recognition has to be used with consideration in order for the summary not to be too difficult for the reader to follow (de Smedt et al. 2005)

6.2 Future Work

More advanced method to implement in future versions is the usage of abstract summarization methods, synonym resolution and pronoun resolution which the resulting summary is an interpretation of the original text. The results will be much more coherent but this method is difficult to implement.

The overall result is very satisfactory. SweSum is very flexible and the problems encountered with the current version, which had to overcome the encoding of the different language, open the way for new versions in a number of languages. Small adjustments can be easily made on the existing code to support other differences (for example the Greek question mark).

Finally as resulted from the brief evaluation, GreekSum worked really well. Of course there are problems and it is not perfect but has added much to an existing gap in the Greek community. Being the first complete machine for this language and since it is free on the Internet for everyone to use, we hope all those that have access to it can find a useful tool.

References

Boguraev, B. and Kennedy, C. 1997. *Salience-based Content Characterization of Text Documents.* In Proceedings of the Workshop on Intelligent Scalable Text Summarization at the ACL/EACL Conference, 2-9. Madrid, Spain.

Brunn, M., Chali, Y. and Pincha, C.J. 2001. *Text summarization using lexical chains, in Document Understanding* Conference (DUC), New Orleans, Louisiana USA, September 13-14, 2001.

Dalianis, H. 1999. *Aggregation in Natural Language Generation*, Journal of Computational Intelligence, Volume 15, Number 4, pp. 384-414, November 1999.

Dalianis, H. 2000. *SweSum - A Text Summarizer for Swedish* http://www.dsv.su.se/~hercules/papers/Textsumsummary.html TRITA-NA-P0015, IPLab-174, NADA, KTH, October 2000

Dalianis, H. and Hassel, M. 2001. *Development of a Swedish Corpus for Evaluating Summarizers and other IR-tools*. Technical report, TRITA-NA-P0112, IPLab-188, NADA, KTH, June 2001

Dalianis, H. and Åström, E. 2001. *SweNam - A Swedish Named Entity recognizer, its construction, training and evaluation*. Technical report, TRITA-NA-P0113, IPLab-189, NADA, KTH.

Dalianis, H., Hassel, M., de Smedt, K., Liseth, A., Lech, T.C. and Wedekind, J. 2003. *Porting and evaluation of automatic summarization*. In Holmboe, H. (ed.) Nordisk Sprogteknologi 2003. Årbog for Nordisk Språkteknologisk Forskningsprogram 2000-2004 Museum Tusculanums Forlag 2004.

de Smedt, K.,A. Liseth, M. Hassel, H. Dalianis 2005. *How short is good? An evaluation of automatic summarization*. In Holmboe, H. (ed.) Nordisk Sprogteknologi 2004. Årbog for Nordisk Språkteknologisk Forskningsprogram 2000-2004 Museum Tusculanums Forlag.

Edmundson, H.P. 1969. *New Methods in Automatic Extraction*. Journal of the ACM 16(2) pp 264-285.

Fallahi, S. 2003. *Presentation at Fifth ScandSum network meeting*, Jan 25-28, 2003, Norway. http://www.dsv.su.se/~hercules/scandsum/OHSasanFeforJan2003.pdf

Gong, Y. and Liu, X. 2001. *Generic text summarization using relevance measure and latent semantic analysis*. In Proceedings of SIGIR 2001, page 19-25.

Hassel, M. 2000. *Pronominal Resolution in Automatic Text Summarisation*. Master Thesis, University of Stockholm, Department of Computer and Systems Sciences (DSV).

Hassel, M. 2003. *Exploitation of Named Entities in Automatic Text Summarization for Swedish.* In Proceedings of NODALIDA 03 - 14th Nordic Conference on Computational Linguistics, May 30-31 2003, Uppsala, Sweden.

Hassel 2004. *Evaluation of Automatic Text Summarization, a practical implementation*. Licentiate thesis of Martin Hassel. KTH Stockholm, Sweden, 2004.

Karlgren, J. and Sahlgren, M. 2001. *Vector-based Semantic Analysis using Random Indexing and Morphological Analysis for Cross-Lingual Information Retrieval,* Technical report, SICS.

Landauer, T., Laham, K. and Foltz, D. 1998. *Learning human-like knowledge by Singular Value Decomposition:* A progress report. In M. I. Jordan, M. J. Kearns & S. A. Solla (Eds.), Advances in Neural Information Processing Systems 10, (pp. 4551). Cambridge: MIT Press.

Lin, C.Y. 1995. *Knowledge Based Automated Topic Identification*. In the Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics. Cambridge, Massachusetts, USA, June 1995.

Lin, C.Y. and Hovy, E. 1997. *Identify Topics by Position*, Proceedings of the 5th Conference on Applied Natural Language Processing, March.

Lin, C.Y. 1999. *Training a Selection Function for Extraction*. In the 8th International Conference on Information and Knowledge Management (CIKM 99), Kansa City, Missouri, November 2-6, 1999.

Luhn, H.P. 1959. *The Automatic Creation of Literature Abstracts*. IBM Journal of Research and Development pp 159-165.

Mani, I. and Maybury, M. 1999. *Advances in Automatic Text Summarization*, MIT Press, Cambridge, MA, 1999. ISBN: 0262133598, Publisher: MIT Press

Maybury T. M. and Inderjeet M. 2001, *Automatic Summarization Tutorial Notes for the American/European Conference on Computational Linguistics (ACL/EACL '01)*. Toulouse, France 8 July 2001

Mazdak 2004. *Farsi-Sum, a Persian Text Summarizer*, Master Thesis of Nima Mazdak. Stockholm University, Sweden, 2004.

Neto, L., Freitas, A. and Kaestner, C. 2002. In G Bittencourt and GL Ramalho, editors, Proc. 16th Brazilian Symp. on Artificial Intelligence (SBIA-2002). *Lecture Notes in Artificial Intelligence* 2507, pages 205-215. Springer-Verlag, November 2002.)

Swesum 2005 *SweSum-Automatic text summarizer* demonstrator http://swesum.nada.kth.se

Wall L., Christiansen T. and Orwant J. 2000, *Programming Perl*, Third edition, ISBN: 0596000278, Publisher: O'Reilly Media, Inc.

Wiemer-Hastings, P. 1999. *How Latent is Latent Semantic Analysis?* in Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99), Stockholm, 1999.

Other Sources

(All links checked February 2005)

Information from Microsoft Encarta Standard Encyclopedia, Edition 2003.
© ® 1993-2002 Microsoft Corporation.

Information from Open-Encyclopedia. open-encyclopedia.com,
© 2003-2004

Information from Apache Server Website. www.apache.org

Information about the Open Text Summarizer, from the website: http://libots.sourceforge.net/
Inxight SmartDiscovery™, Search and Summarization. Information from the company website http://www.inxight.com/ © 2005 Inxight Software, Inc. All Rights Reserved.

List of languages Microsoft Word supports summarization from: Information retrieved from http://office.microsoft.com website © 2004 Microsoft Corporation. All rights reserved.

Subject Search Summarizer™, Product Highlights. Text and test of trial version from product website http://www.kryltech.com/summarizer.htm Copyright © 1997-2005 Kryloff Technologies, Inc. All Rights Reserved.

Text parts from the Greek News website: http://www.in.gr used for summarization and example purposes.

The Copernic summarizer, Product Overview,
http://www.copernic.com/en/products/summarizer/ © 2004 Copernic Technologies, Inc. All rights reserved.

APPENDIX A : User Interface

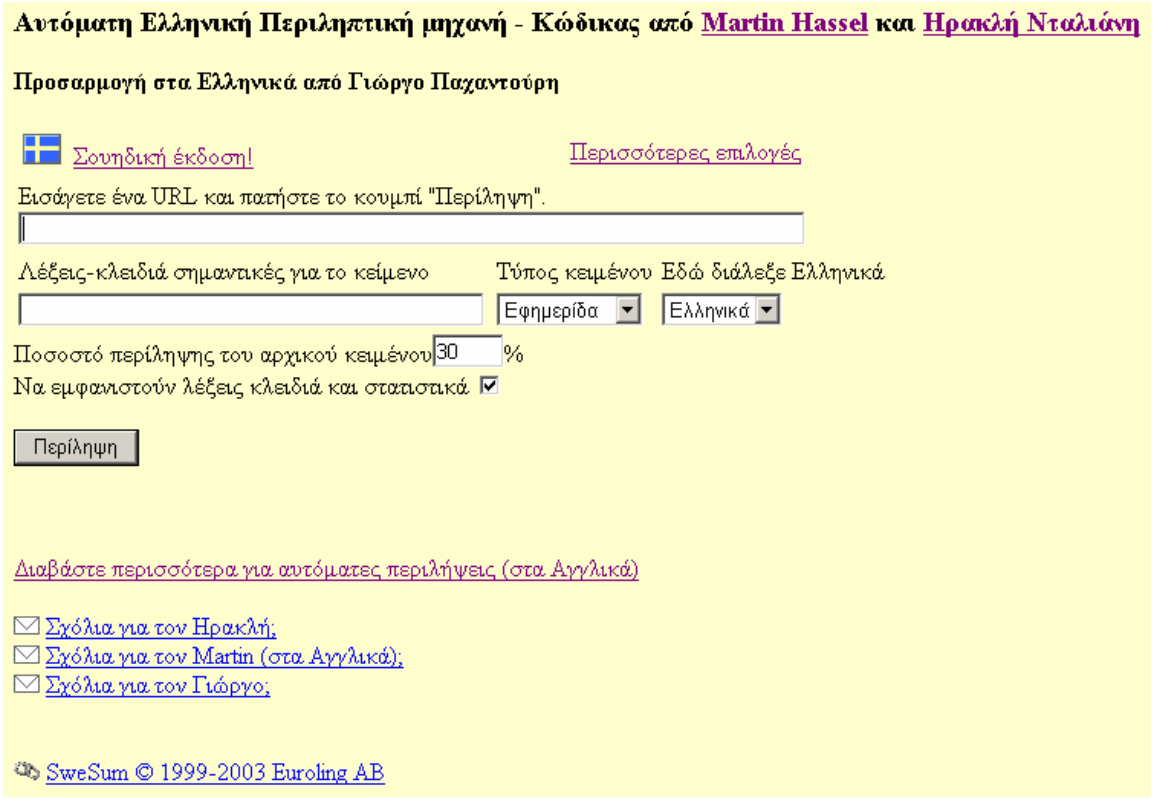

Figure 9: User Interface 1

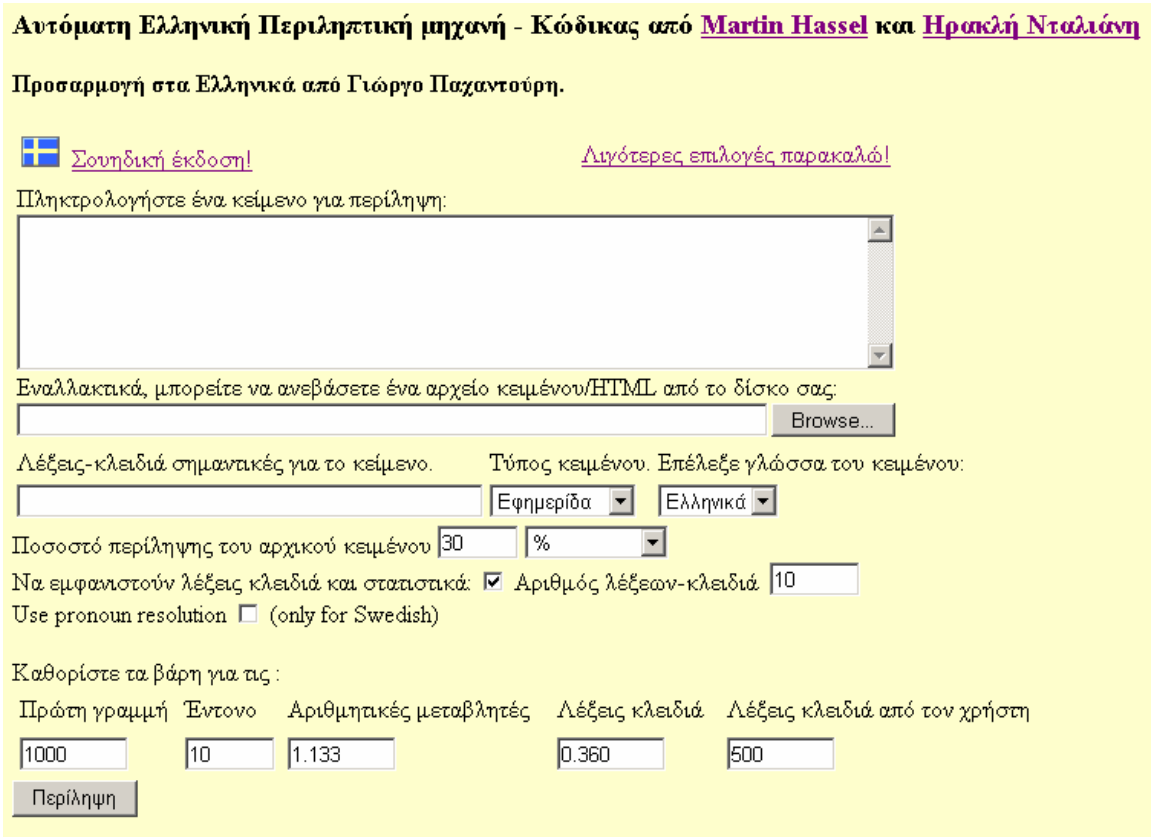

Figure 10: User Interface 2

APPENDIX B: Evaluation Results

Tables used for evaluation

As described in section 5.1 the following hold
T1,...,T8 are the eight texts
M1,M2 are the two different methods used, GreekSum and Generic
U1,…,U4 are the five different evaluators

Table 1: Which method is better?

	U1	**U2**	**U3**	**U4**	**Result**
T1	M1	M2	-	-	M1(1),M2(1)
T2	M1	M1	-	-	M1(2)
T3	M1	M1	-	-	M1(2)
T4	M1	M1	-	-	M1(2)
T5	-	-	M1	M1	M1(2)
T6	-	-	M1	M1	M1(2)
T7	-	-	M1	M1	M1(2)
T8	-	-	M1	M1	M1(2)

Table 10: Best method-analytic

Table 2: In which one the most important information is being kept?

	U1	**U2**	**U3**	**U4**	**Result**
T1	M1,M2	M1,M2	-	-	M1(2),M2(2)
T2	M1,M2	M1,M2	-	-	M1(2),M2(2)
T3	M1,M2	M1,M2	-	-	M1(2),M2(2)
T4	M1,M2	M1,M2	-	-	M1(2),M2(2)
T5	-	-	M1,M2	M1,M2	M1(2),M2(2)
T6	-	-	M1,M2	M1,M2	M1(2),M2(2)
T7	-	-	M1,M2	M1,M2	M1(2),M2(2)
T8	-	-	M1	M1,M2	M1(2),M2(1)

Table 11: Information preserved result table-analytic

Table 3: Which summary is more coherent?

	U1	U2	U3	U4	Result
T1	M1	M2	-	-	M1(1),M2(1)
T2	M1,M2	M1,M2	-	-	M1(2),M2(2)
T3	M1,M2	M1,M2	-	-	M1(2),M2(2)
T4	M1,M2	M1	-	-	M1(2),M2(1)
T5	-	-	M1,M2	M1,M2	M1(2),M2(2)
T6	-	-	M1,M2	M1,M2	M1(2),M2(2)
T7	-	-	M1,M2	M1,M2	M1(2),M2(2)
T8	-	-	M1	M1,M2	M1(2),M2(1)

Table 12: Coherent information result table-analytic

Table 4: Out of a scale from 1-5, where 5 is the best, what score would you assign to each summary?

	U1	U2	U3	U4	Total Result
T1(M1)	4	3	-	-	7
T1(M2)	3	2	-	-	5
T2(M1)	5	5	-	-	10
T2(M2)	4	4	-	-	8
T3(M1)	5	5	-	-	10
T3(M2)	3	4	-	-	7
T4(M1)	5	5	-	-	10
T4(M2)	3	4	-	-	7
T5(M1)	-	-	4	5	9
T5(M2)	-	-	3	4	7
T6(M1)	-	-	4	5	9
T6(M2)	-	-	2	3	5
T7(M1)	-	-	5	4	9
T7(M2)	-	-	4	3	7
T8(M1)	-	-	5	4	9
T8(M2)	-	-	4	3	7

Table 13: 1-5 rating result table-analytic

APPENDIX C: Example Summary

In this appendix we present an example summary. (Article used from news website http://tovima.dolnet.gr, 09/01/2005)

Original text:

Η γλώσσα ως δημόσιο αγαθό

Δ. ΔΗΜΗΤΡΑΚΟΣ

Η γλώσσα είναι ένα δημόσιο αγαθό - μάλιστα το κατ' εξοχήν δημόσιο αγαθό, εφόσον αποτελεί το κύριο επικοινωνιακό εργαλείο μιας κοινωνίας, ενώ συγχρόνως λειτουργεί και ως ιστός της ενότητάς της. Υπάρχουν και άλλα δημόσια αγαθά, όπως είναι το καθαρό περιβάλλον, το νομικό σύστημα, η πολιτιστική κληρονομιά ή η ασφάλεια μιας χώρας. Δημόσιο λογίζεται ένα αγαθό η χρήση του οποίου από κάποιον δεν συνεπάγεται την ανάγκη αποκλεισμού άλλων - όπως γίνεται στην περίπτωση ιδιωτικών αγαθών.

Το σημαντικό με τα δημόσια αγαθά είναι ότι η προστασία τους είναι δημόσια υπόθεση. Δεν αφήνεται ένα δημόσιο αγαθό στο έλεος του ατομικού κεφιού και της προσωπικής προτίμησης του καθενός. Η πολιτεία, μάλιστα, προστατεύει με αρκετό ζήλο τη δημόσια περιουσία, είτε πρόκειται περί χρήματος, είτε περί ακινήτων, είτε περί τηλεοπτικών συχνοτήτων. Δεν υπερασπίζεται με τον ίδιο ζήλο τα δημόσια αγαθά, διότι ο χαρακτήρας τους ως δημόσια κτήση έχει πιο αφηρημένο χαρακτήρα. Στο πεδίο αυτό, το κράτος αρκείται στην επιβολή ορισμένων απαγορεύσεων, χωρίς να αντανακλά η νομοθεσία ή να συνειδητοποιεί η πολιτική τάξη την ύπαρξη δημόσιων αγαθών που αποτελούν ιδιοκτησία της κοινωνίας στο σύνολό της και έχουν επιτακτική ανάγκη προστασίας.

Είναι ανάγκη να δει κανείς τα δημόσια αγαθά ως μέρος της κοινής ιδιοκτησίας, διότι μόνο έτσι γίνεται αντιληπτή ως αδικοπραξία η εμπρόθετη ή απρόθετη καταστροφή τους. Η καταστροφή αυτή δεν γίνεται άμεσα αντιληπτή, ιδίως όσον αφορά τη γλώσσα, διότι επέρχεται βαθμιαία και ως εκ τούτου δεν γίνεται άμεσα αισθητή σε όλους.

Η ελληνική γλώσσα, το πολύτιμο αυτό δημόσιο αγαθό, καταστρέφεται μέρα με τη μέρα, μπρος στα μάτια μας, χωρίς να εξεγείρονται παρά ελάχιστοι συμπολίτες μας. Είναι δυνατό να αντιτάξει κανείς σ' αυτό ότι εκείνο που οι «γλωσσαμύντορες» ονομάζουν «καταστροφή» της γλώσσας είναι η απρόσωπη αυτό-δημιουργικότητά της, που είναι πηγή της δυναμικής της και ότι η δυναμική αυτή δεν μπαίνει σε καλούπια που φτιάχνουν οι γραμματικοί. Η απάντηση είναι ότι οι τελευταίοι δεν καταγράφουν απλώς, αλλά έχουν και κανονιστικό ρόλο - ακριβώς όπως ο χαρακτήρας ενός λεξικού είναι ρυθμιστικός και όχι μόνο περιγραφικός. Και αυτό ισχύει ιδιαίτερα για την Ελλάδα και την ελληνική γλώσσα.

Στο σημείο αυτό, όμως, η είσοδος στον κόσμο της νεωτερικότητας υπό συνθήκες ραγδαίας παγκοσμιοποίησης απειλεί τη γλώσσα μας. Δεν εννοώ ότι τα αγγλικά θα την εκτοπίσουν. Ούτε διακατέχομαι από οποιαδήποτε μορφή πολιτιστικής απαισιοδοξίας ή μελλοντοφοβίας. Πιστεύω, όμως, ότι η γλώσσα εξελίσσεται άναρχα και δεν συμβαδίζει η εξέλιξη αυτή με την εξέλιξη εκείνου που είναι σημαντικό στην πρόοδο της κοινωνίας που είναι η επιστήμη, οι τέχνες και τα γράμματα. «Δημιουργοί» της γλώσσας είναι μάλλον οι πρωταγωνιστές της

γλωσσικής αναρχίας με αποτέλεσμα να απειλείται η γλώσσα μας με κάτι ανάλογο με αυτό που ο Durkheim ονομάζει «ανομία» (anomie) όταν δεν ισχύει κανένας κανόνας σε μια κοινωνία. Συγχρόνως παρατηρούμε αυξανόμενα κρούσματα ακυρολογίας στα ΜΜΕ, στα σχολεία, στα πανεπιστημιακά έδρανα και στη Βουλή των Ελλήνων με φόντο την απόλυτη αδιαφορία του κοινού μπρος στο γλωσσικό αλαλούμ. Συμπεριφερόμαστε προς τη γλώσσα, σαν να ήταν ένας απέραντος χώρος για ελεύθερη τσουλήθρα όπου ο καθένας πάει όπου θέλει, όπως θέλει. Η γλώσσα φθείρεται και τελικά καταστρέφεται όταν οι χρήστες της δεν σέβονται τις μορφές της και ορισμένους παγιωμένους τύπους της - στοιχεία που συνδέονται με τη δύναμή της και επομένως με τις δικές μας επικοινωνιακές δυνατότητες.

Αν θεωρήσουμε ότι η γλώσσα μας είναι δημόσιο αγαθό, όπως ο αέρας που αναπνέουμε, θα πρέπει η πολιτεία να λάβει ορισμένα μέτρα, τουλάχιστον στα ηλεκτρονικά ΜΜΕ, που είναι οι πιο κεντρικοί και ισχυροί διαμορφωτές γλωσσικών τύπων. Διότι μέσα από αυτά αν δεν παράγονται, πάντως αναπαράγονται οι γλωσσικές φρικαλεότητες - ο «ΟκτώΜβριος», ο «ΣεΜπτέβριος» και άλλα ανάλογα με τα οποία, όσοι από μας έχουν διατηρήσει κάποια γλωσσική ευαισθησία, έχουν κυριολεκτικά «απΗυδήσει».

Ο κ. Δημήτρης Δημητράκος είναι καθηγητής Πολιτικής Φιλοσοφίας στο Πανεπιστήμιο Αθηνών.

Figure 11: Summary example-Original text

Summary:

Η γλώσσα ως δημόσιο αγαθό

Δ. ΔΗΜΗΤΡΑΚΟΣ

Η γλώσσα είναι ένα δημόσιο αγαθό - μάλιστα το κατ' εξοχήν δημόσιο αγαθό, εφόσον αποτελεί το κύριο επικοινωνιακό εργαλείο μιας κοινωνίας, ενώ συγχρόνως λειτουργεί και ως ιστός της ενότητάς της.

Το σημαντικό με τα δημόσια αγαθά είναι ότι η προστασία τους είναι δημόσια υπόθεση. Δεν αφήνεται ένα δημόσιο αγαθό στο έλεος του ατομικού κεφιού και της προσωπικής προτίμησης του καθενός. Δεν υπερασπίζεται με τον ίδιο ζήλο τα δημόσια αγαθά, διότι ο χαρακτήρας τους ως δημόσια κτήση έχει πιο αφηρημένο χαρακτήρα.

Η ελληνική γλώσσα, το πολύτιμο αυτό δημόσιο αγαθό, καταστρέφεται μέρα με τη μέρα, μπρος στα μάτια μας, χωρίς να εξεγείρονται παρά ελάχιστοι συμπολίτες μας.

Ο κ. Δημήτρης Δημητράκος είναι καθηγητής Πολιτικής Φιλοσοφίας στο Πανεπιστήμιο Αθηνών.
Πρωτότυπο κείμενο

Λεξικό: Ελληνικό
Λέξεις πριν 676
Λέξεις μετά 178
Ποσοστό περίληψης: 26%
Είδος κειμένου: Κείμενο εφημερίδας
Λέξεις κλειδιά: *δημόσιος γλώσσα γλωσσικός κοινωνία ανάγκη ορισμένος δυναμική ανάλογος σημαντικός ελληνικός*
<-- Προηγούμενη σελίδα

Figure 12: Summary example-Summarized text

Some comments on the above summary: Most important is to mention that the example used is not totally random as we tried to use something quite small for the sake of easy reading. The result is very satisfying with the most important information, like the first sentence and the title, being kept in the summarized text. Additionally the content is very clear and the basic meaning of the original text is kept. No illegal breaks are found and the rules of the punctuation are used. The percentage of the summary we used was 30 but the result was 26. We also tried (and not present here) to summarize the text with smaller and bigger percentage and it performed also very well. Finally the keywords were all found in the dictionary and their roots were successfully retrieved. This is indicated by the fact that they are printed in italics.