Enhancing a Greek Language Stemmer

-

Efficiency and Accuracy Improvements

Spyridon Saroukos

University of Tampere
Department of Computer Sciences
Computer Science M.Sc. thesis
Supervisor: Eleni Berki
July 2008

University of Tampere

Department of Computer Sciences

Spyridon Saroukos: Enhancing a Greek Language Stemmer - Efficiency and Accuracy Improvements

M.Sc. thesis, 48 pages, 20 Appendix pages

July 2008

Stemming algorithms are used in the field of Information Retrieval in order to improve precision and recall. Although for Greek there are three stemmers published, only one of them is freely available. In this thesis, we use stemmer performance metrics for evaluating the existing algorithm and we improved its accuracy and completeness. These improvements were achieved by providing an alternative implementation in PHP which offers more syntactical rules and exceptions. Finally, the two algorithms are tested and their statistics metrics are compared.

Key words and terms: stemming algorithm, Greek, algorithmic efficiency, PHP

Table of Contents

Index of Tables

1. Introduction

Search engines play a critical role in people's life nowadays. Google reported back in 2006 that it receives more than 100 million queries from US based hosts daily [Witten *et al*., 2007]. For end users, they are perhaps the only way to find the information they seek and to navigate to the appropriate web pages. In addition, the users of search engines tend to navigate through the top results that the search engines return to them. It is quite common that end users know which piece and what type of information they seek but they are quite often unable to construct a proper query that will fully describe their request. As a result, malformed and badly structured queries tend to return fewer and more irrelevant results than well/structured queries. For example, someone seeking information for “World War II” may form a query as “World Wars”. Since “Wars” is not a part of “War”, documents containing the reference “War” will not be returned and, thus, the number of relevant results will be reduced. One of the solutions proposed to increase the ratio of relevant documents to total documents retrieved, also known as *recall*, is *stemming.*

1.1 Stemming

Stemming is the process of reducing words to their stem, base or root form, [Lovins, 1968] as shown in Table 1.

Table 1: Stemming Examples

in English	
Original word	**Stem**
Seas	sea
Wars	war
determination	determin-
Developed	develop
in Greek	
Αναπνέω	αναπνέ-
Ελληνικός	ελληνικ-

This is a simple but effective operation used in the fields of information extraction and natural languages [Carlberger *et al*., 2001]. Stemming can be utilized when storing information about a web page in a search engine's database or for *query expansion* [Xu & Croft, 1998]. In that case, the user's query is evaluated and reformulated. The search engine may reduce the word to its stem and thus return more results to the user. An evaluation research in 1981 showed that stemming improved search precision [Brants, 2003].

1.2 Definitions of Key Terms and Concepts

As stemming is also a linguistic process, any discussion of it, even in the field of IR, assumes the knowledge of some basic linguistic terms. The ones used in this thesis are described in Table 2.

Table 2: Definitions of key terms in the stemming process

Stem: a base part of a word that may have or may not have semantic meaning and no *affixes* (see below)
Affix: a small linguistic unit with semantic meaning that is attached to the beginning or the end of a stem to form a word
Prefix: an affix that is added to the beginning of a word or stem
Suffix: an affix that is added to the end of a word or stem
Stop Word: a term that appears so frequently in documents that it does not help searches [van Rijsbergen, 1979]. For example: *I, a, to, any, where, you*
Inflection: the modification or marking of a word to reflect semantic and grammatical information like gender, tense, number, case or person. For example: to help - > *helps, helping, helped* (reflects gender and tense)
Derivation : the modification of a word that transforms it from one syntactical category (verb, noun, adverb, adjective) to another. For example: to hope - > *hopefully , hopeless, hopes* (transformation to adverb, adjective and noun/verb)

Conflation class: a group of words that share the same semantic meaning [Paice, 1996]. For example: *group , grouping, teams*
Compounding: the creation of new words by combination of two or more different words into a single form. For example: *solar-powered, breastfeeding, bitter-sweet, antidisestablishmentarianism*
Morphological Variants: two or more words that are related due to inflation, derivation of compounding. For example: *dark, darkening, darks, darkroom*

1.3 Inflectional versus Derivational Variants

Inflectional and derivational are the two categories of morphological variants that stemmers mainly deal with. Any given word can have inflectional and derivational morphological variants. Inflectional morphological variants share the same basic meaning and belong to the same part of speech. For instance, the changes may affect the word's case, number, tense and gender. In contrast derivational morphological variants can belong to a different part of speech and thus, may mean something completely different from their stem since an adjective can be derived from a noun, or a noun from a verb among others. Many of the algorithms so far developed do not treat derivational suffixes or handle them partially. According to Paice [1994], affixes may contain important information about the meaning of the word and so it is advisable not to discard it during the stemming process. Stemming “antidote” to “dote “ creates a word that belongs to a different conflation class since the two words deal with different concepts.

Paice refers to the English language. However, the same phenomenon is noticed also in Greek. Affixes that are added usually alter the meaning of the word completely compared to the initial meaning of the stem. Some examples of this type of derivation are “μόρφωση” (education) to “παραμόρφωση” (disfigurement, deformation) and “δοκιμάζω” (I try) to “αποδοκιμάζω” (I boo, I abjure). A stemmer that includes derivational rules can be helpful for language research, but may be inappropriate as a query expansion tool, since it can supply the search engine with variants of the original words that have a very different meaning. This fact can increase the number of matched documents but the semantic accuracy to the original word will be low.

1.4 Stemming Techniques – Advantages and Disadvantages

In addition to whether or not a stemmer treats prefixes, they are also categorized as (i) dictionary based (ii) based on algorithms or (iii) a hybrid version of both [Ntais, 2006]. Dictionary based stemmers use ready made dictionaries and match a word with its stem from a list. The main drawback of such stemmers is that dictionary maintenance is required and that these stemmers can not scale to handle unlimited words.

In contrast, algorithm based stemmers have been the focus of research with the algorithms of Lovins and Porter being the most representative. The former [Lovins, 1968] precedes the latter by 12 years and it was the first stemming algorithm ever published. It uses an extensive list of 294 endings, 35 transformation rules and 29 conditions. The algorithm is executed in two basic steps. In the first step the longest ending is matched and removed and in the second step the algorithm checks whether one of the 35 transformation rules should be applied.

Although Lovins' algorithm was the first stemming algorithm, Porter's is considered one of the most influential [Krovetz, 1993; Ntais, 2006; Frakes, 2003]. It was initially written for the English language and later ported to other European languages like Italian, Spanish, French and Portuguese. In contrast with Lovins' algorithm, it iteratively applies a set of rules and removes suffixes until no rules apply. The execution is completed into five distinct steps [Porter, 1980] and is considered very aggressive [Krovetz, 1993; Xu and Croft 1998]. Porter's work became influential and many implementations were written and made available by others. Unfortunately, these implementations contained errors. In order to deal with this problem, Porter released a framework with which stemming algorithms can be implemented using a string handling programming language called Snowball.

1.5 Greek Stemmers

Three stemmers have been developed for Greek. The first two are the TZK algorithm by Kalamboukis and Nikolaidis in 1995 and the Automated Morphological Processor (AMP) by Tambouratzis and Carayanis in 2001. These two stemmers have an acceptable accuracy of 90 to 95% but for their development some constraints were implied. The

AMP algorithm assumes that each word just consists of a stem part and an ending part and thus excludes all compound words. One the other hand, the TZK algorithm can only manipulate 65 suffixes although there are more than 166 suffixes in the Greek language. Furthermore, to our knowledge, neither of these algorithms has a freely available implementation.

The latest stemming algorithm developed for the Greek language is by George Ntais, in 2006. This algorithm follows the structure of the Porter algorithm and has a free implementation available on the web [Ntais, 2008]. The author has provided a web interface where users can make simple queries by posting a single word in Greek and have the word's stem returned. The interface is simple and uses Javascript for the implementation of the algorithm. According to the author, the algorithm can handle 158 suffixes of the Greek language, clearly outperforming the TZK and AMP algorithms. Nevertheless, in order to avoid complexity and due to constraints imposed during its development, it can only work with words in capital letters. In the Greek language, lower case words have accent marks that can totally change the meaning of a word, like the adjectives “αβαθής” and “άβαθης”. Both of the words mean “shallow”, with the former being the masculine nominative case and the later the feminine genitive. In addition, the stemmer is able to handle suffixes but not prefixes. The essential information about all stemming algorithms that affected this thesis work and were presented in the previous two sections is listed in Table 3**.**

Table 3: Stemming Algorithms – Summarized Information

Name	Language	Year	Web Availability	Derivation Dealing	Number of Execution Steps	Weaknesses
Lovins'	English	1968	Yes	Yes	2	Aggressive with short stems and words
Porter's	English	1980	Yes	No	5	Quite aggressive and produces overstemming [Carlberger *et al.*, 2001]
TZK	Greek	1995	No	Yes	2	Does not handle all suffixes
AMP	Greek	2001	No	No	4	Unable to handle compound words
Ntais'	Greek	2006	Yes	No	29	Relatively new and untested; can only handle capital letters; produces understemming errors

1.6 Problems and Issues with the Latest Algorithm for Greek

According to its author [Ntais, 2006], the latest stemming algorithm for Greek suffers from a few limitations and constraints that had to be imposed during its development. One of them is the incapability to handle lower case letters. The algorithm is only able to handle words in upper case letters and will not stem any word that contains even one letter in lower case. Since words in upper case do not have tone marks in Greek, the author is solving the problem of the moving mark phenomenon that can be observed during the conjugation of verbs, nouns and adjectives since no tone mark has to be presented in the stem returned. Because of that, the algorithm has a new limitation since most of the words encountered in Greek texts are in lower case.

Another issue with the existing algorithm is incapability to process some important suffixes. The rationale was that the inclusion of these suffixes would introduce more errors if the appropriate exception list was also introduced. The creation of an extensive exception list was not feasible at that moment, so the algorithm is not treating suffixes like "-ατε" ," -αστε" and "-τε". These suffixes correspond to words from many syntactical categories like adverbs, nouns and verbs. Verbs with these endings correspond to past tenses. Past tenses are extensively used in Greek as well as in other Mediterranean languages, since similar observations have been mentioned in the use of Spanish, Portuguese and Italian. Cultural writings and conversational contexts are widely employing tenses like past continuous. It is therefore clear, that the exclusion of these suffixes makes the algorithm incomplete and inaccurate and limits its performance.

In addition to the incompletenesses of the algorithmic design, the implementation of the algorithm offers limited usability. Ntais [2008] has provided a web interface written in Javascript. Through a form, users can insert a Greek word in upper case and have its stem returned. Despite the fact that everyone can examine the algorithm, since it is embedded in the web page, it can not be directly used by any kind of application that requires stemming in Greek. Its implementation language, Javascript, is a powerful language for client side scripting in web applications. Nevertheless, it can not be used for writing a library that can be used by other applications.

1.7 Aims of this Thesis

We are conducting this work in order to fully test the existing algorithm and improve it. From our initial, undocumented tests, we concluded that the existing algorithm is giving satisfactory but inaccurate and incomplete results . We are convinced that documenting and analysing the results and improving the algorithm would be a contribution not only to computer science and computational linguistics in particular but also to all these fields that Greek is used including medicine and mathematics. In addition, the algorithm can later on be used at a production level in a search engine, with the potential to give better results and a better web searching experience to users searching for documents written in Greek.

In addition to improving the existing algorithm, we will also provide a library version of the algorithm written in PHP. The reason is that PHP is currently one of the most widely used languages in the web by providing server-side scripting for building dynamic web sites. By implementing a PHP algorithm, our aim is to provide a stemmer that can be directly used by the engine of any web application, for any kind of web search or linguistics. Other programming languages such as Javascript or even the more powerful like C and C++ lack this ability [Lerdorf and Tatroe, 2002; Flanagan, 2004]. In conclusion, our work, which will be available under an Open Source licence, will lead to a more powerful, more complete and more consistent Greek stemmer that can be directly examined, used and modified by others.

1.8 Research Question

In search for solutions to the previously stated problems, the research question to be answered in this thesis can be formulated as follows:

Up to which point the addition of more syntactical rules and exceptions improves the precision of the Ntais stemming algorithm?

The previous algorithm by Ntais [2006] does not, deliberately, include some suffixes in an attempt to avoid errors that occur when the appropriate exception list is not also introduced with the addition of a new suffix. The creation of an exception list is a trivial but rather time consuming process. We need to identify whether it is feasible or not to create extensive and complete lists of exceptions for new rules at this point, where the existing stemmer is already producing somewhat satisfactory results and covers most of the cases.

1.9 Methodology

One of the initial aims of this work was to test the original stemmer in combination with a search engine, and Google's search engine was a candidate. A web interface that would feed Google with modified, stemmed queries and unmodified ones could be easily built. The results of both modified and unmodified queries could then be compared. Unfortunately, not only the application of a stemmer in a web search engine is beyond the time limitations of this thesis, but it is also unclear whether Google is already utilizing any kind of stemming techniques for Greek. Furthermore, in a previous web search engine evaluation [Lazarinis, 2005], it is pointed out that Google returns a different number of results for different variations of the word “Athens” (Αθήνα: Athens, Αθήνας: of Athens, Αθηνών : of (the city of) Athens). The difference in results can only imply that no stemming is used. Despite that, there are reports that some form of stemming is being conducted [Google, 2003] although it is unclear how extensive. In addition Paice [1994] suggests that evaluating a stemmer solely in terms of IR is incomplete since IR is only one field that stemming can be applied and "*...gives no insight into the specific causes of errors*". Because of all these reasons and due to time limitations we decided not to test our implementations of both the existing algorithm and our improved version with a search engine.

During our research, we will modify Ntais' algorithm to use more grammatical rules, exceptions and stop words. We will improve the algorithm in a constructive and extended manner. More additions will be implemented incrementally, after testing all previous improvements each time. The task of introducing more grammatical rules is challenging since it utilises techniques and requires knowledge from two domains, computer science and linguistics. In order to evaluate Ntais' and our revised algorithm, we will execute both of them in batch mode against a collection of more than half a million Greek words. Both algorithms will stem the input words from the text, and will form groups of words that have the same stem. Our purpose is to manually check whether all the words reduced by the algorithm to the same stem also share the same semantic meaning. The two algorithms are evaluated separately and the results will be compared.

1.10 Overview of Thesis' Contents

In Chapter 2 we will provide a short overview of the history of the Greek language and some of the Greek grammar features and peculiarities that Greek words have during conjugation. Chapter 3 introduces some stemmer performance metrics that will be used during our evaluation in order to compare the output of the original stemmer and our modified version. The design of the existing algorithm and an extensive list of its rules are given in Chapter 4. This chapter also deals with some alternative approaches which we also considered for the re-design of our algorithm and the reasons for which we rejected them, based on their suitability to the application domain.

In Chapter 5, the evaluation of the existing algorithm is described and in Chapter 6 we describe the improvements that we decided to incorporate in our new algorithm along with the new set of rules and exception lists. The final evaluation of the algorithm and a comparison against its predecessor is given in Chapter 7.

2. The Greek Language

2.1 The History of the Greek Language

The earliest traces of written Greek can be found in more than 4400 clay tablets of the Linear B script, which was deciphered during 1951 to 1953 by the architect M. Ventris in England. This form of writing was used from 1600 to 1100 BC, and it is considered as the "earliest European script we can understand" [Robinson, 1995]. For the next 300 years, a period regarded as "the Dark Age" of illiteracy in Greece [Robinson, 1995], no traits of the Greek language have been discovered. During this period, the Homeric Greeks gave their position to the classical Greeks. The classic period of Ancient Greece (500-323 BC) coincides with the emergence of a new alphabet borrowed from the Phoenicians. Although it is debatable whether the Phoenicians or Greeks living in Phoenicia were the creators of this alphabet [Robinson, 1995], this alphabet is the ancestor of not only the Greek alphabet, but through the Etruscan and Latin languages, the ancestor of modern European alphabets.[Baugh & Cable, 2001]. The known fact is that the first consonant-only based Phoenician alphabet came to Greece without vowels, and the ancient Greeks added vowels to it. These added letter-characters improved greatly the communication and increased the use of the alphabet in everyday life and in writing form. After all, the enhanced with vowels new alphabet came nearer to the needs of everyday speech and it mirrored the spoken words more clearly and more accurately than its consonant based predecessor.

After many additions and simplifications through the years, the nowadays alphabet contains twenty four (24) letters. Table 4 presents the latest form of the Greek alphabet including both upper and lower case letters and their equivalent sound in the English alphabet. The third character of Sigma (ς) is only used in the end of a lower case word.

Table 4: The Greek Alphabet. Letters and their equivalent sound in English

Alpha	Beta	Gamma	Delta	Epsilon	Zeta	Eta	Theta	Iota	Kapa	Lamda	Mi
Α α	Β β	Γ γ	Δ δ	Ε ε	Ζ ζ	Η η	Θ θ	Ι ι	Κ κ	Λ λ	Μ μ
a	v	-	th (the)	e	z	i	th	i	k	l	m
Ni	**Xi**	**Omikron**	**Pi**	**Ro**	**Sigma**	**Tau**	**Ypsilon**	**Phi**	**Chi**	**Psi**	**Omega**
Ν ν	Ξ ξ	Ο ο	Π π	Ρ ρ	Σ σ ς	Τ τ	Υ υ	Φ φ	Χ χ	Ψ ψ	Ω ω
n	ks	o	p	r	s	t	i	f	h	ps	o

The Ancient Greeks of the Classical period were organized in city states. Each of the main city states had its own region of influence and with that a different dialect. The differences between these dialects were minor, so Greek was considered as a common language [Triantafyllidis, 1941]. It was only after the conquests of Alexander the Great in Asia when the Athenian dialect, after borrowing words from other dialects, became the common dialect spoken from Greece and Egypt to Syria and Persia. The language continued to evolve for the next centuries, until the fall of Constantinople in 1453 and the beginning of the Ottoman era. For the next 400 years, following the closing of schools, the Greek language is kept oral, divided into local dialects. A few examples of written material from this period can be credited to Greeks living in countries in Central and Western Europe, like Romania, Austria, Russia, the Hungarian Empire and Italy.

After the Greek War of Independence, which started in 1821, and the liberation in 1829, two competing varieties are found. The popular and spoken "Dimotiki", used among the people, and the official and most resembling to Ancient Greek "Katharevousa", used mostly by the intellectuals. Today, modern mainstream Greek is based on "Dimotiki" and is the official state language, since 1975, with simplifications in the intonation system since 1981. In places, there are still local dialects with varying degrees of differentiation from the mainstream language.

Greek is spoken by 14 to 17 million people, officially in Greece and Cyprus and unofficially in countries like Australia, USA and Canada, where there are Greek and Cypriot communities. In addition, medical and philosophical terms are often Greek, Greek-derived or combinations of Latin and Greek words [Kurz & Kilian, 2001]. The domain of Humanities is undeniably a language world with terms and concepts that have originally been founded in the subject of Philosophy and Mathematics and expressed in Greek.

2.2 Stemming in Greek

The Greek language is grammatically more complex than English. It has conjugations and morphologically complex words [Mackridge, 1987]. Articles, adjectives, nouns and even first names and surnames may be in various cases (like nominative and genitive), in singular or plural form and they are differentiated according to their gender (masculine, feminine, neuter). Table 5 contains the singular and plural numbers of all cases and genders that the word "cat" that be found in Greek.

Table 5: The noun "cat" in Greek

	Singular			**Plural**		
Cases	**masculine**	**feminine**	**neuter**	**masculine**	**feminine**	**neuter**
nominative	γάτ**ος**	γάτ**α**	γατ**ί**	γάτ**οι**	γάτες	γατ**ιά**
genitive	γάτ**ου**	γάτ**ας**	γατ**ιού**	γάτ**ων**	γάτ**ων**	γατ**ιών**
accusative	γάτ**ο**	γάτ**α**	γατ**ί**	γάτ**ους**	γάτ**ες**	γατ**ιά**
vocative	γάτ**ε**	γάτ**α**	γατ**ί**	γάτ**οι**	γάτ**ες**	γατ**ιά**

In addition to the complexity mentioned above, verbs are also heavily inflected. The

Greek language consists of present, past and future tenses with both perfective and imperfective aspects. These tenses have active and passive voices. Verbs are divided into two conjugations classes which have different endings and many times there are alternative endings for the same number and person. From the tables given in Appendix A, containing some examples about verb conjugation, it is obvious that Greek is much more complicated than English. Where in English four (4) endings are used, in Greek the distinct endings are 107, even without counting the different endings because of the moving mark phenomenon. Not only a stemmer has to deal with an enormously greater number of endings, but a somewhat perfect stemmer should be aware of how to deal with the “moving” tone mark issue.

3. Stemmer Performance Metrics

In order to evaluate the existing stemmer and measure its effectiveness, we will introduce some of the metrics that can be found in the previous literature.

3.1 Frakes' Metrics

Frakes [2003] defines *stemmer strength* as the degree to which a stemmer changes words. These changes fall into two categories, *removal* and *recording*. *Removal* is the decrease of a word's length due to elimination of an affix, whereas *recording* is the replacing of a word's letter with another. Since the strength of a stemmer can affect the precision and recall in queries, Frakes defines a set of metrics that help to compare algorithms by having the algorithms stem the same texts and compare the results of the following metrics:

- The Mean Number of Words per Conflation Class

The mean number of words per conflation class is the average number of words that are found in each conflation class. For example if the words "engineer," "engineered," and "engineering" are stemmed to "engineer," then this conflation class size is three. Stronger stemmers produce conflation classes with more words than lighter stemmers, from the same text.

- Index Compression Factor

The index compression factor is defined as ICS = $\frac{n-s}{n}$, where n is the number of words in the corpus and s is the number of stems produced by the stemmer. This metric indicates the index reduction that can be achieved through stemming. For example, if during the stemming of a corpus with 1000 words *(n)* we end up with 800 stems *(s)*, we have eliminated 200 words which means an index compression factor of 20%. Stronger stemmers will have a larger index compression factor than lighter stemmers.

- The Number of Words and Stems that Differ

Stemmers often leave words unchanged. The reason behind this behaviour can be either the lack of an appropriate rule, a software bug in the implementation of the algorithm or a design choice from the authors. For example, a stemmer might not alter "engineer" because it is already a dictionary entry. A big ratio of unchanged words to total words can indicate poor algorithmic performance. Stronger stemmers will change words more often than weaker stemmers.

- The median and mean modified Hamming distance

The Hamming distance between two strings of equal length is defined as the number of characters in the two strings that are different at the same position. For strings of unequal length we add the difference in length to the Hamming distance to give a modified Hamming distance function d. This measure takes into account transformations of stem endings. For example, a stemming algorithm might reduce the corpus { try, tried, trying } to the stem “tri”. The mean modified Hamming distance between the original words and the stem is D = (1+2+4)/ 3 = 2.33 characters, and the median is 2.

3.2 Error Metrics

There are two clearly distinct error metrics categories concerning stemmers, *understemming* and *overstemming*.

Understemming occurs when words are not fully stemmed to their potential stem. In that case, words that share the same conceptual meaning are stemmed to different stems and assigned to a different conflation class.

In contrast, overstemming occurs when words that do not share the same conceptual meaning are reduced into the same stem and assigned to the same conflation class. According to other evaluations [Alvares *et al*., 2005], the most accurate way to check for understemming and overstemming errors is through human interference. Some examples for both categories are given in Table 6. The first example shows understemming where two words have different stems although they should had the same, since they both have to do with “selling”. On the other hand, the second examples demonstrates overstemming where two words with different semantics, “selling “ and “bird”, are incorrectly reduced to the same stem.

Table 6: Stemming Errors

Understemming

Word	Meaning	Stem
πουλάω	(I) sell	πουλ-
πουλώντας	selling	πουλοντ-
Overstemming		
Word	**Meaning**	**Stem**
πουλάω	I sell	πουλ-
πουλί	bird	πουλ-

4. Algorithmic Design

4.1 The Existing Algorithm

The algorithm provided by Ntais deals "with each suffix individually" in a decentralized manner [Ntais, 2006]. The algorithm has 29 rules that treat 158 suffixes. Every rule is executed in an individual step and a set of suffixes is provided in order to remove the longest matching suffix. In all but the first steps, a list of exceptions is also examined and some different suffixes are added to the stem if needed in order to deal with the complexity of the Greek language. Additionally, each step may have its own exceptions. We have decided to keep this design and base our work on this.

Table 7 presents the algorithm by Ntais [2008]. In each step the rule with suffixes to be examined, along with actions to be taken and exceptions to be considered are described.

Table 7: Ntais' Algorithm

Step #	Rule	Action	Exceptions
1	Word ends in: ***ΦΑΓΙΑ|ΦΑΓΙΟΥ|ΦΑΓΙΩΝ|ΣΚΑΓΙΑ|ΣΚΑΓΙΟΥ|ΣΚΑΓΙΩΝ|ΟΛΟΓΙΟΥ|ΟΛΟΓΙΑ|ΟΛΟΓΙΩΝ|ΣΟΓΙΟΥ|ΣΟΓΙΑ|ΣΟΓΙΩΝ|ΤΑΤΟΓΙΑ|ΤΑΤΟΓΙΟΥ|ΤΑΤΟΓΙΩΝ|ΚΡΕΑΣ|ΚΡΕΑΤΟΣ|ΚΡΕΑΤΑ|ΚΡΕΑΤΩΝ|ΠΕΡΑΣ|ΠΕΡΑΤΟΣ|ΠΕΡΑΤΑ|ΠΕΡΑΤΩΝ|ΤΕΡΑΣ|ΤΕΡΑΤΟΣ|ΤΕΡΑΤΑ|ΤΕΡΑΤΩΝ|ΦΩΣ|ΦΩΤΟΣ|ΦΩΤΑ|ΦΩΤΩΝ|ΚΑΘΕΣΤΩΣ|ΚΑΘΕΣΤΩΤΟΣ|ΚΑΘΕΣΤΩΤΑ|ΚΑΘΕΣΤΩΤΩΝ|ΓΕΓΟΝΟΣ|ΓΕΓΟΝΟΤΟΣ|ΓΕΓΟΝΟΤΑ|ΓΕΓΟΝΟΤΩΝ***	Replace suffix with: ***ΦΑ|ΣΚΑ|ΟΛΟ|ΣΟ|ΤΑΤΟ|ΚΡΕ|ΠΕΡ|ΤΕΡ|ΦΩ|ΚΑΘΕΣΤ|ΓΕΓΟΝ***	

2a	Word ends in: ***ΑΔΕΣ|ΑΔΩΝ***	Remove suffix / Check exceptions	If after removal the word does not end in: ***ΟΚ|ΜΑΜ|ΜΑΝ|ΜΠΑΜΠ|ΠΑΤΕΡ|ΓΙΑΓΙ|ΝΤΑΝΤ|ΚΥΡ|ΘΕΙ|ΠΕΘΕΡ*** Add “***ΑΔ***” in the end
2b	Word ends in: ***ΕΔΕΣ|ΕΔΩΝ***	Remove suffix / Check exceptions	If after removal the word ends in: ΟΠ|ΙΠ|ΕΜΠ|ΥΠ|ΓΗΠ|ΔΑΠ|ΚΡΑΣΠ|ΜΙΛ Add “***ΕΔ***” in the end
2c	Word ends in: ***ΟΥΔΕΣ|ΟΥΔΩΝ***	Remove suffix / Check exceptions	If after removal the word ends in: ***ΑΡΚ|ΚΑΛΙΑΚ|ΠΕΤΑΛ|ΛΙΧ,|ΠΛΕΧ|ΣΚ|Σ|ΦΛ|ΦΡ|ΒΕΛ|ΛΟΥΛ|ΧΝ|ΣΠ|ΤΡΑ|ΦΕ*** Add “***ΟΥΔ***” in the end
2d	Word ends in: ***ΕΩΣ|ΕΩΝ***	Remove suffix / Check exceptions	If after removal the word is one of: ***ΑΡΚ|ΚΑΛΙΑΚ|ΠΕΤΑΛ|ΛΙΧ|ΠΛΕΞ|ΣΚ|Σ|ΦΛ|ΦΡ|ΒΕΛ|ΛΟΥΛ|ΧΝ|ΣΠ|ΤΡΑΓ|ΦΕ*** Add “***Ε***” in the end
3	Word ends in: ***ΙΑ|ΙΟΥ|ΙΩΝ***	Remove suffix / Check exceptions	If after removal the word ends in Vowel Add “***Ι***” in the end
4	Word ends in: ***ΙΚΑ|ΙΚΟ|ΙΚΟΥ|ΙΚΩΝ***	Remove suffix / Check exceptions	If after removal the word ends in Vowel OR is one of : ***ΑΛ|ΑΔ|ΕΝΔ|ΑΜΑΝ|ΑΜΜΟΧΑΛ|ΗΘ|ΑΝΗΘ|ΑΝΤΙΔ|ΦΥΣ|ΒΡΩΜ|ΓΕΡ|ΕΞΩΔ|ΚΑΛΠ|ΚΑΛΛΙΝ|ΚΑΤΑΔ|ΜΟΥΛ|ΜΠΑΝ|ΜΠΑΓΙΑΤ|ΜΠΟΛ|ΜΠΟΣ|ΝΙΤ|ΞΙΚ|ΣΥΝΟΜΗΛ|ΠΕΤΣ|ΠΙΤΣ|ΠΙΚΑΝΤ|ΠΛΙΑΤΣ|ΠΟΣΤΕΛΝ|ΠΡΩΤΟΔ|ΣΕΡΤ|ΣΥΝΑΔ|ΤΣΑΜ|ΥΠΟΔ|ΦΙΛΟΝ|ΦΥΛΟΔ|ΧΑΣ*** Add “***ΙΚ***” in the end

5a	Word is ***ΑΓΑΜΕ***	Stem is ***ΑΓΑΜ***	
5a	Word ends in : ***ΑΓΑΜΕ|ΗΣΑΜΕ|ΟΥΣΑΜΕ|Η*ΚΑΜΕ|ΗΘΗΚΑΜΕ***	Remove suffix / Check exceptions	
5a	Word ends in: ***ΑΜΕ***	Remove suffix / Check exceptions	If after removal the word is one of: ***ΑΝΑΠ|ΑΠΟΘ|ΑΠΟΚ|ΑΠΟΣΤ|ΒΟΥΒ|ΞΕΘ|ΟΥΛ|ΠΕΘ|ΠΙΚΡ|ΠΟΤ|ΣΙΧ|Χ*** Add ***"ΑΜ"*** in the end
5b	Word ends in: ***ΑΓΑΝΕ|ΗΣΑΝΕ|ΟΥΣΑΝΕ|ΙΟΝΤΑΝΕ|ΙΟΤΑΝΕ|ΙΟΥΝΤΑΝΕ|ΟΝΤΑΝΕ|ΟΤΑΝΕ|ΟΥΝΤΑΝΕ|ΗΚΑΝΕ|ΗΘΗΚΑΝΕ***	Remove suffix / Check exceptions	If after removal the word one of: ***ΤΡ|ΤΣ*** Add "***ΑΓΑΝ"***

5b	Word ends in: ***ΑΓΑΝΕ|ΗΣΑΝΕ| ΟΥΣΑΝΕ|ΙΟΝΤΑΝΕ| ΙΟΤΑΝΕ|ΙΟΥΝΤΑΝΕ| ΟΝΤΑΝΕ|ΟΤΑΝΕ| ΟΥΝΤΑΝΕ|ΗΚΑΝΕ| ΗΘΗΚΑΝΕ***	Remove suffix / Check exceptions	◻ If after removal the word ends in Vowel without "***Υ***" OR is one of : ***ΒΕΤΕΡ|ΒΟΥΛΚ|ΒΡΑΧΜ|Γ| ΔΡΑΔΟΥΜ|Θ|ΚΑΛΠΟΥΖ| ΚΑΣΤΕΛ|ΚΟΡΜΟΡ|ΛΑΟΠΛ| ΜΩΑΜΕΘ|Μ|ΜΟΥΣΟΥΛΜ|Ν| ΟΥΛ|Π|ΠΕΛΕΚ|ΠΛ|ΠΟΛΙΣ| ΠΟΡΤΟΛ|ΣΑΡΑΚΑΤΣ|ΣΟΥΛΤ| ΤΣΑΡΛΑΤ|ΟΡΦ|ΤΣΙΓΓ|ΤΣΟΠ| ΦΩΤΟΣΤΕΦ|Χ|ΨΥΧΟΠΛ|ΑΓ| ΟΡΦ|ΓΑΛ|ΓΕΡ|ΔΕΚ|ΔΙΠΛ| ΑΜΕΡΙΚΑΝ|ΟΥΡ|ΠΙΘ|ΠΟΥΡΙΤ| Σ|ΖΩΝΤ|ΙΚ|ΚΑΣΤ|ΚΟΠ|ΛΙΧ| ΛΟΥΘΗΡ|ΜΑΙΝΤ|ΜΕΛ|ΣΙΓ|ΣΠ| ΣΤΕΓ|ΤΡΑΓ|ΤΣΑΓ|Φ|ΕΡ|ΑΔΑΠ| ΑΘΙΓΓ|ΑΜΗΧ|ΑΝΙΚ|ΑΝΟΡΓ| ΑΠΗΓ|ΑΠΙΘ|ΑΤΣΙΓΓ|ΒΑΣ| ΒΑΣΚ|ΒΑΘΥΓΑΛ|ΒΙΟΜΗΧ| ΒΡΑΧΥΚ|ΔΙΑΤ|ΔΙΑΦ|ΕΝΟΡΓ| ΘΥΣ|ΚΑΠΝΟΒΙΟΜΗΧ| ΚΑΤΑΓΑΛ|ΚΛΙΒ|ΚΟΙΛΑΡΦ|ΛΙΒ| ΜΕΓΛΟΒΙΟΜΗΧ| ΜΙΚΡΟΒΙΟΜΗΧ|ΝΤΑΒ| ΞΗΡΟΚΛΙΒ|ΟΛΙΓΟΔΑΜ| ΟΛΟΓΑΛ|ΠΕΝΤΑΡΦ|ΠΕΡΗΦ| ΠΕΡΙΤΡ|ΠΛΑΤ|ΠΟΛΥΔΑΠ| ΠΟΛΥΜΗΧ|ΣΤΕΦ|ΤΑΒ|ΤΕΤ| ΥΠΕΡΗΦ|ΥΠΟΚΟΠ| ΧΑΜΗΛΟΔΑΠ|ΨΗΛΟΤΑΒ*** Add "***ΑΝ***" in the end
5c	Word ends in: ***ΗΣΕΤΕ***	Remove suffix / Check exceptions	

5c	Word ends in: ***ΕΤΕ***	Remove suffix / Check exceptions	☐ If after removal the word ends in Vowel without “***Υ***” OR is one of : ☐ ***ΑΒΑΡ|ΒΕΝ|ΕΝΑΡ|ΑΒΡ|ΑΔ|ΑΘ|ΑΝ|ΑΠΛ|ΒΑΡΟΝ|ΝΤΡ|ΣΚ|ΚΟΠ|ΜΠΟΡ|ΝΙΦ|ΠΑΓ|ΠΑΡΑΚΑΛ|ΣΕΡΠ|ΣΚΕΛ|ΣΥΡΦ|ΤΟΚ|Υ|Δ|ΕΜ|ΘΑΡΡ|Θ*** OR ends in : ***ΟΔ|ΑΙΡ|ΦΟΡ|ΤΑΘ|ΔΙΑΘ|ΣΧ|ΕΝΔ|ΕΥΡ|ΤΙΘ|ΥΠΕΡΘ|ΡΑΘ|ΕΝΘ|ΡΟΘ|ΣΘ|ΠΥΡ|ΑΙΝ|ΣΥΝΔ|ΣΥΝ|ΣΥΝΘ|ΧΩΡ|ΠΟΝ|ΒΡ|ΚΑΘ|ΕΥΘ|ΕΚΘ|ΝΕΤ|ΡΟΝ|ΑΡΚ|ΒΑΡ|ΒΟΛ|ΩΦΕΛ*** Add “***ΕΤ***” in the end
5d	Word ends in: ***ΟΝΤΑΣ|ΩΝΤΑΣ***	Remove suffix / Check exceptions	If after removal the word is: ***ΑΡΧ*** add “***ΟΝΤ***” in the end OR If after removal the word is: ***ΚΡΕ*** add “**Ω*ΝΤ***” in the end
5e	Word ends in: ***ΟΜΑΣΤΕ|ΙΟΜΑΣΤΕ***	Remove suffix / Check exceptions	If after removal the word is: ***ΟΝ*** add “***ΟΜΑΣΤ***” in the end
5f	Word ends in: ***ΙΕΣΤΕ***	Remove suffix / Check exceptions	If after removal the word is one of: ***Π|ΑΠ|ΣΥΜΠ|ΑΣΥΜΠ|ΑΚΑΤΑΠ|ΑΜΕΤΑΜΦ*** *Add **“ΙΕΣΤ”** in the end*
5f	Word ends in: ***ΕΣΤΕ***	Remove suffix / Check exceptions	If after removal the word is one of: ***ΑΛ|ΑΡ|ΕΚΤΕΛ|Ζ|Μ|Ξ|ΠΑΡΑΚΑΛ|ΑΡ|ΠΡΟ|ΝΙΣ*** *Add **“ΕΣΤ”** in the end*
5g	Word ends in: **ΗΘΗΚΑ|ΗΘΗΚΕΣ|ΗΘΗΚΕ**	Remove suffix / Check exceptions	

5g	Word ends in: **ΗΚΑ|ΗΚΕΣ|ΗΚΕ**	Remove suffix / Check exceptions ☐	☐ If after removal the word is one of: ***ΔΙΑΘ|Θ|ΠΑΡΑΚΑΤΑΘ|ΠΡΟΣΘ|ΣΥΝΘ*** OR ends in: ***ΣΚΩΛ|ΣΚΟΥΛ|ΝΑΡΘ|ΣΦ|ΟΘ|ΠΙΘ*** ☐ Add "***ΗΚ***" in the end
5h	Word ends in: **ΟΥΣΑ|ΟΥΣΕΣ|ΟΥΣΕ**	Remove suffix / Check exceptions ☐	☐ If after removal the word is one of: ***ΦΑΡΜΑΚ|ΧΑΔ|ΑΓΚ|ΑΝΑΡΡ|ΒΡΟΜ|ΕΚΛΙΠ|ΛΑΜΠΙΔ|ΛΕΧ|Μ|ΠΑΤ|Ρ|Λ|ΜΕΔ|ΜΕΣΑΖ|ΥΠΟΤΕΙΝ|ΑΜ|ΑΙΘ|ΑΝΗΚ|ΔΕΣΠΟΖ|ΕΝΔΙΑΦΕΡ|ΔΕ|ΔΕΥΤΕΡΕΥ|ΚΑΘΑΡΕΥ|ΠΛΕ|ΤΣΑ*** OR ends in: ***ΠΟΔΑΡ|ΒΛΕΠ|ΠΑΝΤΑΧ|ΦΡΥΔ|ΜΑΝΤΙΛ|ΜΑΛΛ|ΚΥΜΑΤ|ΛΑΧ|ΛΗΓ|ΦΑΓ|ΟΜ|ΠΡΩΤ*** ☐ Add "***ΟΥΣ***" in the end

5i	Word ends in: **ΑΓΑ|ΑΓΕΣ|ΑΓΕ**	Remove suffix / Check exceptions □	□ If after removal the word is one of: ***ΑΒΑΣΤ|ΠΟΛΥΦ|ΑΔΗΦ|ΠΑΜΦ|Ρ|ΑΣΠ|ΑΦ|ΑΜΑΛ|ΑΜΑΛΛΙ|ΑΝΥΣΤ|ΑΠΕΡ|ΑΣΠΑΡ|ΑΧΑΡ|ΔΕΡΒΕΝ|ΔΡΟΣΟΠ|ΞΕΦ|ΝΕΟΠ|ΝΟΜΟΤ|ΟΛΟΠ|ΟΜΟΤ|ΠΡΟΣΤ|ΠΡΟΣΩΠΟΠ|ΣΥΜΠ|ΣΥΝΤ|Τ|ΥΠΟΤ|ΧΑΡ|ΑΕΙΠ|ΑΙΜΟΣΤ|ΑΝΥΠ|ΑΠΟΤ|ΑΡΤΙΠ|ΔΙΑΤ|ΕΝ|ΕΠΙΤ|ΚΡΟΚΑΛΟΠ|ΣΙΔΗΡΟΠ|Λ|ΝΑΥ|ΟΥΛΑΜ|ΟΥΡ|Π|ΤΡ|Μ*** OR ends in: ***ΟΦ|ΠΕΛ|ΧΟΡΤ|ΛΛ|ΣΦ|ΡΠ|ΦΡ|ΠΡ|ΛΟΧ|ΣΜΗΝ*** BUT is not one of: ***ΨΟΦ|ΝΑΥΛΟΧ*** AND does not end in: ***ΚΟΛΛ*** Add "***ΑΓ***" in the end
5j	Word ends in: ***ΗΣΕ|ΗΣΟΥ|ΗΣΑ***	Remove suffix □ □ □ □	If after removal the word is one of: ***Ν|ΧΕΡΣΟΝ|ΔΩΔΕΚΑΝ|ΕΡΗΜΟΝ|ΜΕΓΑΛΟΝ|ΕΠΤΑΝ*** Add "***ΗΣ***" in the end
5k	Word ends in: ***ΗΣΤΕ***	Remove suffix / Check exceptions □ □ □ □	If after removal the word is one of: ***ΑΣΒ|ΣΒ|ΑΧΡ|ΧΡ|ΑΠΛ|ΑΕΙΜΝ|ΔΥΣΧΡ|ΕΥΧΡ|ΚΟΙΝΟΧΡ|ΠΑΛΙΜΨ*** Add "***ΗΣΤ***" in the end
5l	Word ends in: ***ΟΥΝΕ|ΗΣΟΥΝΕ|ΗΘΟΥΝΕ***	Remove suffix / Check exceptions □ □ □ □	If after removal the word is one of: ***Ν|Ρ|ΣΠΙ|ΣΤΡΑΒΟΜΟΥΤΣ|ΚΑΚΟΜΟΥΤΣ|ΕΞΩΝ*** Add "***ΟΥΝ***" in the end

51	Word ends in: ***ΟΥΜΕ|ΗΣΟΥΜΕ|ΗΘΟΥΜΕ***	Remove suffix / Check exceptions □ □ □ □	If after removal the word is one of: ***ΠΑΡΑΣΟΥΣ|Φ|Χ|ΩΡΙΟΠΛ|ΑΖ|ΑΛΛΟΣΟΥΣ|ΑΣΟΥΣ*** Add "***ΟΥΜ***" in the end
6	Word ends in: ***ΜΑΤΑ|ΜΑΤΩΝ|ΜΑΤΟΣ***	Remove suffix / Check exceptions □ □	Always Add "***ΜΑ***" in the end
6	Word ends in: ***Α|ΑΓΑΤΕ|ΑΓΑΝ|ΑΕΙ|ΑΜΑΙ|ΑΝ|ΑΣ|ΑΣΑΙ|ΑΤΑΙ|ΑΩ|Ε|ΕΙ|ΕΙΣ|ΕΙΤΕ|ΕΣΑΙ|ΕΣ|ΕΤΑΙ|Ι|ΙΕΜΑΙ|ΙΕΜΑΣΤΕ|ΙΕΤΑΙ|ΙΕΣΑΙ|ΙΕΣΑΣΤΕ|ΙΟΜΑΣΤΑΝ|ΙΟΜΟΥΝ|ΙΟΜΟΥΝΑ|ΙΟΝΤΑΝ|ΙΟΝΤΟΥΣΑΝ|ΙΟΣΑΣΤΑΝ|ΙΟΣΑΣΤΕ|ΙΟΣΟΥΝ|ΙΟΣΟΥΝΑ|ΙΟΤΑΝ|ΙΟΥΜΑ|ΙΟΥΜΑΣΤΕ|ΙΟΥΝΤΑΙ|ΙΟΥΝΤΑΝ|Η|ΗΔΕΣ|ΗΔΩΝ|ΗΘΕΙ|ΗΘΕΙΣ|ΗΘΕΙΤΕ|ΗΘΗΚΑΤΕ|ΗΘΗΚΑΝ|ΗΘΟΥΝ|ΗΘΩ|ΗΚΑΤΕ|ΗΚΑΝ|ΗΣ|ΗΣΑΝ|ΗΣΑΤΕ|ΗΣΕΙ|ΗΣΕΣ|ΗΣΟΥΝ|ΗΣΩ|Ο|ΟΙ|ΟΜΑΙ|ΟΜΑΣΤΑΝ|ΟΜΟΥΝ|ΟΜΟΥΝΑ|ΟΝΤΑΙ|ΟΝΤΑΝ|ΟΝΤΟΥΣΑΝ|ΟΣ|ΟΣΑΣΤΑΝ|ΟΣΑΣΤΕ|ΟΣΟΥΝ|ΟΣΟΥΝΑ|ΟΤΑΝ|ΟΥ|ΟΥΜΑΙ|ΟΥΜΑΣΤΕ|ΟΥΝ|ΟΥΝΤΑΙ|ΟΥΝΤΑΝ|ΟΥΣ|ΟΥΣΑΝ|ΟΥΣΑΤΕ|Υ|ΥΣ|Ω|ΩΝ***		Remove suffix
7	Word ends in: ***ΕΣΤΕΡ|ΕΣΤΑΤ|ΟΤΕΡ|ΟΤΑΤ|ΥΤΕΡ|ΥΤΑΤ|ΩΤΕΡ|ΩΤΑΤ***	Remove suffix □	

4.2 Rejected Designs

During the evaluation of the existing work we rejected some alternative to Ntais' algorithm designs for our stemmer.

4.2.1 Lovins' Design

As mentioned in Section 1.4, Lovins' algorithm uses two steps in order to remove suffixes from words. The algorithm is used for stemming in English. As a design, it offers more simplicity than the algorithm provided by Ntais, which is executed in 29 steps. During our evaluation, we implemented a design similar to Lovins' algorithm using the same list of suffixes that Ntais used. The reason was that we observed Ntais algorithm's behaviour and it removes suffixes in one or two steps on average, despite the fact that it always executes on 29 steps. Unfortunately, our implementation of a Lovins-like algorithm didn't offer any improvement. Ntais' algorithm uses an exception list after each step in order to deal with the richness and irregularities of Greek. Even if 29 steps may be regarded as poor design, since the rest of the algorithms mentioned in Table 3 execute in two to five steps, tracking and matching endings with exceptions is easily conducted. Thus, the algorithm can be studied and improved easily. Despite of the fact that eventually in our algorithm we kept Ntais' design and we even added more steps, we made sure that the algorithm is not making unnecessary checks by returning the correct stem right after matching all possible suffixes.

4.2.2 Context-Free Grammar Design

Another alternative design that we examined was a more theoretical and sophisticated one based on the theory of context-free grammars. The main idea behind the theory is that all natural languages are based on elements, which in turn can be based into other elements recursively [Lewis and Papadimitriou, 1998]. Parse trees can be used and by applying rewrite grammatical rules the elements of sentences can be constructed and, in the case of a stemmer, de-constructed and analysed.

This approach is similar to corpus-based stemming, where each word is examined and assigned to a conflation class not only according to the grammatical meaning it holds but also to its semantic meaning it contains in the text [Xu and Croft, 2000]. Our hypothesis was that an algorithm based in these principles could be trained to recognize words and build more complex conflation classes. In that way, when a word was given, the stemmer could return a list of alternative words that have already been extracted from texts and have the same meaning. Nevertheless, this approach is beyond stemming even though it could, theoretically, offer better results in search queries. Another prohibitive reason was that although there is literature available about this subject, there is no evidence, to our knowledge, of any production or working systems and stemmer implementation using this approach, at least publicly available.

4.2.3 Dictionary Based Design

A rule based stemming algorithm has to be aware of exceptions. By adding more rules to an algorithm, a researcher must also add more exceptions, in order to deal with the complexity of a natural language. One may argue that a stemmer with many exceptions resembles a dictionary stemmer, since it has embedded in its design lists of words. A dictionary stemmer could be built by using a rule-based stemmer for creating an initial list of stems, and then manually checking the stems created. We rejected that design, since our personal interest was not only to create a solution for Greek stemming but also to study the behaviour of the language.

4.2.4 Krovetz's Experimental Algorithm

Another possible use of a dictionary for our algorithmic design could be using a dictionary to check whether a rule produced a word that is a dictionary entry. Krovetz [1993] experimented with this design. In his algorithm, prior to the execution of each rule, the word is looked up in a dictionary. If the word is an entry in the dictionary, the algorithm returns it. In the opposite case, the algorithm proceeds in evaluating more rules. When the word is modified by a rule, the resulting word is again looked up in the dictionary. This pattern continues until the remaining word is an entry or no more rules can be applied.

Krovetz experimented with this design, in an attempt to deal with the aggressiveness of Porter's algorithm and ended up with a "weaker "stemmer. For example, this experimental algorithm would stem "generalisations" to "generalisation" and not to "general". This would provide optimal results in IR since "general" is a word with multiple entries in a dictionary. One of these entries has the same semantic meaning with "generalisations" while the other deals with military ranks. A search engine using Porter's algorithm would probably retrieve documents of both categories while trying to serve a query with the term "generalisation".

We decided not to follow this design for three reasons. Our main objection in following this design is the fact that we are conducting this work having both linguistics and IR in mind. A stemming algorithm with similar design could possibly produce better results in IR, but it would be an incomplete algorithm. In addition, this design would require many modifications in the suffix and exception lists. Krovetz's design is manipulating suffixes in a piece by piece fashion. In contrast, Ntais' algorithm is removing suffixes in one or two steps. By following Krovetz's approach, the suffix and exception list would have to be created from the beginning, a task that would require an enormous amount of effort. Finally, Krovetz reported [1993] that in many cases this design offered poor results compared to the original algorithm by Porter, even in IR uses.

4.3 Evaluation of Ntais' Algorithm

In order to evaluate the algorithm developed by George Ntais, we first ported the algorithm in PHP. Furthermore, we implemented a set of helper applications that will be using directly the algorithm and will keep statistics about the stems returned. The operating system used for the evaluation was Gentoo Linux but because of the portability of PHP and our style of coding the source code is portable and can be used in any platform that PHP is ported to.

Our evaluation begun by executing our port of Ntais' stemmer against a list of Greek words. We set the application in a manner that words with co mmon stems will be grouped into conflation classes and then the output will be directed into a text file. This text file was examined manually for understemming and overstemming errors. In addition to that, we used modified Hamming Distances in order to find similar stems. From our observations, we concluded that two stems with a modified Hamming Distance of four or less can be possibly merged into one, indicating an understemming or overstemming error of the stemmer that generated them. An example is the comparison of the "ΒΑΔΙΖ" and "ΒΑΔΙΖΑΤ"stems, erroneously generated by the original stemmer instead of one. The first is a stem for 23 words that have to do with the verb "to walk" while the latter is the stem generated for the word "ΒΑΔΙΖΑΤΕ" which is the third person plural in the Past Continuous tense of the same verb. The modified Hamming Distance between them is 2. We implemented a small application that outputs pairs of stems generated by the stemmer with a modified Hamming Distance between them of 3 or less. The pairs generated were potential understemming or overstemming errors and candidates for merge into one conflation class. After thorough examination of the lists generated we begun with the modification and the improvement of the algorithm.

5. Our Improvements

5.1 Introduction of Stop-Word Elimination

Stop-word removal is one of the most commonly used techniques in IR [Baeza-Yates and Ribeiro-Neto, 1999; Risvik *et al*., 2003; Lazarinis, 2007]. We use stop-word elimination in order to improve the performance of the stemming algorithm. The stop-word list mainly contains words of length of at most four letters. These words are mainly articles, adverbs and conjunctions that can not be conjugated or stemmed. In addition, some common words that can be found in Greek texts, like initials, are also added to this list. In our modified algorithm, stop-word elimination is the first step of execution, a step that not only produces better results but also improves the running time of the algorithm. These words tend to deceive the stemmer and as a result, non existing stems of minimal length are created. The initial algorithm by Ntais solved this problem by processing only words of four letters or more. Although this approach left just a few words of 3 that could be stemmed unprocessed, we decided to add a stop-word list of more than 500 words, in order to increase precision.

5.2 Addition of More Grammatical Rules

In the course of our evaluation, and in order to conduct our algorithm testing and comparison, we created a set of helper applications that directly use our implementations of both Ntais' and our modified algorithm. One of these applications uses as input a list of words and creates conflation classes according to the stems returned by the stemmers. These classes were manually checked for overstemming and understemming errors in a manner similar to previous literature [Alvares *et al*., 2005; Ntais, 2006]. According to the results, more suffixes were added in order to deal with understemming. As Ntais [2006] had pointed out, the introduction of more rules for additional suffixes raises stemming errors due to overstemming. In order to deal with this situation, we additionally added more exceptions in order to deal with overstemming and keep precision at an acceptable level. Our efforts concentrated mainly on the the addition of rules that deal with past tenses as they play a great role and can be often found in Greek texts.

5.3 Introduction of Lower Case Letters

The initial stemming algorithm of Ntais only accepts as input words in upper case letters, as we mentioned in Section 1.6. Our algorithm is capable of handling words given in any case, upper, lower or combinations of both. The main body of the algorithm remains unchanged and all rules are still in capital letters. Before the evaluation of the rules, each letter in the given word is converted into upper case. The algorithm stores the case of each letter individually in a different variable. The algorithm examines all rules and creates the stem in upper case. Before returning the stem of the given word, a final alteration of the stem occurs as the algorithm consults the case of each letter on the stem, and alters the case of a letter if needed.

The initial algorithm was only accepting words in upper case in order to deal with the "moving" tone mark. Although in our implementation the problem of the "moving" tone-mark still remains, we decided to treat both upper case and lower case words. The complexity of the Greek language and the time limitations of this thesis prohibit any serious attempt to solve this problem. Moreover, in a hypothetical situation in which this problem was solved, the improvement in precision would be minimal. Finally we believe that since a stem is not a real word, but just a linguistic unit, returning stems in Greek without tone marks in not an important issue.

5.4 The Revised Algorithm

After careful examination of the output of the original stemmer, we tried to incorporate as many modifications as possible. The original algorithm has some inaccuracies but, searching for omissions and errors can be compared with searching for a needle in a haystack. Moreover, an addition of a rule that corrects some errors may create other errors, unless an appropriate exception list is also created. Nevertheless, we added more rules in order to correct wrong patterns that kept appearing in the output. One striking example was the omission of any rules for suffixes that appear in Past Continuous (***IZA, IZEΣ, IZE, IZAME, IZATE IZAN***) and past tenses in general. This detailed work appears in Table 8.

Table 8: Our revised algorithm (additions and modifications highlighted)

Step #	Rule	Action	Exceptions
UL1	Word contains letters in lower case	Convert letters in upper case Store their position in the word	
SWR	Word is one of the Stop Word List (Apendix C)	Return word unchanged	
1	Word ends in: ***ΦΑΓΙΑ|ΦΑΓΙΟΥ|ΦΑΓΙΩΝ|ΣΚΑΓΙΑ|ΣΚΑΓΙΟΥ|ΣΚΑΓΙΩΝ|ΟΛΟΓΙΟΥ|ΟΛΟΓΙΑ|ΟΛΟΓΙΩΝ|ΣΟΓΙΟΥ|ΣΟΓΙΑ|ΣΟΓΙΩΝ|ΤΑΤΟΓΙΑ|ΤΑΤΟΓΙΟΥ|ΤΑΤΟΓΙΩΝ|ΚΡΕΑΣ|ΚΡΕΑΤΟΣ|ΚΡΕΑΤΑ|ΚΡΕΑΤΩΝ|ΠΕΡΑΣ|ΠΕΡΑΤΟΣ|ΠΕΡΑΤΗ|ΠΕΡΑΤΑ|ΠΕΡΑΤΩΝ|ΤΕΡΑΣ|ΤΕΡΑΤΟΣ|ΤΕΡΑΤΑ|ΤΕΡΑΤΩΝ|ΦΩΣ|ΦΩΤΟΣ|ΦΩΤΑ|ΦΩΤΩΝ|ΚΑΘΕΣΤΩΣ|ΚΑΘΕΣΤΩΤΟΣ|ΚΑΘΕΣΤΩΤΑ|ΚΑΘΕΣΤΩΤΩΝ|ΓΕΓΟΝΟΣ|ΓΕΓΟΝΟΤΟΣ|ΓΕΓΟΝΟΤΑ|ΓΕΓΟΝΟΤΩΝ***	Replace suffix with: ***ΦΑ|ΣΚΑ|ΟΛΟ|ΣΟ|ΤΑΤΟ|ΚΡΕ|ΠΕΡ|ΤΕΡ|ΦΩ|ΚΑΘΕΣΤ|ΓΕΓΟΝ***	

S1	Word ends in: ***ΙΖΑ|ΙΖΕΣ|ΙΖΕ|ΙΖΑΜΕ|ΙΖΑΤΕ|ΙΖΑΝ|ΙΖΑΝΕ|ΙΖΩ|ΙΖΕΙΣ|ΙΖΕΙ|ΙΖΟΥΜΕ|ΙΖΕΤΕ|ΙΖΟΥΝ|ΙΖΟΥΝΕ***	Remove suffix / Check exceptions / Exit	If after removal the word ends in: ***ΑΝΑΜΠΑ|ΕΜΠΑ|ΕΠΑ|ΞΑΝΑΠΑ|ΠΑ|ΠΕΡΙΠΑ|ΑΘΡΟ|ΣΥΝΑΘΡΟ|ΔΑΝΕ*** Add "***Ι***" in the end If after removal the word ends in: ***ΜΑΡΚ|ΚΟΡΝ|ΑΜΠΑΡ|ΑΡΡ|ΒΑΘΥΡΙ|ΒΑΡΚ|Β|ΒΟΛΒΟΡ|ΓΚΡ|ΓΛΥΚΟΡ|ΓΛΥΚΥΡ|ΙΜΠ|Λ|ΛΟΥ|ΜΑΡ|Μ|ΠΡ|ΜΠΡ|ΠΟΛΥΡ|Π|Ρ|ΠΙΠΕΡΟΡ*** Add "***ΙΖ***" in the end
S2	Word ends in: ***ΩΘΗΚΑ|ΩΘΗΚΕΣ|ΩΘΗΚΕ|ΩΘΗΚΑΜΕ|ΩΘΗΚΑΤΕ|ΩΘΗΚΑΝ|ΩΘΗΚΑΝΕ***	Remove suffix / Check exceptions / Exit	If after removal the word is: ***ΑΛ|ΒΙ|ΕΝ|ΥΨ|ΛΙ|ΖΩ|Σ|Χ*** Add "***ΩΝ***" in the end
S3	Word ends in: ***ΙΣΑ|ΙΣΕΣ|ΙΣΕ|ΙΣΑΜΕ|ΙΣΑΤΕ|ΙΣΑΝ|ΙΣΑΝΕ***	Remove suffix / Check exceptions / Exit	If word is: ***ΙΣΑ*** Stem is ***"ΙΣ"*** If after removal the word is: ***ΑΝΑΜΠΑ|ΑΘΡΟ|ΕΜΠΑ|ΕΣΕ|ΕΣΩΚΛΕ|ΕΠΑ|ΞΑΝΑΠΑ|ΕΠΕ|ΠΕΡΙΠΑ|ΑΘΡΟ|ΣΥΝΑΘΡΟ|ΔΑΝΕ|ΚΛΕ|ΧΑΡΤΟΠΑ|ΕΞΑΡΧΑ|ΜΕΤΕΠΕ|ΑΠΟΚΛΕ|ΑΠΕΚΛΕ|ΕΚΛΕ|ΠΕ|ΠΕΡΙΠΑ*** Add *"**Ι**"*in the end If after removal the word is: ***ΑΝ|ΑΦ|ΓΕ|ΓΙΓΑΝΤΟΑΦ|ΓΚΕ|ΔΗΜΟΚΡΑΤ|ΚΟΜ|ΓΚ|Μ|Π|ΠΟΥΚΑΜ|ΟΛΟ|ΛΑΡ*** Add *"**ΙΣ**"*in the end

S4	Word ends in: ***ΙΣΩ|ΙΣΕΙΣ|ΙΣΕΙ|ΙΣΟΥΜΕ|ΙΣΕΤΕ|ΙΣΟΥΝ|ΙΣΟΥΝΕ***	Remove suffix / Check exceptions / Exit	If after removal the word is: ***ΑΝΑΜΠΑ|ΑΘΡΟ|ΕΜΠΑ|ΕΣΕ|ΕΣΩΚΛΕ|ΕΠΑ|ΞΑΝΑΠΑ|ΕΠΕ|ΠΕΡΙΠΑ|ΑΘΡΟ|ΣΥΝΑΘΡΟ|ΔΑΝΕ|ΚΛΕ|ΧΑΡΤΟΠΑ|ΕΞΑΡΧΑ|ΜΕΤΕΠΕ|ΑΠΟΚΛΕ|ΑΠΕΚΛΕ|ΕΚΛΕ|ΠΕ|ΠΕΡΙΠΑ*** Add *“Ι”*in the end
S5	Word ends in: ***ΙΣΤΟΣ|ΙΣΤΟΥ|ΙΣΤΟ|ΙΣΤΕ|ΙΣΤΟΙ|ΙΣΤΩΝ|ΙΣΤΟΥΣ|ΙΣΤΗ|ΙΣΤΗΣ|ΙΣΤΑ|ΙΣΤΕΣ***	Remove suffix / Check exceptions / Exit	If after removal the word is: ***Μ|Π|ΑΠ|ΑΡ|ΗΔ|ΚΤ|ΣΚ|ΣΧ|ΥΨ|ΦΑ|ΧΡ|ΧΤ|ΑΚΤ|ΑΟΡ|ΑΣΧ|ΑΤΑ|ΑΧΝ|ΑΧΤ|ΓΕΜ|ΓΥΡ|ΕΜΠ|ΕΥΠ|ΕΧΘ|ΗΦΑ|ΉΦΑ|ΚΑΘ|ΚΑΚ|ΚΥΛ|ΛΥΓ|ΜΑΚ|ΜΕΓ|ΤΑΧ|ΦΙΛ|ΧΩΡ*** Add *“ΙΣΤ”*in the end If after removal the word is: ***ΔΑΝΕ|ΣΥΝΑΘΡΟ|ΚΛΕ|ΣΕ|ΕΣΩΚΛΕ|ΑΣΕ|ΠΛΕ*** Add *“Ι”*in the end

S6	Word ends in: ***ΙΣΜΟ|ΙΣΜΟΙ|ΙΣΜΟΣ|ΙΣΜΟΥ|ΙΣΜΟΥΣ|ΙΣΜΩΝ***	Remove suffix / Check exceptions / Exit	* If after removal the word is: ***ΑΓΝΩΣΤΙΚ|ΑΤΟΜΙΚ|ΓΝΩΣΤΙΚ|ΕΘΝΙΚ|ΕΚΛΕΚΤΙΚ|ΣΚΕΠΤΙΚ|ΤΟΠΙΚ*** Remove *“**ΙΚ**”* from the end *If after removal the word is: ***ΣΕ|ΜΕΤΑΣΕ|ΜΙΚΡΟΣΕ|ΕΓΚΛΕ|ΑΠΟΚΛΕ*** Add *“**ΙΣΜ**”*in the end *If after removal the word is: ***ΔΑΝΕ|ΑΝΤΙΔΑΝΕ*** Add *“**Ι**”*in the end *If after removal the word is: ***ΑΛΕΞΑΝΔΡΙΝ|ΒΥΖΑΝΤΙΝ|ΘΕΑΤΡΙΝ*** Remove *“**ΙΝ**”* from the end
S7	Word ends in: ***ΑΡΑΚΙ|ΑΡΑΚΙΑ|ΟΥΔΑΚΙ|ΟΥΔΑΚΙΑ***	Remove suffix / Check exceptions / Exit	*If after removal the word is: ***Χ|Σ*** Add *“**ΑΡΑΚΙ**”*in the end

<table>
<tr><td>S8</td><td>Word ends in:
ΑΚΙ|ΑΚΙΑ|ΙΤΣΑ|ΙΤΣΑΣ|ΙΤΣΕΣ|ΙΤΣΩΝ|ΑΡΑΚΙ|ΑΡΑΚΙΑ</td><td>Remove suffix / Check exceptions / Exit</td><td>*If after removal the word is:
ΑΝΘΡ|ΒΑΜΒ|ΒΡ|ΚΑΙΜ|ΚΟΝ|ΚΟΡ|ΛΑΒΡ|ΛΟΥΛ|ΜΕΡ|ΜΟΥΣΤ|ΝΑΓΚΑΣ|ΠΛ|Ρ|ΡΥ|Σ|ΣΚ|ΣΟΚ|ΣΠΑΝ|ΤΖ|ΦΑΡΜ|Χ|ΚΑΠΑΚ|ΑΛΙΣΦ|ΑΜΒΡ|ΑΝΘΡ|Κ|ΦΥΛ|ΚΑΤΡΑΠ|ΚΛΙΜ|ΜΑΛ|ΣΛΟΒ|ΣΦ|ΤΣΕΧΟΣΛΟΒ

Add “***ΑΚ***”in the end

*If after removal the word is:
Β|ΒΑΛ|ΓΙΑΝ|ΓΛ||Ζ|ΗΓΟΥΜΕΝ|ΚΑΡΔ|ΚΟΝ|ΜΑΚΡΥΝ|ΝΥΦ||ΠΑΤΕΡ|Π||ΣΚ|ΤΟΣ|ΤΡΙΠΟΛ

Add “***ΙΤΣ***”in the end

*If after removal the word end in:
ΚΟΡ
Add “***ΙΤΣ***”in the end</td></tr>
<tr><td>S9</td><td>Word ends in:
ΙΔΙΟ|ΙΔΙΑ|ΙΔΙΩΝ</td><td>Remove suffix / Check exceptions / Exit</td><td>*If after removal the word is:
ΑΙΦΝ|ΙΡ|ΟΛΟ|ΨΑΛ

Add “***ΙΔ***”in the end

*If after removal the word iend in
Ε|ΠΑΙΧΝ

Add “***ΙΔ***”in the end</td></tr>
<tr><td>S10</td><td>Word ends in:
ΙΣΚΟΣ|ΙΣΚΟΥ|ΙΣΚΟ|ΙΣΚΕ</td><td>Remove suffix / Check exceptions / Exit</td><td>*If after removal the word is:
Δ|ΙΒ|ΜΗΝ|Ρ|ΦΡΑΓΚ|ΛΥΚ|ΟΒΕΛ

Add “***ΙΣΚ***”in the end</td></tr>
<tr><td>2a</td><td>Word ends in:
ΑΔΕΣ|ΑΔΩΝ</td><td>Remove suffix / Check exceptions / Exit</td><td>If after removal the word does not end in:
ΟΚ|ΜΑΜ|ΜΑΝ|ΜΠΑΜΠ|ΠΑΤΕΡ|ΓΙΑΓΙ|ΝΤΑΝΤ|ΚΥΡ|ΘΕΙ|ΠΕΘΕΡ

Add “***ΑΔ***” in the end</td></tr>
</table>

<table>
<tr><td>2b</td><td>Word ends in:
ΕΔΕΣ|ΕΔΩΝ</td><td>Remove suffix / Check exceptions / Exit</td><td>If after removal the word ends in:
ΟΠ|ΙΠ|ΕΜΠ|ΥΠ|ΓΗΠ|ΔΑΠ|ΚΡΑΣΠ|ΜΙΛ

Add “***ΕΔ***” in the end</td></tr>
<tr><td>2c</td><td>Word ends in:
ΟΥΔΕΣ|ΟΥΔΩΝ</td><td>Remove suffix / Check exceptions / Exit</td><td>If after removal the word ends in:
ΑΡΚ|ΚΑΛΙΑΚ|ΠΕΤΑΛ|ΛΙΧ,|ΠΛΕΧ|ΣΚ|Σ|ΦΛ|ΦΡ|ΒΕΛ|ΛΟΥΛ|ΧΝ|ΣΠ|ΤΡΑ|ΦΕ

Add “***ΟΥΔ***” in the end</td></tr>
<tr><td>2d</td><td>Word ends in:
ΕΩΣ|ΕΩΝ</td><td>Remove suffix / Check exceptions / Exit</td><td>If after removal the word is one of:
ΑΡΚ|ΚΑΛΙΑΚ|ΠΕΤΑΛ|ΛΙΧ|ΠΛΕΞ|ΣΚ|Σ|ΦΛ|ΦΡ|ΒΕΛ|ΛΟΥΛ|ΧΝ|ΣΠ|ΤΡΑΓ|ΦΕ

Add “***Ε***” in the end</td></tr>
<tr><td>3</td><td>Word ends in:
ΙΑ|ΙΟΥ|ΙΩΝ</td><td>Remove suffix</td><td>If after removal the word ends in Vowel

Add “***Ι***” in the end</td></tr>
<tr><td>4</td><td>Word ends in:
ΙΚΑ|ΙΚΟ|ΙΚΟΥ|ΙΚΩΝ</td><td>Remove suffix / Check exceptions / Exit</td><td>If after removal the word ends in Vowel
OR
is one of :
ΑΛ|ΑΔ|ΕΝΔ|ΑΜΑΝ|ΑΜΜΟΧΑΛ|ΗΘ|ΑΝΗΘ|ΑΝΤΙΔ|ΦΥΣ|ΒΡΩΜ|ΓΕΡ|ΕΞΩΔ|ΚΑΛΠ|ΚΑΛΛΙΝ|ΚΑΤΑΔ|ΜΟΥΛ|ΜΠΑΝ|ΜΠΑΓΙΑΤ|ΜΠΟΛ|ΜΠΟΣ|ΝΙΤ|ΞΙΚ|ΣΥΝΟΜΗΛ|ΠΕΤΣ|ΠΙΤΣ|ΠΙΚΑΝΤ|ΠΛΙΑΤΣ|ΠΟΣΤΕΛΝ|ΠΡΩΤΟΔ|ΣΕΡΤ|ΣΥΝΑΔ|ΤΣΑΜ|ΥΠΟΔ|ΦΙΛΟΝ|ΦΥΛΟΔ|ΧΑΣ

Add “***ΙΚ***” in the end</td></tr>
<tr><td>5a</td><td>Word is ***ΑΓΑΜΕ***</td><td>Stem is
ΑΓΑΜ
/ Exit</td><td></td></tr>
<tr><td>5a</td><td>Word ends in :
ΑΓΑΜΕ|ΗΣΑΜΕ|ΟΥΣΑΜΕ|ΗΚΑΜΕ|ΗΘΗΚΑΜΕ</td><td>Remove suffix / Check exceptions / Exit</td><td></td></tr>
</table>

5a	Word ends in: ***ΑΜΕ***	Remove suffix / Check exceptions / Exit	If after removal the word is one of: ***ΑΝΑΠ|ΑΠΟΘ|ΑΠΟΚ|ΑΠΟΣΤ|ΒΟΥΒ|ΞΕΘ|ΟΥΛ|ΠΕΘ|ΠΙΚΡ|ΠΟΤ|ΣΙΧ|Χ*** Add ***"ΑΜ"*** in the end
5b	Word ends in: ***ΑΓΑΝΕ|ΗΣΑΝΕ|ΟΥΣΑΝΕ|ΙΟΝΤΑΝΕ|ΙΟΤΑΝΕ|ΙΟΥΝΤΑΝΕ|ΟΝΤΑΝΕ|ΟΤΑΝΕ|ΟΥΝΤΑΝΕ|ΗΚΑΝΕ|ΗΘΗΚΑΝΕ***	Remove suffix / Check exceptions / Exit	If after removal the word one of: ***ΤΡ|ΤΣ*** Add "***ΑΓΑΝ"***

5b	Word ends in: ***ΑΓΑΝΕ|ΗΣΑΝΕ| ΟΥΣΑΝΕ|ΙΟΝΤΑΝΕ| ΙΟΤΑΝΕ|ΙΟΥΝΤΑΝΕ| ΟΝΤΑΝΕ|ΟΤΑΝΕ| ΟΥΝΤΑΝΕ|ΗΚΑΝΕ| ΗΘΗΚΑΝΕ***	Remove suffix / Check exceptions / Exit	☐ If after removal the word ends in Vowel without "***Υ***" OR is one of : ***ΒΕΤΕΡ|ΒΟΥΛΚ|ΒΡΑΧΜ|Γ| ΔΡΑΔΟΥΜ|Θ|ΚΑΛΠΟΥΖ| ΚΑΣΤΕΛ|ΚΟΡΜΟΡ| ΛΑΟΠΛ|ΜΩΑΜΕΘ|Μ| ΜΟΥΣΟΥΛΜ|Ν|ΟΥΛ|Π| ΠΕΛΕΚ|ΠΛ|ΠΟΛΙΣ| ΠΟΡΤΟΛ|ΣΑΡΑΚΑΤΣ| ΣΟΥΛΤ|ΤΣΑΡΛΑΤ|ΟΡΦ| ΤΣΙΓΓ|ΤΣΟΠ| ΦΩΤΟΣΤΕΦ|Χ|ΨΥΧΟΠΛ| ΑΓ|ΟΡΦ|ΓΑΛ|ΓΕΡ|ΔΕΚ| ΔΙΠΛ|ΑΜΕΡΙΚΑΝ|ΟΥΡ| ΠΙΘ|ΠΟΥΡΙΤ|Σ|ΖΩΝΤ|ΙΚ| ΚΑΣΤ|ΚΟΠ|ΛΙΧ|ΛΟΥΘΗΡ| ΜΑΙΝΤ|ΜΕΛ|ΣΙΓ|ΣΠ| ΣΤΕΓ|ΤΡΑΓ|ΤΣΑΓ|Φ|ΕΡ| ΑΔΑΠ|ΑΘΙΓΓ|ΑΜΗΧ| ΑΝΙΚ|ΑΝΟΡΓ|ΑΠΗΓ| ΑΠΙΘ|ΑΤΣΙΓΓ|ΒΑΣ| ΒΑΣΚ|ΒΑΘΥΓΑΛ| ΒΙΟΜΗΧ|ΒΡΑΧΥΚ|ΔΙΑΤ| ΔΙΑΦ|ΕΝΟΡΓ|ΘΥΣ| ΚΑΠΝΟΒΙΟΜΗΧ| ΚΑΤΑΓΑΛ|ΚΛΙΒ| ΚΟΙΛΑΡΦ|ΛΙΒ| ΜΕΓΛΟΒΙΟΜΗΧ| ΜΙΚΡΟΒΙΟΜΗΧ|ΝΤΑΒ| ΞΗΡΟΚΛΙΒ|ΟΛΙΓΟΔΑΜ| ΟΛΟΓΑΛ|ΠΕΝΤΑΡΦ| ΠΕΡΗΦ|ΠΕΡΙΤΡ|ΠΛΑΤ| ΠΟΛΥΔΑΠ|ΠΟΛΥΜΗΧ| ΣΤΕΦ|ΤΑΒ|ΤΕΤ| ΥΠΕΡΗΦ|ΥΠΟΚΟΠ| ΧΑΜΗΛΟΔΑΠ| ΨΗΛΟΤΑΒ*** Add "***ΑΝ***" in the end
5c	Word ends in: ***ΗΣΕΤΕ***	Remove suffix / Check exceptions / Exit	

5c	Word ends in: ***ΕΤΕ***	Remove suffix / Check exceptions / Exit	□ If after removal the word ends in Vowel without “***Υ***” OR is one of : □ ***ΑΒΑΡ|ΒΕΝ|ΕΝΑΡ|ΑΒΡ|ΑΔ|ΑΘ|ΑΝ|ΑΠΛ|ΒΑΡΟΝ|ΝΤΡ|ΣΚ|ΚΟΠ|ΜΠΟΡ|ΝΙΦ|ΠΑΓ|ΠΑΡΑΚΑΛ|ΣΕΡΠ|ΣΚΕΛ|ΣΥΡΦ|ΤΟΚ|Υ|Δ|ΕΜ|ΘΑΡΡ|Θ*** OR ends in : ***ΟΔ|ΑΙΡ|ΦΟΡ|ΤΑΘ|ΔΙΑΘ|ΣΧ|ΕΝΔ|ΕΥΡ|ΤΙΘ|ΥΠΕΡΘ|ΡΑΘ|ΕΝΘ|ΡΟΘ|ΣΘ|ΠΥΡ|ΑΙΝ|ΣΥΝΔ|ΣΥΝ|ΣΥΝΘ|ΧΩΡ|ΠΟΝ|ΒΡ|ΚΑΘ|ΕΥΘ|ΕΚΘ|ΝΕΤ|ΡΟΝ|ΑΡΚ|ΒΑΡ|ΒΟΛ|ΩΦΕΛ*** Add “***ΕΤ***” in the end
5d	Word ends in: ***ΟΝΤΑΣ|ΩΝΤΑΣ***	Remove suffix / Check exceptions / Exit	If after removal the word is: ***ΑΡΧ*** add “***ΟΝΤ***” in the end OR If after removal the word is: ***ΚΡΕ*** add “**Ω*ΝΤ***” in the end
5e	Word ends in: ***ΟΜΑΣΤΕ|ΙΟΜΑΣΤΕ***	Remove suffix / Check exceptions / Exit	If after removal the word is: ***ΟΝ*** add “***ΟΜΑΣΤ***” in the end
5f	Word ends in: ***ΙΕΣΤΕ***	Remove suffix / Check exceptions / Exit	If after removal the word is one of: ***Π|ΑΠ|ΣΥΜΠ|ΑΣΥΜΠ|ΑΚΑΤΑΠ|ΑΜΕΤΑΜΦ*** *Add **“ΙΕΣΤ”** in the end*
5f	Word ends in: ***ΕΣΤΕ***	Remove suffix / Check exceptions / Exit	If after removal the word is one of: ***ΑΛ|ΑΡ|ΕΚΤΕΛ|Ζ|Μ|Ξ|ΠΑΡΑΚΑΛ|ΑΡ|ΠΡΟ|ΝΙΣ*** *Add **“ΕΣΤ”** in the end*

5g	Word ends in: **ΗΘΗΚΑ|ΗΘΗΚΕΣ|ΗΘΗΚΕ**	Remove suffix / Check exceptions / Exit	
5g	Word ends in: **ΗΚΑ|ΗΚΕΣ|ΗΚΕ**	Remove suffix / Check exceptions / Exit □	□ If after removal the word is one of: ***ΔΙΑΘ|Θ|ΠΑΡΑΚΑΤΑΘ|ΠΡΟΣΘ|ΣΥΝΘ*** OR ends in: ***ΣΚΩΛ|ΣΚΟΥΛ|ΝΑΡΘ|ΣΦ|ΟΘ|ΠΙΘ*** □ Add "***ΗΚ***" in the end
5h	Word ends in: **ΟΥΣΑ|ΟΥΣΕΣ|ΟΥΣΕ**	Remove suffix / Check exceptions / Exit □	□ If after removal the word is one of: ***ΦΑΡΜΑΚ|ΧΑΔ|ΑΓΚ|ΑΝΑΡΡ|ΒΡΟΜ|ΕΚΛΙΠ|ΛΑΜΠΙΔ|ΛΕΧ|Μ|ΠΑΤ|Ρ|Λ|ΜΕΔ|ΜΕΣΑΖ|ΥΠΟΤΕΙΝ|ΑΜ|ΑΙΘ|ΑΝΗΚ|ΔΕΣΠΟΖ|ΕΝΔΙΑΦΕΡ|ΔΕ|ΔΕΥΤΕΡΕΥ|ΚΑΘΑΡΕΥ|ΠΛΕ|ΤΣΑ*** OR ends in: ***ΠΟΔΑΡ|ΒΛΕΠ|ΠΑΝΤΑΧ|ΦΡΥΔ|ΜΑΝΤΙΛ|ΜΑΛΛ|ΚΥΜΑΤ|ΛΑΧ|ΛΗΓ|ΦΑΓ|ΟΜ|ΠΡΩΤ*** □ Add "***ΟΥΣ***" in the end

5i	Word ends in: **ΑΓΑ|ΑΓΕΣ|ΑΓΕ**	Remove suffix / Check exceptions / Exit	☐ If after removal the word is one of: ***ΑΒΑΣΤ|ΠΟΛΥΦ|ΑΔΗΦ|ΠΑΜΦ|Ρ|ΑΣΠ|ΑΦ|ΑΜΑΛ|ΑΜΑΛΛΙ|ΑΝΥΣΤ|ΑΠΕΡ|ΑΣΠΑΡ|ΑΧΑΡ|ΔΕΡΒΕΝ|ΔΡΟΣΟΠ|ΞΕΦ|ΝΕΟΠ|ΝΟΜΟΤ|ΟΛΟΠ|ΟΜΟΤ|ΠΡΟΣΤ|ΠΡΟΣΩΠΟΠ|ΣΥΜΠ|ΣΥΝΤ|Τ|ΥΠΟΤ|ΧΑΡ|ΑΕΙΠ|ΑΙΜΟΣΤ|ΑΝΥΠ|ΑΠΟΤ|ΑΡΤΙΠ|ΔΙΑΤ|ΕΝ|ΕΠΙΤ|ΚΡΟΚΑΛΟΠ|ΣΙΔΗΡΟΠ|Λ|ΝΑΥ|ΟΥΛΑΜ|ΟΥΡ|Π|ΤΡ|Μ*** ☐ OR ends in: ***ΟΦ|ΠΕΛ|ΧΟΡΤ|ΛΛ|ΣΦ|ΡΠ|ΦΡ|ΠΡ|ΛΟΧ|ΣΜΗΝ*** BUT is not one of: ***ΨΟΦ|ΝΑΥΛΟΧ*** AND does not end in: ***ΚΟΛΛ*** Add "***ΑΓ***" in the end
5j	Word ends in: ***ΗΣΕ|ΗΣΟΥ|ΗΣΑ***	Remove suffix / Check exceptions / Exit	☐ ☐ ☐ ☐ If after removal the word is one of: ***Ν|ΧΕΡΣΟΝ|ΔΩΔΕΚΑΝ|ΕΡΗΜΟΝ|ΜΕΓΑΛΟΝ|ΕΠΤΑΝ*** Add "***ΗΣ***" in the end
5k	Word ends in: ***ΗΣΤΕ***	Remove suffix / Check exceptions / Exit	☐ ☐ ☐ ☐ If after removal the word is one of: ***ΑΣΒ|ΣΒ|ΑΧΡ|ΧΡ|ΑΠΛ|ΑΕΙΜΝ|ΔΥΣΧΡ|ΕΥΧΡ|ΚΟΙΝΟΧΡ|ΠΑΛΙΜΨ*** Add "***ΗΣΤ***" in the end
5l	Word ends in: ***ΟΥΝΕ|ΗΣΟΥΝΕ|ΗΘΟΥΝΕ***	Remove suffix / Check exceptions / Exit	☐ ☐ ☐ ☐ If after removal the word is one of: ***Ν|Ρ|ΣΠΙ|ΣΤΡΑΒΟΜΟΥΤΣ|ΚΑΚΟΜΟΥΤΣ|ΕΞΩΝ*** Add "***ΟΥΝ***" in the end

51	Word ends in: ***ΟΥΜΕ|ΗΣΟΥΜΕ| ΗΘΟΥΜΕ***	Remove □ suffix / □ Check exceptions□ / Exit □	If after removal the word is one of: ***ΠΑΡΑΣΟΥΣ|Φ|Χ|ΩΡΙΟΠΛ|ΑΖ| ΑΛΛΟΣΟΥΣ|ΑΣΟΥΣ*** Add “***ΟΥΜ***” in the end
6	Word ends in: ***ΜΑΤΑ|ΜΑΤΩΝ|ΜΑΤΟΣ***	Remove □ suffix □	Always Add “***ΜΑ***” in the end
6	Word ends in: ***Α|ΑΓΑΤΕ|ΑΓΑΝ|ΑΕΙ|ΑΜΑΙ|ΑΝ|ΑΣ|ΑΣΑΙ|ΑΤΑΙ|ΑΩ|Ε|ΕΙ|ΕΙΣ| ΕΙΤΕ|ΕΣΑΙ|ΕΣ|ΕΤΑΙ|Ι|ΙΕΜΑΙ|ΙΕΜΑΣΤΕ|ΙΕΤΑΙ|ΙΕΣΑΙ| ΙΕΣΑΣΤΕ|ΙΟΜΑΣΤΑΝ|ΙΟΜΟΥΝ|ΙΟΜΟΥΝΑ|ΙΟΝΤΑΝ| ΙΟΝΤΟΥΣΑΝ|ΙΟΣΑΣΤΑΝ|ΙΟΣΑΣΤΕ|ΙΟΣΟΥΝ|ΙΟΣΟΥΝΑ| ΙΟΤΑΝ|ΙΟΥΜΑ|ΙΟΥΜΑΣΤΕ|ΙΟΥΝΤΑΙ|ΙΟΥΝΤΑΝ|Η|ΗΔΕΣ| ΗΔΩΝ|ΗΘΕΙ|ΗΘΕΙΣ|ΗΘΕΙΤΕ|ΗΘΗΚΑΤΕ|ΗΘΗΚΑΝ|ΗΘΟΥΝ| ΗΘΩ|ΗΚΑΤΕ|ΗΚΑΝ|ΗΣ|ΗΣΑΝ|ΗΣΑΤΕ|ΗΣΕΙ|ΗΣΕΣ|ΗΣΟΥΝ| ΗΣΩ|Ο|ΟΙ|ΟΜΑΙ|ΟΜΑΣΤΑΝ|ΟΜΟΥΝ|ΟΜΟΥΝΑ|ΟΝΤΑΙ| ΟΝΤΑΝ|ΟΝΤΟΥΣΑΝ|ΟΣ|ΟΣΑΣΤΑΝ|ΟΣΑΣΤΕ|ΟΣΟΥΝ| ΟΣΟΥΝΑ|ΟΤΑΝ|ΟΥ|ΟΥΜΑΙ|ΟΥΜΑΣΤΕ|ΟΥΝ|ΟΥΝΤΑΙ| ΟΥΝΤΑΝ|ΟΥΣ|ΟΥΣΑΝ|ΟΥΣΑΤΕ|Υ|ΥΣ|Ω|ΩΝ***	Remove suffix	
7	Word ends in: ***ΕΣΤΕΡ|ΕΣΤΑΤ|ΟΤΕΡ| ΟΤΑΤ|ΥΤΕΡ|ΥΤΑΤ|ΩΤΕΡ| ΩΤΑΤ***	Remove □ suffix	
UL2	Word has converted letters to upper case from step UL1	Convert these letters back to lower case	

6. Final Evaluation

After the implementation of all the modifications that would improve the existing algorithm, we begun to evaluate our revised algorithm. We used the same word list and the same applications we created for testing the initial algorithm. By doing so, we made sure that the statistics produced can be directly comparable.

As a result of our modifications, we ended with a stronger stemmer. Although the two stemmers leave unchanged roughly the same number of words, our modified version produces fewer and bigger conflation classes by altering more letters in every word, on average. Table 9 presents the statistics gathered after executing both algorithms against a list of 574.621 Greek words.

Table 9: Statistics: Comparison of the original and revised algorithm

	Original Stemmer by Ntais	**Our modified Stemmer**
Mean number of words per conflation class	4,055	5,664
Index compression factor	75,34%	82,34
Ratio of unchanged to total words	2%	2%
Mean modified Humming Distance	2.441	2,916
Median Modified Humming Distance	2	2
Correct Stems (for a sample of 12468 words)	10885 (87,3%)	11669 (93,52%)
Distribution of Stemming errors per algorithm		
	Original Stemmer by Ntais	**Our modified Stemmer**
Understemming Errors (Section 3.2)	88,44%	23,67%
Overstemming Errors (Section 3.2)	11,56%	76,33%
Number of different stems generated by the two stemmers (for a sample of 574.621 words)	35885 (6,24%)	

It is worth noting that, according to our tests, the majority of the errors of the initial algorithm had to do with understemming (88,44%). Our algorithm produces more overstemming errors (76,33%) despite of the fact that the total number of errors is reduced. The two stemmers produced 35885 different stems for the same list of words.

In addition the number of executions steps was increased from 29 in the original algorithm to 42. Ten of the new execution steps have to do with the 72 new stems that were added while the remaining three deal with stop-word removal and lower to upper and upper to lower case treatment.

Although the number of steps was increased by approximately 44%, the algorithm now executes 23.17 steps on average. The reason is that while the original algorithm always executes all of its 29 steps, our modified algorithms returns the correct stem and then exits earlier if the remaining rules are not going to modify the word any further. For example, if a rule that treats verb suffixes is evaluated and executed, it is certain that no rules that deal with noun suffixes will be executed later. This modification will certainly offer better running times in any implementation of our algorithm. Nevertheless, we avoid mentioning any running times and we only refer to the average number of steps executed. The reason is that the actual running time of an algorithm depends on factors like the implementation language, coding style and coding efficiency, hardware, current load of the system and operating system among others. Because of that, we believe that comparing running times of our tests between our algorithm and the initial algorithm by Ntais would be misleading.

7. Conclusion

Our purpose has been to evaluate and improve the existing stemming algorithm of the Greek language. After an evaluation of the results that the original algorithm provided, we incrementally improved it by adding new rules and exceptions. Overall, we managed to gain significant improvements in completeness and accuracy to the existing algorithm.

After manually checking the output of the stemmer we conclude that our new algorithm returns more correct results than its predecessor. Due to our efforts the understemming and overstemming errors are less. The new algorithm is more complete since it supports most tenses and correctly stems suffixes that were not included before, like diminutives and others.

Moreover, we offered a more usable implementation that can be used with slight modifications or even directly. Due to the implementation language that we chose, PHP, our implementation can be used by any web or non web application that may require stemming of Greek words. Our implementation is directly usable by any kind of application, web or not, linguistic or IR related that requires stemming for Greek.

To answer our research question, we conclude that the addition of more suffixes is attainable but the effort required for each additional suffix increases geometrically. The previous algorithm already deals with the majority of suffixes that can be found in the Greek grammar. Since the majority of cases is already covered, each addition to the stemmer requires a considerable amount of evaluating the stemmer's output and searching in dictionaries for possible exceptions.

Despite of the fact that we are pleased with the outcome of our efforts, there is a lot of space for improvements. Our algorithm, like its predecessor, is not dealing with the moving tone-mark issue. Any attempt to deal with issue will probably require a considerable amount of effort due to the complexity of the Greek language. Although it will not provide any great improvement in precision, a stemmer must successfully deal with this issue in order to be considered complete. Furthermore, the stemmer can be enhanced by the addition of more suffixes and exceptions. In addition to the 158 suffixes of the initial algorithm, we added rules for 72 more. Although the majority of the cases is covered, more rules can be added in order to produce a more complete stemmer. Finally, the stemmer can be more thoroughly tested. Although we used metrics to measure our performance and compare it with its predecessor, we had to manually check the stems generated for overstemming and understemming errors. This is a task that can only be done manually, by experts of the Greek language. Although we had an enormous collection of Greek words, we were able to test only one small portion of it, as we mention in Table 9. A more complete testing will indicate more errors and can serve as a basis for further improvement.

We believe that this thesis work contributed in stemming research by continuing and extending the work done by others in the past and by offering a stemmer for the Greek language which others can use and extend even more.

References

[Alvares *et al*, 2005] Alvares Reinaldo Viana , Garcia Ana Cristina Bicharra, & Ferraz Inhaúma. STEMBR: A Stemming Algorithm for the Brazilian Portuguese Language. In: *Proc. of the12th Portuguese Conference on Artificial Intelligence,* EPIA 2005, *Lecture Notes in Artificial Intelligence* 3808, 693-701.

[Baeza-Yates & Ribeiro-Neto, 1999] Baeza-Yates Ricardo & Ribeiro-Neto Berthirer. *Modern Information Retrieval.* Addison Wesley, New York, 1999.

[Baugh & Cable,2001] Baugh Albert & Cable Thomas. *A History of the English Language.* Prentice Hall, Upper Saddle River, 2001.

[Brants, 2003] Brants Thorsten. Natural Language Processing in Information Retrieval. In *Proceedings of the 14th Meeting of Computational Linguistics in the Netherlands* Antwerp, Belgium.

[Carlberger *et al*, 2001] Carlberger Johan, Dalianis Hercules, Hassel Martin & Knutsson Ola. Improving Precision in Information Retrieval for Swedish using Stemming. In *Proceedings of NODALIDA 01 - 13th Nordic Conference on Computational Linguistics*, May 21-22, Uppsala, Sweden.

[Flanagan, 2004] Flanagan David. *Javascript: The Definitive Guide.* O'Reilly, Sebastopol , CA.

[Frakes, 2003] Frakes William B. *S*trength and Similarity of Affix Removal Stemming Algorithms. *ACM SIGIR Forum archive*, 37, 1 (Spring 2003), 26 – 30.

[Google, 2003] Google Starts Auto Stemming Searches. http://www.searchengineshowdown.com/blog/2003/11/google_starts_auto_stemming_se.sh (Retrieved 04/05/2008).

[Krovetz, 1993] Krovetz Robert. Viewing morphology as an inference process. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 191-202.

[Kurz & Kilian, 2001] Kurz Thorsten & Stoffel Killian.Going beyond Stemming: Creating Concept Signatures of Complex Medical Terms. *Knowledge Based Systems Journal* 15, 5 (2001), 309-313.

[Lazarinis, 2005] Lazarinis Fotis. Do search engines understand Greek or users requests “sound Greek” to them ? In: *M. Beigbeder and W.G. Yee (eds), Open Source Web*

Information Retrieval Workshop in conjunction with IEEE/WIC/ACM International Conference on Web Intelligence & Intelligent Agent Technology, Compiegne, France, 19 September 2005 (IEEE, 2005) 43-6.

[Lazarinis, 2007] Lazarinis Fotis. Lemmatization and stopword elimination in Greek web searching. *Proceedings of the 2007 Euro American Conference on Telematics and information Systems EATIS '07*, ACM, New York, NY.

[Lerdorf & Tatroe, 2002] Lerdorf Rasmus & Tatroe Kevin. *Programming PHP*. O'Reilly, Sebastopol , CA.

[Lewis & Papadimitriou, 1998] Lewis R. H. & Papadimitriou. H. C. *Elements of the Theory of Computation*. Prentice Hall, Upper Saddle River, 1998.

[Lovins, 1968] Lovins JB. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11, 1 (1968). 22-31.

[Mackridge, 1987] Mackridge P. *The Modern Greek Language. A Descriptive Analysis of Standard Modern Greek.* Clarenton Press, Oxford.

[Ntais, 2006] Ntais Georgios. *Development of a stemmer for the Greek language*. Master's Thesis at Stockholm University / Royal Institute of Technology.

[Ntais, 2008] Ntais Georgios. Online Greek Stemmer. Web Interface implemented in Javascript http://people.dsv.su.se/~hercules/greek_stemmer.gr.html (Retrieved 15/04/2008).

[Paice, 1994] Paice D. Chris. An Evaluation Method for Stemming Algorithms. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 42-50.

[Paice, 1996] Paice D. Chris. Method for Evaluation of Stemming Algorithms Based on Error Counting. *Journal of the American Society for Information Science,* Volume 47 , 8 (August 1996), 632 – 649.

[Porter, 1980] Porter Michael. An algorithm for suffix stripping. *Program*, 14(3), 130-137.

[Risvik *et al*, 2003] Risvik Knut Magne, Mikolajewski Tomasz & Boros Peter.Query Segmentation for Web Search. In *Proceedings of the 12th International World Wide Web Conference*, 52.

[Robinson, 1995] Robinson Andrew. *The Story of Writing*. Thames & Hudson

[Triantafyllidis, 1941] Triantafyllidis Manolis. *Modern Greek Grammar*. Institute M

Triantalyllidis.

[van Rijsbergen, 1979] van Rijsbergen, C.J. *Information Retrieval* .Butterworths, London.

[Witten et al, 2007] Witten Ian H., Gori Marco & Numerico Teresa. *Web Dragons: Inside the Myths of Search Engine Technology*. Morgan Kaufmann Publishers.

[Xu & Croft, 1998] Xu Jinxi & Croft Bruce. Corpus-Based Stemming Using Cooccurrence of Word Variants. *ACM Transactions on Information Systems*, 16, 1 (January 1998), 61-81.

[Xu & Croft, 2000] Xu Jinxi & Croft Bruce. Improving the Effectiveness of Information Retrieval with Local Context Analysis. *ACM Transactions on Information Systems* 18, 1 (January 2000), 79-112.

Appendices

Appendix A: Verb Conjugation Classes In Greek

Verbs of 1st Conjugation Classes

The verb "to read" in Greek Active Voice / Past Tenses						
	Υπερσυντέλικος (Past Perfect)		**Αόριστος (Simple Past)**		**Παρατατικός (Past continuous)**	
Singular	είχα διαβά**σει**	(I) had studied	διάβα**σα**	(I) studied	διάβα**ζα**	(I) was studying
	είχες διαβά**σει**	(you) had studied	διάβα**σες**	(you) studied	διάβα**ζες**	(you) were studying
	είχε διαβά**σει**	(he) had studied	διάβα**σε**	(he) studied	διάβα**ζε**	(he) was studying
Plural	είχαμε διαβά**σει**	(we) had studied	διαβά**σαμε**	(we) studied	διαβά**ζαμε**	(we) were studying
	είχατε διαβά**σει**	(you) had studied	διαβά**σατε**	(you) studied	διαβά**ζατε**	(you) were studying
	είχαν διαβά**σει**	(they) had studied	διαβά**σανε** (διάβα**σαν**)	(they) studied	διάβα**ζαν**	(they) were studying

The verb "to read" in Greek Active Voice / Present Tenses				
	Ενεστώτας **(Present)**		**Παρακείμενος** **(Present Perfect)**	
Singular	διαβά**ζω**	(I) read /am reading	έχω διαβά**σει**	(I) have read /have been reading
	διαβά**ζεις**	(you) read / are reading	έχεις διαβά**σει**	(you) have read/have been reading
	διαβά**ζει**	(he) reads/ is reading	έχει διαβά**σει**	(he) has read/has been reading
Plural	διαβά**ζουμε**	(we) read/ are reading	έχουμε διαβά**σει**	(we) have read/have been reading
	διαβά**ζετε**	(you) read / are reading	έχετε διαβά**σει**	(you) have read/have been reading
	διαβά**ζουν**	(they) read /are reading	έχουν διαβά**σει**	(they) have read/have been reading

The verb “to read” in Greek Active Voice / Future Tenses						
	Εξακολουθητικός Μέλλοντας (Future Continuous)		**Στιγμιαίος Μέλλοντας (Simple Future)**		**Συντελεσμένος Μέλλοντας (Future Perfect/Imperfect)**	
Singular	θα διαβάζ**ω**	(I) will be reading	θα διαβά**σω**	(I) will read	θα έχω διαβά**σει**	(I) will have read / have been reading
	θα διαβάζ**εις**	(you) will be reading	θα διαβά**σεις**	(you) will read	θα έχεις διαβά**σει**	(you) will have read/have been reading
	θα διαβάζ**ει**	(he) will be reading	θα διαβά**σει**	(he) will read	θα έχει διαβά**σει**	(he) will have read/ have been reading
Plural	θα διαβάζ**ουμε**	(we)will reading	θα διαβά**σουμε**	(we) will read	θα έχουμε διαβά**σει**	(we) will have read/have been reading
	θα διαβάζ**ετε**	(you)will be reading	θα διαβά**σετε**	(you) will read	θα έχετε διαβά**σει**	(you) will have read/have been reading
	θα διαβάζ**ουν**	(they) will be reading	θα διαβά**σουνε**	(they) will read	θα έχουν διαβά**σει**	(they) will have read/have been reading

The verb "to read" in Greek Passive Voice / Past Tenses						
	Υπερσυντέλικος (Past Perfect)		**Αόριστος (Simple Past)**		**Παρατατικός (Past Continuous)**	
Singular	είχα διαβα**στεί**	(I) had been studied	διαβά**στηκα**	(I) was studied	διαβα**ζόμουν**	(I) was being studied
	είχες διαβα**στεί**	(you) had been studied	διαβά**στηκες**	(you) were studied	διαβα**ζόσουν**	(you) were being studied
	είχε διαβα**στεί**	(he) had been studied	διαβά**στηκε**	(he) was studied	διαβα**ζόταν**	(he) was being studied
Plural	είχαμε διαβα**στεί**	(we) had been studied	διαβα**στήκαμε**	(we) were studied	διαβα**ζόμαστε**	(we) were being studied
	είχατε διαβα**στεί**	(you) had been studied	διαβα**στήκατε**	(you) were studied	διαβα**ζόσαστε**	(you) were being studied
	είχαν διαβα**στεί**	(they) had been studied	διαβα**στήκανε** (διαβά**στηκαν**)	(they) were studied	διαβά**ζονταν**	(they) were being studied

The verb "to read" in Greek Active Voice / Present Tenses				
	Ενεστώτας (Present)		**Παρακείμενος (Present Perfect)**	
Singular	διαβά**ζομαι**	(I) am read /am being read	έχω διαβα**στεί**	(I) have been read
	διαβά**ζεσαι**	(you) are read / are being read	έχεις διαβα**στεί**	(you) have been read
	διαβά**ζεται**	(he) is read/ is being read	έχει διαβα**στεί**	(he) has been read
Plural	διαβα**ζόμαστε**	(we) are read/ are being read	έχουμε διαβα**στεί**	(we) have been read
	διαβά**ζεστε**	(you) are read / are being read	έχετε διαβα**στεί**	(you) have been read
	διαβά**ζονται**	(they) are read /are being read	έχουν διαβα**στεί**	(they) have been read

<table>
<tr><th colspan="7">The verb “to read” in Greek
Active Voice / Future Tenses</th></tr>
<tr><th></th><th colspan="2">Εξακολουθητικός Μέλλοντας
(Future Continuous)</th><th colspan="2">Στιγμιαίος Μέλλοντας
(Simple Future)</th><th colspan="2">Συντελεσμένος Μέλλοντας
(Future Perfect/Imperfect)</th></tr>
<tr><td rowspan="3">Singular</td><td>θα διαβάζομαι</td><td>(I) will be being reading</td><td>θα διαβαστώ</td><td>(I) will be read</td><td>θα έχω διαβαστεί</td><td>(I) will have been read</td></tr>
<tr><td>θα διαβάζεσαι</td><td>(you) will be being read</td><td>θα διαβαστείς</td><td>(you) will be read</td><td>θα έχεις διαβαστεί</td><td>(you) will have been read</td></tr>
<tr><td>θα διαβάζεται</td><td>(he) will be being read</td><td>θα διαβαστεί</td><td>(he) will be read</td><td>θα έχει διαβαστεί</td><td>(he) will have been read</td></tr>
<tr><td rowspan="3">Plural</td><td>θα διαβαζόμαστε</td><td>(we) will be being read</td><td>θα διαβαστούμε</td><td>(we) will be read</td><td>θα έχουμε διαβαστεί</td><td>(we) will have been read</td></tr>
<tr><td>θα διαβαζόσαστε</td><td>(you) will be being read</td><td>θα διαβαστείτε</td><td>(you) will be read</td><td>θα έχετε διαβαστεί</td><td>(you) will have been read</td></tr>
<tr><td>θα διαβάζονται</td><td>(they) will be being read</td><td>θα διαβαστούν</td><td>(they) will be read</td><td>θα έχουν διαβαστεί</td><td>(they) will have been read</td></tr>
</table>

Verbs of 2nd Conjugation Class

The verb "to love" in Greek Active Voice / Past Tenses						
	Υπερσυντέλικος (Past Perfect)		**Αόριστος (Simple Past)**		**Παρατατικός (Past continuous)**	
Singular	είχα αγαπ**ήσει**	(I) had loved	αγάπ**ησα**	(I) loved	αγαπ**ούσα** (αγάπ**αγα**)	(I) was loving
	είχες αγαπ**ήσει**	(you) had loved	αγάπ**ησες**	(you) loved	αγαπ**ούσες** (αγάπ**αγες**)	(you) were loving
	είχε αγαπ**ήσει**	(he) had loved	αγάπ**ησε**	(he) loved	αγαπ**ούσε** (αγάπ**αγε**)	(he) was loving
Plural	είχαμε αγαπ**ήσει**	(we) had loved	αγαπ**ήσαμε**	(we) loved	αγαπ**ούσαμε** (αγαπ**άγαμε**)	(we) were loving
	είχατε αγαπ**ήσει**	(you) had loved	αγαπ**ήσατε**	(you) loved	αγαπ**ούσατε** (αγαπ**άγατε**)	(you) were loving
	είχαν αγαπ**ήσει**	(they) had loved	αγάπη**σαν**	(they) loved	αγαπ**ούσαν** (αγάπ**αγαν**)	(they) were loving

The verb "to love" in Greek Active Voice / Present Tenses				
	Ενεστώτας (Present)		**Παρακείμενος (Present Perfect)**	
Singular	αγαπ**ώ** (αγαπ**άω**)	(I) love /am loving	έχω αγαπ**ήσει**	(I) have loved /have been loving
	αγαπ**άς**	(you) love / are loving	έχει αγαπ**ήσει**	(you) have loved/have been loving
	αγαπ**ά** **(αγαπάει)**	(he) loves/ is loving	έχει αγαπ**ήσει**	(he) has loved/has been loving
Plural	αγαπ**ούμε** (αγαπ**άμε**)	(we) love / are loving	έχουμε αγαπ**ήσει**	(we) have loved/have been loving
	αγαπ**άτε**	(you) love / are loving	έχετε αγαπ**ήσει**	(you) have loved/have been loving
	αγαπ**άνε**	(they) love /are loving	έχουν αγαπ**ήσει**	(they) have loved/have been loving

The verb "to love" in Greek Active Voice / Future Tenses						
	Εξακολουθητικός Μέλλοντας (Future Continuous)		**Στιγμιαίος Μέλλοντας (Simple Future)**		**Συντελεσμένος Μέλλοντας (Future Perfect/Imperfect)**	
Singular	θα αγαπ**ώ** (θα αγαπ**άω**)	(I) will be loving	θα αγαπ**ήσω**	(I) will love	θα έχω αγαπ**ήσει**	(I) will have loved /have been loving
	θα αγαπ**άς**	(you) will be loving	θα αγαπ**ήσεις**	(you) will love	θα έχει αγαπ**ήσει**	(you) will have loved/have been loving
	θα αγαπ**άει**	(he) will be loving	θα αγαπ**ήσει**	(he) will love	θα έχει αγαπ**ήσει**	(he) will has loved/has been loving
Plural	θα αγαπ**ούμε** **(θα αγαπάμε)**	(we)will loving	θα αγαπ**ήσουμε**	(we) will love	θα έχουμε αγαπ**ήσει**	(we) will have loved/have been loving
	θα αγαπ**άτε**	(you)will be loving	θα αγαπ**ήσετε**	(you) will love	θα έχετε αγαπ**ήσει**	(you) will have loved/have been loving
	θα αγαπ**ούν** (θα αγαπ**άνε**)	(they) will be loving	θα αγαπ**ήσουν**	(they) will love	θα έχουν αγαπ**ήσει**	(they) will have loved/have been loving

<table>
<tr><th colspan="7">The verb “to love” in Greek
Passive Voice / Past Tenses</th></tr>
<tr><th></th><th colspan="2">Υπερσυντέλικος
(Past Perfect)</th><th colspan="2">Αόριστος (Simple Past)</th><th colspan="2">Παρατατικός
(Past continuous)</th></tr>
<tr><td rowspan="3">Singular</td><td>είχα αγαπηθεί</td><td>(I) had been loved</td><td>αγαπήθηκα</td><td>(I) was loved</td><td>αγαπιόμουν</td><td>(I) was being loved</td></tr>
<tr><td>είχες αγαπηθεί</td><td>(you) had been loved</td><td>αγαπήθηκες</td><td>(you) were loved</td><td>αγαπιόσουν</td><td>(you) were being loved</td></tr>
<tr><td>είχε αγαπηθεί</td><td>(he) had been loved</td><td>αγαπήθηκε</td><td>(he) was loved</td><td>αγαπιόταν</td><td>(he) was being loved</td></tr>
<tr><td rowspan="3">Plural</td><td>είχαμε αγαπηθεί</td><td>(we) had been loved</td><td>αγαπηθήκαμε</td><td>(we) were loved</td><td>αγαπιόμαστε
(αγαπιόμασταν)</td><td>(we) were being loved</td></tr>
<tr><td>είχατε αγαπηθεί</td><td>(you) had been loved</td><td>αγαπηθήκατε</td><td>(you) were loved</td><td>αγαπιόσαστε
(αγαπιόσασταν)</td><td>(you) were being loved</td></tr>
<tr><td>είχαν αγαπηθεί</td><td>(they) had been loved</td><td>αγαπήθηκαν</td><td>(they) were loved</td><td>αγαπιόνταν
(αγαπιόντουσαν)</td><td>(they) were being loved</td></tr>
</table>

The verb “to love” in Greek Passive Voice / Present Tenses				
	Ενεστώτας (Present)		**Παρακείμενος (Present Perfect)**	
Singular	αγαπ**ιέμαι**	(I) am being loved	έχω αγαπ**ηθεί**	(I) have been loved
	αγαπ**ιέσαι**	(you) are being loved	έχεις αγαπ**ηθεί**	(you) have been loved
	αγαπ**ιέται**	(he) is being loved	έχει αγαπ**ηθεί**	(he) has been loved
Plural	αγαπ**ιόμαστε**	(we) are being loved	έχουμε αγαπ**ηθεί**	(we) have been loved
	αγαπ**ιόσαστε**	(you) are being loved	έχετε αγαπ**ηθεί**	(you) have been loved
	αγαπ**ιούνται**	(they) are being loved	έχουν αγαπ**ηθεί**	(they) have been loved

The verb “to love” in Greek Passive Voice / Future Tenses						
	Εξακολουθητικός Μέλλοντας (Future Continuous)		**Στιγμιαίος Μέλλοντας (Simple Future)**		**Συντελεσμένος Μέλλοντας (Future Perfect/Imperfect)**	
Singular	θα αγαπ**ιέμαι**	(I) will be loved	θα αγαπ**ηθώ**	(I) will love	θα έχω αγαπ**ηθεί**	(I) will have been loved
	θα αγαπ**ιέσαι**	(you) will be loved	θα αγαπ**ηθείς**	(you) will love	θα έχεις αγαπ**ηθεί**	(you) will have been loved
	θα αγαπ**ιέται**	(he) will be loved	θα αγαπ**ηθεί**	(he) will love	θα έχει αγαπ**ηθεί**	(he) will have been loved
Plural	θα αγαπ**ιόμαστε**	(we)will loved	θα αγαπ**ηθούμε**	(we) will love	θα έχουμε αγαπ**ηθεί**	(we) will have been loved
	θα αγαπ**ιόσαστε**	(you)will be loved	θα αγαπη**θείτε**	(you) will love	θα έχετε αγαπ**ηθεί**	(you) will have been loved
	θα αγαπ**ιούνται**	(they) will be loved	θα αγαπ**ηθούν**	(they) will love	θα έχουν αγαπ**ηθεί**	(they) will have been loved

Appendix B: Evaluation of Modified Algorithm Testing Output

Number of words: 562242

Number of Stems: 138629

Frake's statistics:

Mean number of words per conflation class : 4.05573148475

Index compression factor : 0.75343535346

Unchanged Words : 7309 ------> ratio 0.0130720262642 (unchanged words /total words)

Mean Modified Humming Distance 2.44133602915

Median Modified Humming Distance: 2

Evaluation Output (part) The number next to every word notes the modified Hamming distance according to Frakes from its stem			
---------ΑΒΑΘ------- is a stem for 12 words	---------ΚΟΠΑΝΙΖ------- is a stem for 20 words	---------ΠΡΟΠΩΛ------- is a stem for 31 words	---------ΠΟΥΛ------- is a stem for 50 words
ΑΒΑΘΩΝ 2	ΚΟΠΑΝΙΖΩ 1	ΠΡΟΠΩΛΩΝΤΑΣ 5	
ΑΒΑΘΟΥΣ 3	ΚΟΠΑΝΙΖΟΥΝ 3	ΠΡΟΠΩΛΩ 1	ΠΟΥΛΩΝΤΑΣ 5
ΑΒΑΘΟΥ 2l	ΚΟΠΑΝΙΖΟΥΜΕ 4	ΠΡΟΠΩΛΟΥΣΕΣ 5	ΠΟΥΛΩ 1
ΑΒΑΘΟΣ 2	ΚΟΠΑΝΙΖΟΣΟΥΝ 5	ΠΡΟΠΩΛΟΥΣΕ 4	ΠΟΥΛΟΥΣΕΣ 5
ΑΒΑΘΟΙ 2	ΚΟΠΑΝΙΖΟΣΑΣΤΕ 6	ΠΡΟΠΩΛΟΥΣΑΤΕ 6	ΠΟΥΛΟΥΣΕ 4
ΑΒΑΘΟ 1	ΚΟΠΑΝΙΖΟΝΤΑΣ 5	ΠΡΟΠΩΛΟΥΣΑΜΕ 6	ΠΟΥΛΟΥΣΑΤΕ 6
ΑΒΑΘΗΣ 2	ΚΟΠΑΝΙΖΟΝΤΑΙ 5	ΠΡΟΠΩΛΟΥΣΑ 4	ΠΟΥΛΟΥΣΑΜΕ 6
ΑΒΑΘΗ 1	ΚΟΠΑΝΙΖΟΜΑΣΤΕ 6	ΠΡΟΠΩΛΟΥΝΤΑΙ 6	ΠΟΥΛΟΥΣΑ 4
ΑΒΑΘΕΣ 2	ΚΟΠΑΝΙΖΟΜΑΙ 4	ΠΡΟΠΩΛΟΥΝ 3	ΠΟΥΛΟΥΝ 3
ΑΒΑΘΕΙΣ 3	ΚΟΠΑΝΙΖΕΤΕ 3	ΠΡΟΠΩΛΟΥΜΕ 4	ΠΟΥΛΟΥΜΕ 4
ΑΒΑΘΕ 1	ΚΟΠΑΝΙΖΕΤΑΙ 4	ΠΡΟΠΩΛΟΥΜΑΣΤΕ 7	ΠΟΥΛΙΩΝ 3
ΑΒΑΘΑ 1	ΚΟΠΑΝΙΖΕΣΤΕ 4	ΠΡΟΠΩΛΟΥΜΑΙ 5	ΠΟΥΛΙΟΥΝΤΑΙ 7
	ΚΟΠΑΝΙΖΕΣΑΙ 4	ΠΡΟΠΩΛΗΣΩ 3	ΠΟΥΛΙΟΥ 3
---------ΑΒΑΘΟΥΛΩΤ-------	ΚΟΠΑΝΙΖΕΣ 2	ΠΡΟΠΩΛΗΣΤΕ 4	ΠΟΥΛΙΟΝΤΑΙ 6
is a stem for 11 words	ΚΟΠΑΝΙΖΕΙΣ 3	ΠΡΟΠΩΛΗΣΟΥ 4	ΠΟΥΛΙΕΤΑΙ 5
	ΚΟΠΑΝΙΖΕΙ 2	ΠΡΟΠΩΛΗΣΕΤΕ 5	ΠΟΥΛΙΕΣΤΕ 5
	ΚΟΠΑΝΙΖΕ 1	ΠΡΟΠΩΛΗΣΕ 3	ΠΟΥΛΙΕΣΑΙ 5
	ΚΟΠΑΝΙΖΑΝ 2	ΠΡΟΠΩΛΗΣΑΤΕ 5	ΠΟΥΛΙΕΣ 3
	ΚΟΠΑΝΙΖΑΜΕ 3	ΠΡΟΠΩΛΗΣΑΜΕ 5	ΠΟΥΛΙΕΜΑΙ 5

ΑΒΑΘΟΥΛΩΤΩΝ 2
ΑΒΑΘΟΥΛΩΤΟΥΣ 3
ΑΒΑΘΟΥΛΩΤΟΥ 2
ΑΒΑΘΟΥΛΩΤΟΣ 2
ΑΒΑΘΟΥΛΩΤΟΙ 2
ΑΒΑΘΟΥΛΩΤΟ 1
ΑΒΑΘΟΥΛΩΤΗΣ 2
ΑΒΑΘΟΥΛΩΤΗ 1
ΑΒΑΘΟΥΛΩΤΕΣ 2
ΑΒΑΘΟΥΛΩΤΕ 1
ΑΒΑΘΟΥΛΩΤΑ 1

---------ΑΒΑΘΜΟΛΟΓΗΤ-------
is a stem for 11 words

ΑΒΑΘΜΟΛΟΓΗΤΩΝ 2
ΑΒΑΘΜΟΛΟΓΗΤΟΥΣ 3
ΑΒΑΘΜΟΛΟΓΗΤΟΥ 2
ΑΒΑΘΜΟΛΟΓΗΤΟΣ 2
ΑΒΑΘΜΟΛΟΓΗΤΟΙ 2
ΑΒΑΘΜΟΛΟΓΗΤΟ 1
ΑΒΑΘΜΟΛΟΓΗΤΗΣ 2
ΑΒΑΘΜΟΛΟΓΗΤΗ 1
ΑΒΑΘΜΟΛΟΓΗΤΕΣ 2
ΑΒΑΘΜΟΛΟΓΗΤΕ 1
ΑΒΑΘΜΟΛΟΓΗΤΑ 1

---------ΑΒΑΘΜΙΔΩΤ-------
is a stem for 11 words

ΑΒΑΘΜΙΔΩΤΩΝ 2
ΑΒΑΘΜΙΔΩΤΟΥΣ 3
ΑΒΑΘΜΙΔΩΤΟΥ 2
ΑΒΑΘΜΙΔΩΤΟΣ 2
ΑΒΑΘΜΙΔΩΤΟΙ 2
ΑΒΑΘΜΙΔΩΤΟ 1
ΑΒΑΘΜΙΔΩΤΗΣ 2
ΑΒΑΘΜΙΔΩΤΗ 1
ΑΒΑΘΜΙΔΩΤΕΣ 2
ΑΒΑΘΜΙΔΩΤΕ 1
ΑΒΑΘΜΙΔΩΤΑ 1

ΚΟΠΑΝΙΖΑ 1

---------ΚΟΠΑΝΙΖΟΤ-------
is a stem for 1 words

ΚΟΠΑΝΙΖΟΤΑΝ 2

---------ΚΟΠΑΝΙΖΟΣΑΣΤ-------
is a stem for 1 words

ΚΟΠΑΝΙΖΟΣΑΣΤΑΝ 2

---------ΚΟΠΑΝΙΖΟΝΤΟΥΣ-------
is a stem for 1 words

ΚΟΠΑΝΙΖΟΝΤΟΥΣΑΝ 2

---------ΚΟΠΑΝΙΖΟΝΤ-------
is a stem for 1 words

ΚΟΠΑΝΙΖΟΝΤΑΝ 2

---------ΚΟΠΑΝΙΖΟΜ-------
is a stem for 1 words

ΚΟΠΑΝΙΖΟΜΟΥΝ 3

---------ΚΟΠΑΝΙΖΟΜΑΣΤ-------
is a stem for 1 words

ΚΟΠΑΝΙΖΟΜΑΣΤΑΝ 2

---------ΚΟΠΑΝΙΖΑΤ-------
is a stem for 1 words

ΚΟΠΑΝΙΖΑΤΕ 1

---------ΚΟΠΑΝΗΣ-------
is a stem for 6 words

ΚΟΠΑΝΗΣΟΥΝ 3
ΚΟΠΑΝΗΣΟΥΜΕ 4
ΚΟΠΑΝΗΣΕΣ 2
ΚΟΠΑΝΗΣΕΙΣ 3
ΚΟΠΑΝΗΣΕΙ 2
ΚΟΠΑΝΗΣΑΝ 2

ΠΡΟΠΩΛΗΣΑ 3
ΠΡΟΠΩΛΗΜΕΝΟΣ 6
ΠΡΟΠΩΛΗΜΕΝΟ 5
ΠΡΟΠΩΛΗΜΕΝΗ 5
ΠΡΟΠΩΛΗΜΕΝΕ 5
ΠΡΟΠΩΛΗΘΩ 3
ΠΡΟΠΩΛΗΘΗΚΕΣ 6
ΠΡΟΠΩΛΗΘΗΚΕ 5
ΠΡΟΠΩΛΗΘΗΚΑ 5
ΠΡΟΠΩΛΕΙΤΕ 4
ΠΡΟΠΩΛΕΙΣ 3
ΠΡΟΠΩΛΕΙ 2

---------ΠΡΟΠΩΛΟΥΤ-------
is a stem for 1 words

ΠΡΟΠΩΛΟΥΤΑΝ 2

---------ΠΡΟΠΩΛΟΥΣ-------
is a stem for 2 words

ΠΡΟΠΩΛΟΥΣΟΥΝ 3
ΠΡΟΠΩΛΟΥΣΑΝ 2

ΠΡΟΠΩΛΟΥΣΑΣΤ-------
is a stem for 1 words

ΠΡΟΠΩΛΟΥΣΑΣΤΑΝ 2

---------ΠΡΟΠΩΛΟΥΝΤ-------
is a stem for 1 words

ΠΡΟΠΩΛΟΥΝΤΑΝ 2

ΠΡΟΠΩΛΟΥΜΑΣΤ-------
is a stem for 1 words

ΠΡΟΠΩΛΟΥΜΑΣΤΑΝ 2

---------ΠΡΟΠΩΛΗΣ-------
is a stem for 10 words

ΠΡΟΠΩΛΗΣΟΥΝ 3
ΠΡΟΠΩΛΗΣΟΥΜΕ 4
ΠΡΟΠΩΛΗΣΗΣ 2
ΠΡΟΠΩΛΗΣΗ 1
ΠΡΟΠΩΛΗΣΕΩΣ 3
ΠΡΟΠΩΛΗΣΕΩΝ 3

ΠΟΥΛΙΑΣ 3
ΠΟΥΛΙΑ 2
ΠΟΥΛΙ 1
ΠΟΥΛΗΣΩ 3
ΠΟΥΛΗΣΤΕ 4
ΠΟΥΛΗΣΟΥ 4
ΠΟΥΛΗΣΕΤΕ 5
ΠΟΥΛΗΣΕ 3
ΠΟΥΛΗΣΑΤΕ 5
ΠΟΥΛΗΣΑΜΕ 5
ΠΟΥΛΗΣΑ 3
ΠΟΥΛΗΜΕΝΟΣ 6
ΠΟΥΛΗΜΕΝΟ 5
ΠΟΥΛΗΜΕΝΗ 5
ΠΟΥΛΗΜΕΝΕ 5
ΠΟΥΛΗΘΩ 3
ΠΟΥΛΗΘΗΚΕΣ 6
ΠΟΥΛΗΘΗΚΕ 5
ΠΟΥΛΗΘΗΚΑ 5
ΠΟΥΛΑΩ 2
ΠΟΥΛΑΣ 2
ΠΟΥΛΑΝΕ 3
ΠΟΥΛΑΝ 2
ΠΟΥΛΑΜΕ 3
ΠΟΥΛΑΕΙ 3
ΠΟΥΛΑΓΕΣ 4
ΠΟΥΛΑΓΕ 3
ΠΟΥΛΑΓΑΤΕ 5
ΠΟΥΛΑΓΑΝΕ 5
ΠΟΥΛΑΓΑΜΕ 5
ΠΟΥΛΑΓΑ 3
ΠΟΥΛΑ 1

ΠΟΥΛΧΕΡΙΑΡ-------
is a stem for 1 words

ΠΟΥΛΧΕΡΙΑΡ 0

ΠΟΥΛΧΕΡ-------
is a stem for 1 words

ΠΟΥΛΧΕΡΙΑ 2

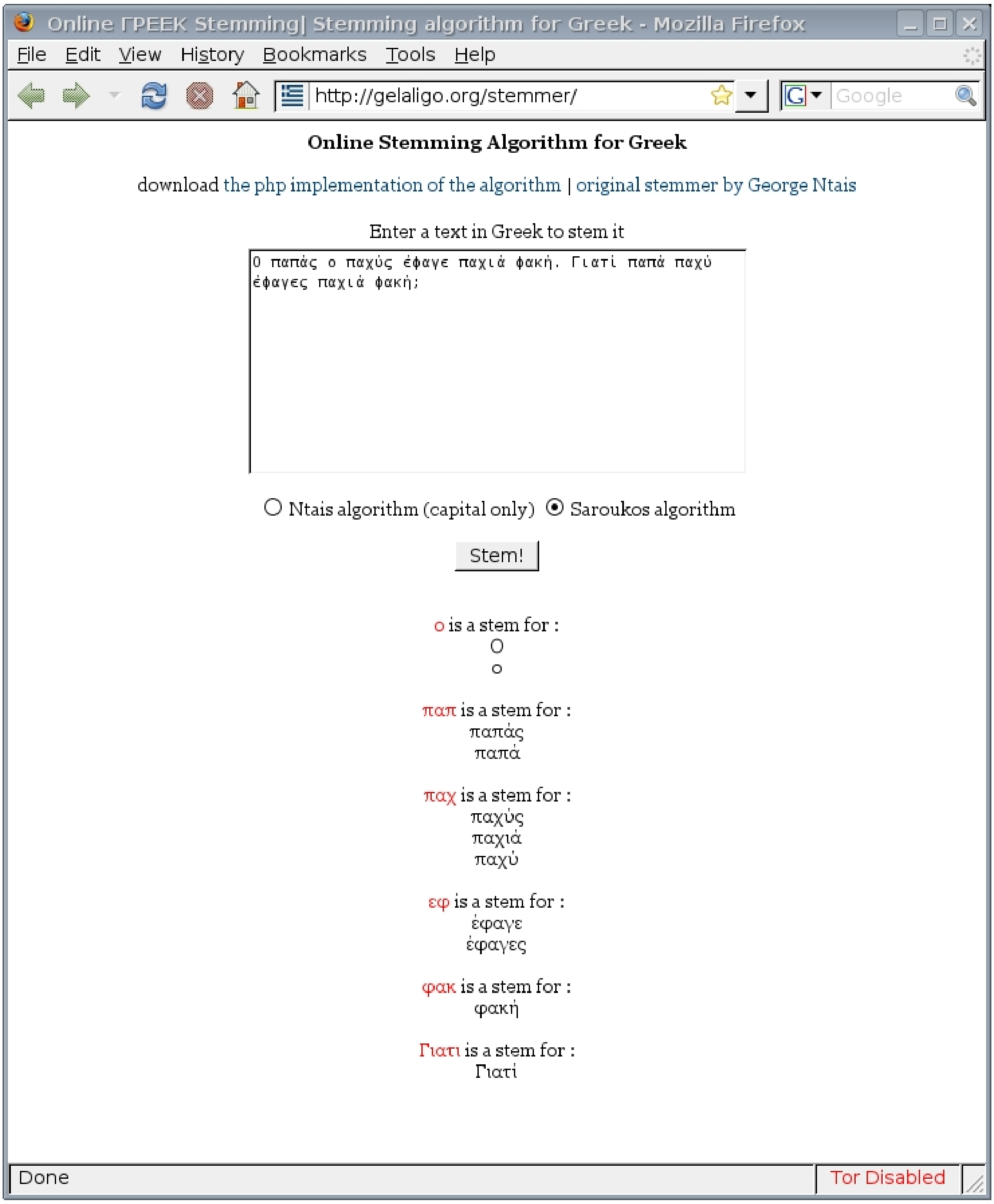
ΠΟΥΛΟΥΣ-------
is a stem for 1 words

ΠΟΥΛΟΥΣΑΝ 2

---------ΑΒΑΕΙ------- is a stem for 1 words ΑΒΑΕΙΟ 1		ΠΡΟΠΩΛΗΣΕΣ 2 ΠΡΟΠΩΛΗΣΕΙΣ 3 ΠΡΟΠΩΛΗΣΕΙ 2 ΠΡΟΠΩΛΗΣΑΝ 2	--------- ΠΟΥΛΟΛΟΓ------- is a stem for 1 words ΠΟΥΛΟΛΟΓΟΣ 2 --------- ΠΟΥΛΟΒΕΡΑΚ------ - is a stem for 2 words ΠΟΥΛΟΒΕΡΑΚΙΑ 2 ΠΟΥΛΟΒΕΡΑΚΙ 1 --------- ΠΟΥΛΟΒΕΡ------- is a stem for 1 words ΠΟΥΛΟΒΕΡ 0

The previous table is a small sample from an extensive list of conflation classes created by the modified stemmer. In this small list we included examples of both understemming and overstemming. One overstemming example is the conflation class of “ΠΟΥΛ” which includes words from the word “πουλί” (bird) and “πουλάω” (to sell). On the contrary, understemming examples have to do with the grammatical changes that words undergo in past tenses.

Appendix B: On-line Stemmer



Our implementation of the algorithm is freely available at http://gelaligo.org/stemmer under an LGPL licence, along with an on-line demo.

Appendix C: Stop Word List

ΑΒΑ
ΑΓΑ
ΑΓΗ
ΑΓΩ
ΑΔΗ
ΑΔΩ
ΑΕ
ΑΕΙ
ΑΘΩ
ΑΙ
ΑΙΚ
ΑΚΗ
ΑΚΟΜΑ
ΑΚΟΜΗ
ΑΚΡΙΒΩΣ
ΑΛΑ
ΑΛΗΘΕΙΑ
ΑΛΗΘΙΝΑ
ΑΛΛΑΧΟΥ
ΑΛΛΙΩΣ
ΑΛΛΙΩΤΙΚΑ
ΑΛΛΟΙΩΣ
ΑΛΛΟΙΩΤΙΚΑ
ΑΛΛΟΤΕ
ΑΛΤ
ΑΛΩ
ΑΜΑ
ΑΜΕ
ΑΜΕΣΑ
ΑΜΕΣΩΣ
ΑΜΩ
ΑΝ
ΑΝΑ
ΑΝΑΜΕΣΑ
ΑΝΑΜΕΤΑΞΥ
ΑΝΕΥ
ΑΝΤΙ
ΑΝΤΙΠΕΡΑ
ΑΝΤΙΣ
ΑΝΩ
ΑΝΩΤΕΡΩ
ΑΞΑΦΝΑ
ΑΠ
ΑΠΕΝΑΝΤΙ
ΑΠΟ
ΑΠΟΨΕ
ΑΠΩ
ΑΡΑ
ΑΡΑΓΕ
ΑΡΕ
ΑΡΚ
ΑΡΚΕΤΑ
ΑΡΛ
ΑΡΜ
ΑΡΤ
ΑΡΥ
ΑΡΩ
ΑΣ
ΑΣΑ
ΑΣΟ
ΑΤΑ
ΑΤΕ
ΑΤΗ
ΑΤΙ
ΑΤΜ
ΑΤΟ
ΑΥΡΙΟ
ΑΦΗ
ΑΦΟΤΟΥ
ΑΦΟΥ
ΑΧ
ΑΧΕ
ΑΧΟ
ΑΨΑ
ΑΨΕ
ΑΨΗ
ΑΨΥ
ΑΩΕ
ΑΩΟ
ΒΑΝ
ΒΑΤ
ΒΑΧ
ΒΕΑ
ΒΕΒΑΙΟΤΑΤΑ
ΒΗΞ
ΒΙΑ
ΒΙΕ
ΒΙΗ
ΒΙΟ
ΒΟΗ
ΒΟΩ
ΒΡΕ
ΓΑ
ΓΑΒ
ΓΑΡ
ΓΕΝ
ΓΕΣ
ΓΗ
ΓΗΝ

ΓΙ
ΓΙΑ
ΓΙΕ
ΓΙΝ
ΓΙΟ
ΓΚΙ
ΓΚΥ
ΓΟΗ
ΓΟΟ
ΓΡΗΓΟΡΑ
ΓΡΙ
ΓΡΥ
ΓΥΗ
ΓΥΡΩ
ΔΑ
ΔΕ
ΔΕΗ
ΔΕΙ
ΔΕΝ
ΔΕΣ
ΔΗ
ΔΗΘΕΝ
ΔΗΛΑΔΗ
ΔΗΩ
ΔΙ
ΔΙΑ
ΔΙΑΡΚΩΣ
ΔΙΟΛΟΥ
ΔΙΣ
ΔΙΧΩΣ
ΔΟΛ
ΔΟΝ
ΔΡΑ
ΔΡΥ
ΔΡΧ
ΔΥΕ
ΔΥΟ
ΔΩ
ΕΑΜ
ΕΑΝ
ΕΑΡ
ΕΘΗ
ΕΙ
ΕΙΔΕΜΗ
ΕΙΘΕ
ΕΙΜΑΙ
ΕΙΜΑΣΤΕ
ΕΙΝΑΙ
ΕΙΣ
ΕΙΣΑΙ
ΕΙΣΑΣΤΕ
ΕΙΣΤΕ
ΕΙΤΕ
ΕΙΧΑ
ΕΙΧΑΜΕ
ΕΙΧΑΝ
ΕΙΧΑΤΕ
ΕΙΧΕ
ΕΙΧΕΣ
ΕΚ
ΕΚΟ
ΕΚΕΙ
ΕΛΑ
ΕΛΙ
ΕΜΠ
ΕΝ
ΕΝΤΕΛΩΣ
ΕΝΤΟΣ
ΕΝΤΩΜΕΤΑΞΥ
ΕΝΩ
ΕΞ
ΕΞΑΦΝΑ
ΕΞΙ
ΕΞΙΣΟΥ
ΕΞΩ
ΕΟΚ
ΕΠΑΝΩ
ΕΠΕΙΔΗ
ΕΠΕΙΤΑ
ΕΠΗ
ΕΠΙ
ΕΠΙΣΗΣ
ΕΠΟΜΕΝΩΣ
ΕΡΑ
ΕΣ
ΕΣΑΣ
ΕΣΕ
ΕΣΕΙΣ
ΕΣΕΝΑ
ΕΣΗ
ΕΣΤΩ
ΕΣΥ
ΕΣΩ
ΕΤΙ
ΕΤΣΙ
ΕΥ
ΕΥΑ
ΕΥΓΕ
ΕΥΘΥΣ
ΕΥΤΥΧΩΣ
ΕΦΕ
ΕΦΕΞΗΣ
ΕΦΤ
ΕΧΕ
ΕΧΕΙ

ΕΧΕΙΣ
ΕΧΕΤΕ
ΕΧΘΕΣ
ΕΧΟΜΕ
ΕΧΟΥΜΕ
ΕΧΟΥΝ
ΕΧΤΕΣ
ΕΧΩ
ΕΩΣ
ΖΕΑ
ΖΕΗ
ΖΕΙ
ΖΕΝ
ΖΗΝ
ΖΩ
Η
ΗΔΗ
ΗΔΥ
ΗΘΗ
ΗΛΟ
ΗΜΙ
ΗΠΑ
ΗΣΑΣΤΕ
ΗΣΟΥΝ
ΗΤΑ
ΗΤΑΝ
ΗΤΑΝΕ
ΗΤΟΙ
ΗΤΤΟΝ
ΗΩ
ΘΑ
ΘΥΕ
ΘΩΡ
Ι
ΙΑ
ΙΒΟ
ΙΔΗ
ΙΔΙΩΣ
ΙΕ
ΙΙ
ΙΙΙ
ΙΚΑ
ΙΛΟ
ΙΜΑ
ΙΝΑ
ΙΝΩ
ΙΞΕ
ΙΞΟ
ΙΟ
ΙΟΙ
ΙΣΑ
ΙΣΑΜΕ
ΙΣΕ
ΙΣΗ
ΙΣΙΑ
ΙΣΟ
ΙΣΩΣ
ΙΩΒ
ΙΩΝ
ΙΩΣ
Ιαν
ΚΑΘ
ΚΑΘΕ
ΚΑΘΕΤΙ
ΚΑΘΟΛΟΥ
ΚΑΘΩΣ
ΚΑΙ
ΚΑΝ
ΚΑΠΟΤΕ
ΚΑΠΟΥ
ΚΑΠΩΣ
ΚΑΤ
ΚΑΤΑ
ΚΑΤΙ
ΚΑΤΙΤΙ
ΚΑΤΟΠΙΝ
ΚΑΤΩ
ΚΑΩ
ΚΒΟ
ΚΕΑ
ΚΕΙ
ΚΕΝ
ΚΙ
ΚΙΜ
ΚΙΟΛΑΣ
ΚΙΤ
ΚΙΧ
ΚΚΕ
ΚΛΙΣΕ
ΚΛΠ
ΚΟΚ
ΚΟΝΤΑ
ΚΟΧ
ΚΤΛ
ΚΥΡ
ΚΥΡΙΩΣ
ΚΩ
ΚΩΝ
ΛΑ
ΛΕΑ
ΛΕΝ
ΛΕΟ
ΛΙΑ
ΛΙΓΑΚΙ
ΛΙΓΟ

ΛΙΓΩΤΕΡΟ
ΛΙΟ
ΛΙΡ
ΛΟΓΩ
ΛΟΙΠΑ
ΛΟΙΠΟΝ
ΛΟΣ
ΛΣ
ΛΥΩ
ΜΑ
ΜΑΖΙ
ΜΑΚΑΡΙ
ΜΑΛΙΣΤΑ
ΜΑΛΛΟΝ
ΜΑΝ
ΜΑΞ
ΜΑΣ
ΜΑΤ
ΜΕ
ΜΕΘΑΥΡΙΟ
ΜΕΙ
ΜΕΙΟΝ
ΜΕΛ
ΜΕΛΕΙ
ΜΕΛΛΕΤΑΙ
ΜΕΜΙΑΣ
ΜΕΝ
ΜΕΣ
ΜΕΣΑ
ΜΕΤ
ΜΕΤΑ
ΜΕΤΑΞΥ
ΜΕΧΡΙ
ΜΗ
ΜΗΔΕ
ΜΗΝ
ΜΗΠΩΣ
ΜΗΤΕ
ΜΙ
ΜΙΞ
ΜΙΣ
ΜΜΕ
ΜΝΑ
ΜΟΒ
ΜΟΛΙΣ
ΜΟΛΟΝΟΤΙ
ΜΟΝΑΧΑ
ΜΟΝΟΜΙΑΣ
ΜΟΥ
ΜΠΑ
ΜΠΟΡΕΙ
ΜΠΟΡΟΥΝ
ΜΠΡΑΒΟ
ΜΠΡΟΣ
ΜΠΩ
ΜΥ
ΜΥΑ
ΜΥΝ
ΝΑ
ΝΑΕ
ΝΑΙ
ΝΑΟ
ΝΔ
ΝΕΐ
ΝΙ
ΝΙΑ
ΝΙΚ
ΝΙΛ
ΝΙΝ
ΝΙΟ
ΝΤΑ
ΝΤΕ
ΝΤΙ
ΝΤΟ
ΝΥΝ
ΝΩΕ
ΝΩΡΙΣ
ΞΑΝΑ
ΞΑΦΝΙΚΑ
ΞΕΩ
ΞΙ
Ο
ΟΑ
ΟΑΠ
ΟΔΟ
ΟΕ
ΟΖΟ
ΟΗΕ
ΟΙ
ΟΙΑ
ΟΙΗ
ΟΚΑ
ΟΛΟΓΥΡΑ
ΟΛΟΝΕΝ
ΟΛΟΤΕΛΑ
ΟΛΩΣΔΙΟΛΟΥ
ΟΜΩΣ
ΟΝ
ΟΝΕ
ΟΝΟ
ΟΠΑ
ΟΠΕ
ΟΠΗ
ΟΠΟ
ΟΠΟΙΑΔΗΠΟΤΕ

ΟΠΟΙΑΝΔΗΠΟΤΕ
ΟΠΟΙΑΣΔΗΠΟΤΕ
ΟΠΟΙΔΗΠΟΤΕ
ΟΠΟΙΕΣΔΗΠΟΤΕ
ΟΠΟΙΟΔΗΠΟΤΕ
ΟΠΟΙΟΝΔΗΠΟΤΕ
ΟΠΟΙΟΣΔΗΠΟΤΕ
ΟΠΟΙΟΥΔΗΠΟΤΕ
ΟΠΟΙΟΥΣΔΗΠΟΤΕ
ΟΠΟΙΩΝΔΗΠΟΤΕ
ΟΠΟΤΕΔΗΠΟΤΕ
ΟΠΟΥ
ΟΠΟΥΔΗΠΟΤΕ
ΟΠΩΣ
ΟΡΑ
ΟΡΕ
ΟΡΗ
ΟΡΟ
ΟΡΦ
ΟΡΩ
ΟΣΑ
ΟΣΑΔΗΠΟΤΕ
ΟΣΕ
ΟΣΕΣΔΗΠΟΤΕ
ΟΣΗΔΗΠΟΤΕ
ΟΣΗΝΔΗΠΟΤΕ
ΟΣΗΣΔΗΠΟΤΕ
ΟΣΟΔΗΠΟΤΕ
ΟΣΟΙΔΗΠΟΤΕ
ΟΣΟΝΔΗΠΟΤΕ
ΟΣΟΣΔΗΠΟΤΕ
ΟΣΟΥΔΗΠΟΤΕ
ΟΣΟΥΣΔΗΠΟΤΕ
ΟΣΩΝΔΗΠΟΤΕ
ΟΤΑΝ
ΟΤΕ
ΟΤΙ
ΟΤΙΔΗΠΟΤΕ
ΟΥ
ΟΥΔΕ
ΟΥΚ
ΟΥΣ
ΟΥΤΕ
ΟΥΦ
ΟΧΙ
ΟΨΑ
ΟΨΕ
ΟΨΗ
ΟΨΙ
ΟΨΟ
ΠΑ
ΠΑΛΙ
ΠΑΝ
ΠΑΝΤΟΤΕ
ΠΑΝΤΟΥ
ΠΑΝΤΩΣ
ΠΑΠ
ΠΑΡ
ΠΑΡΑ
ΠΕΙ
ΠΕΡ
ΠΕΡΑ
ΠΕΡΙ
ΠΕΡΙΠΟΥ
ΠΕΡΣΙ
ΠΕΡΥΣΙ
ΠΕΣ
ΠΙ
ΠΙΑ
ΠΙΘΑΝΟΝ
ΠΙΚ
ΠΙΟ
ΠΙΣΩ
ΠΙΤ
ΠΙΩ
ΠΛΑΙ
ΠΛΕΟΝ
ΠΛΗΝ
ΠΛΩ
ΠΜ
ΠΟΑ
ΠΟΕ
ΠΟΛ
ΠΟΛΥ
ΠΟΠ
ΠΟΤΕ
ΠΟΥ
ΠΟΥΘΕ
ΠΟΥΘΕΝΑ
ΠΡΕΠΕΙ
ΠΡΙ
ΠΡΙΝ
ΠΡΟ
ΠΡΟΚΕΙΜΕΝΟΥ
ΠΡΟΚΕΙΤΑΙ
ΠΡΟΠΕΡΣΙ
ΠΡΟΣ
ΠΡΟΤΟΥ
ΠΡΟΧΘΕΣ
ΠΡΟΧΤΕΣ
ΠΡΩΤΥΤΕΡΑ
ΠΥΑ
ΠΥΞ
ΠΥΟ
ΠΥΡ

ΠΧ
ΠΩ
ΠΩΛ
ΠΩΣ
ΡΑ
ΡΑΙ
ΡΑΠ
ΡΑΣ
ΡΕ
ΡΕΑ
ΡΕΕ
ΡΕΙ
ΡΗΣ
ΡΘΩ
ΡΙΟ
ΡΟ
ΡΟΐ
ΡΟΕ
ΡΟΖ
ΡΟΗ
ΡΟΘ
ΡΟΙ
ΡΟΚ
ΡΟΛ
ΡΟΝ
ΡΟΣ
ΡΟΥ
ΣΑΙ
ΣΑΝ
ΣΑΟ
ΣΑΣ
ΣΕ
ΣΕΙΣ
ΣΕΚ
ΣΕΞ
ΣΕΡ
ΣΕΤ
ΣΕΦ
ΣΗΜΕΡΑ
ΣΙ
ΣΙΑ
ΣΙΓΑ
ΣΙΚ
ΣΙΧ
ΣΚΙ
ΣΟΙ
ΣΟΚ
ΣΟΛ
ΣΟΝ
ΣΟΣ
ΣΟΥ
ΣΡΙ
ΣΤΑ
ΣΤΗ
ΣΤΗΝ
ΣΤΗΣ
ΣΤΙΣ
ΣΤΟ
ΣΤΟΝ
ΣΤΟΥ
ΣΤΟΥΣ
ΣΤΩΝ
ΣΥ
ΣΥΓΧΡΟΝΩΣ
ΣΥΝ
ΣΥΝΑΜΑ
ΣΥΝΕΠΩΣ
ΣΥΝΗΘΩΣ
ΣΧΕΔΟΝ
ΣΩΣΤΑ
ΤΑ
ΤΑΔΕ
ΤΑΚ
ΤΑΝ
ΤΑΟ
ΤΑΥ
ΤΑΧΑ
ΤΑΧΑΤΕ
ΤΕ
ΤΕΙ
ΤΕΛ
ΤΕΛΙΚΑ
ΤΕΛΙΚΩΣ
ΤΕΣ
ΤΕΤ
ΤΖΟ
ΤΗ
ΤΗΛ
ΤΗΝ
ΤΗΣ
ΤΙ
ΤΙΚ
ΤΙΜ
ΤΙΠΟΤΑ
ΤΙΠΟΤΕ
ΤΙΣ
ΤΝΤ
ΤΟ
ΤΟΙ
ΤΟΚ
ΤΟΜ
ΤΟΝ
ΤΟΠ
ΤΟΣ
ΤΟΣΩΝ

ΤΟΣΑ
ΤΟΣΕΣ
ΤΟΣΗ
ΤΟΣΗΝ
ΤΟΣΗΣ
ΤΟΣΟ
ΤΟΣΟΙ
ΤΟΣΟΝ
ΤΟΣΟΣ
ΤΟΣΟΥ
ΤΟΣΟΥΣ
ΤΟΤΕ
ΤΟΥ
ΤΟΥΛΑΧΙΣΤΟ
ΤΟΥΛΑΧΙΣΤΟΝ
ΤΟΥΣ
ΤΣ
ΤΣΑ
ΤΣΕ
ΤΥΧΟΝ
ΤΩ
ΤΩΝ
ΤΩΡΑ
ΥΑΣ
ΥΒΑ
ΥΒΟ
ΥΙΕ
ΥΙΟ
ΥΛΑ
ΥΛΗ
ΥΝΙ
ΥΠ
ΥΠΕΡ
ΥΠΟ
ΥΠΟΨΗ
ΥΠΟΨΙΝ
ΥΣΤΕΡΑ
ΥΦΗ
ΥΨΗ
ΦΑ
ΦΑΐ
ΦΑΕ
ΦΑΝ
ΦΑΞ
ΦΑΣ
ΦΑΩ
ΦΕΖ
ΦΕΙ
ΦΕΤΟΣ
ΦΕΥ
ΦΙ
ΦΙΛ
ΦΙΣ
ΦΟΞ
ΦΠΑ
ΦΡΙ
ΧΑ
ΧΑΗ
ΧΑΛ
ΧΑΝ
ΧΑΦ
ΧΕ
ΧΕΙ
ΧΘΕΣ
ΧΙ
ΧΙΑ
ΧΙΛ
ΧΙΟ
ΧΛΜ
ΧΜ
ΧΟΗ
ΧΟΛ
ΧΡΩ
ΧΤΕΣ
ΧΩΡΙΣ
ΧΩΡΙΣΤΑ
ΨΕΣ
ΨΗΛΑ
ΨΙ
ΨΙΤ
Ω
ΩΑ
ΩΑΣ
ΩΔΕ
ΩΕΣ
ΩΘΩ
ΩΜΑ
ΩΜΕ
ΩΝ
ΩΟ
ΩΟΝ
ΩΟΥ
ΩΣ
ΩΣΑΝ
ΩΣΗ
ΩΣΟΤΟΥ
ΩΣΠΟΥ
ΩΣΤΕ
ΩΣΤΟΣΟ
ΩΤΑ
ΩΧ
ΩΩΝ
ΓΙΑΤΙ