# Using a WWW Search Engine to Evaluate Normalization Performance for a Highly Inflectional Language

**Alexandros Ntoulas**
Computer Engineering & Informatics Department
Patras University
26500 Rion, Greece
ntoulas@cti.gr

**Sofia Stamou**
Computer Engineering & Informatics Department
Patras University
26500 Rion, Greece
stamou@cti.gr

**Manolis Tzagarakis**
Computer Engineering & Informatics Department
Patras University
26500 Rion, Greece
tzagara@cti.gr

## Abstract

Incorporation of language techniques in various web search engines has shown both positive and negative effects on retrieval performance. This paper describes an experiment conducted to determine the impact document and query normalization has on retrieval performance of a Greek web search engine. Experiments especially focus on the measurement of Recall and Precision of the obtained results and show whether information retrieval can benefit from normalization techniques. Results show that normalization yields significant improvement concerning recall of the retrieved data, whereas precision strongly depends on the queries issued to the engine.

## 1 Introduction

Language techniques are widely used in Information Retrieval (IR) applications aiming at a better representation of text documents for indexing purposes and at a further improvement of retrieval performance. A conventional approach aiming at improving the query-document matching process is achieved by reducing morphological variants of wordforms to their common roots by stemming. However, stemmers are suited for morphologically poor languages like English (Tzoukermann 1997) and their contribution to the retrieval performance is frequently reported as statistically insignificant (Hull 1996) (Harman 1991).

Another technique employed to improve performance is normalization of document and query terms. Normalization is an operation that provides a unique and identical representative for all wordforms representing the same salient concept. Identification of wordform variants is essential for retrieving texts written in highly inflectional languages, since reduction of a term's declinable wordforms to their respective first inflected form[1] enhances retrieval performance. Otherwise, many different terms are used to represent a concept expressed by wordform variants, resulting in disappointingly low recall (Krovetz 1993). This paper addresses the application of normalization techniques on a commercial Greek web search engine and examines the retrieval performance considering both recall and precision of the search results.

To determine the utility of normalization in retrieving information from the Web we conducted a series of experiments by using two different sets of queries issued in both normalized and un-normalized form. The first set of queries was extracted from real reference questions submitted by the users of the search engine and the other set consisted of manually constructed search requests submitted by the authors. Our aim was to compare the final results obtained for each query set and reach to some early conclusions on whether normalization improves retrieval performance in terms of Recall and Precision. Thus, the query sets were deliberately imposed twice in the engine, with and without the adoption of normalization respectively. We evaluated our results using Recall and Relevance figures

[1] For Greek the first inflected form for nouns is the nominative singular and for verbs the indicative of the present tense in the active voice.

provided by the engine along with Relevance figures provided by a small group of end users. In the following section we discuss previous research and we continue with a description of the normalizer (3) and how it was embodied into the search engine (4). Following on from this, a detailed description of the criteria applied for the selection and composition of the sample queries is provided (5.1). In the remaining sections we present the actual implementation of the experiments carried out (5.2) along with a global evaluation of the search results (6). We conclude with an overall assessment (7) on how normalization affects IR performance for Greek.

## 2 Background

The goal of an IR system is to locate relevant documents in response to a user's query (Krovetz 1992). Towards the acquisition of more relevant data, many approaches have been proposed and incorporated in various IR systems, most of which concentrate on the usage of advanced NLP methods. A widely used technique to improve retrieval performance is the direct extraction of words as they appear in the input text followed by the reduction of variant wordforms to common roots by stemming. Stemmers conflate word variants by truncating their endings. Suffix stripping (Porter 1980) is the simplest technique proposed to achieve this goal by using a list of frequent suffixes to reduce words to their baseform or stem. Stemming has been studied mainly for English, but there is evidence that it is useful for other languages as well (Xu 1998). Many experiments have been conducted trying to estimate its effect on retrieval performance and there is a lot of variation and inconsistencies in the results obtained, since quite a lot of factors seem to be of importance, e.g. language, evaluation measures etc. (Kraaij 1996). For highly inflectional languages, however, accurate identification of wordform variations by stemming is impossible due to their inflectional or derivational morphology (Krovetz 1993).

Popovic and Willet (1992) after investigating whether suffix stripping would be effective for a morphologically more complex language (like Slovene) concluded that the effectiveness of stemming is determined by the morphological complexity of a language. Even Inflectional Stemming, the most successful simple linguistic stemming method, improves recall without significant loss in precision, while removing derivational morphology despite its usefulness, it generally reduces Precision too much (Kraaij 1996). For the Greek language in particular, a difference in form might correspond to a difference in meaning since a morphological root (stem) might be common for words with a different part of speech or with a different sense.

The default assumption that a difference in form is associated with a difference in meaning unless there is strong evidence that the wordforms are related was adopted by Krovetz (1997) in experiments he conducted when trying to estimate the effect distinct word meanings have on retrieval performance. The outcome of the experiments was that conflating inflectional variants harmed the performance of about a third of the queries. Taking into consideration the research summarized above and by focusing on the complexity of the Greek inflectional system our intention was to examine the effect of normalization on retrieval performance instead of applying the aforementioned language techniques. Normalization is a stemming-like process that provides a unique and identical representative (e.g. a string or a number) for all wordforms of the same lemma. This operation is called morphological normalization or lemmatization and can be assisted by lemma lexicons (Strzalkowski 1997) or is entirely based on full lexicons that provide a grouped or clustered organization of wordforms of the same lemma. In the remaining sections we describe the normalizer incorporated into the search engine and we evaluate the experiments conducted in order to assess its impact on retrieval effectiveness over the web.

## 3 Lexicon-Based Normalization

Morphology is the area of linguistics concerned with the internal structure of words and is usually broken down into two subclasses: inflectional and derivational. For highly inflectional languages, the different forms of a word might correspond to a difference in meaning and in some cases a basic form of one word might be an inflected form of another word (Pirkola 1998). The complexity of the Greek inflectional system disallows the

application of automatic means for the extraction of lexical knowledge (Orphanos 1999) thus, our approach concentrates on using normalization for the acquisition of lexical information from the web. Word normalization is an operation that provides a unique and identical representative for inflectional and derivational variants of wordforms representing the same concept, by reducing alternative formulations of wordforms to a normalized form relying on the usage of linguistic knowledge, encoded in computational lexicons.

One of the primary motivations for using a normalizer is that it plays an important role in resolving lexical ambiguity (Arampatzis 2000), since we are no longer dealing with stems, but with words. For a morphologically agglutinative language, like Turkish a rule-based morphological disambiguation approach achieved precision of 93 to 94% (Oflazer 1996). Thus, we incorporated a normalizer into the search engine instead of a stemmer so that IR is conducted by words rather than truncated wordforms. Our hypothesis is that retrieval performance for Greek can be improved by indexing documents not by morphological roots (stems) of words but by their respective normalized forms. The normalizer we embodied in our search engine consists of the following components: a morphological lexicon, a tokenizer and a Part-Of-Speech (POS) Disambiguator (Figure 1).

The ***Morphological Lexicon*** we incorporated into the normalizer contains ~90,000 lemmas along with their morphosyntactic attributes, which can take values such as 'noun', 'singular', 'present' etc. The lexicon facilitates the normalization process by providing morphological attributes for all declinable wordforms of a lemma. Raw text passes through the ***Tokenizer***, where it is converted to a stream of tokens. Non-word tokens (e.g. punctuation marks, numbers etc.) are resolved and receive a tag according to their category. Word-tokens are looked up in the morphological lexicon and those found receive one or more tags. Words with more than one POS tag and those not found in the lexicon pass through the ***Disambiguator***, where the contextually appropriate tag is decided or guessed through the traversal of corpus-based decision trees (Orphanos 1999), using the dictionary files.

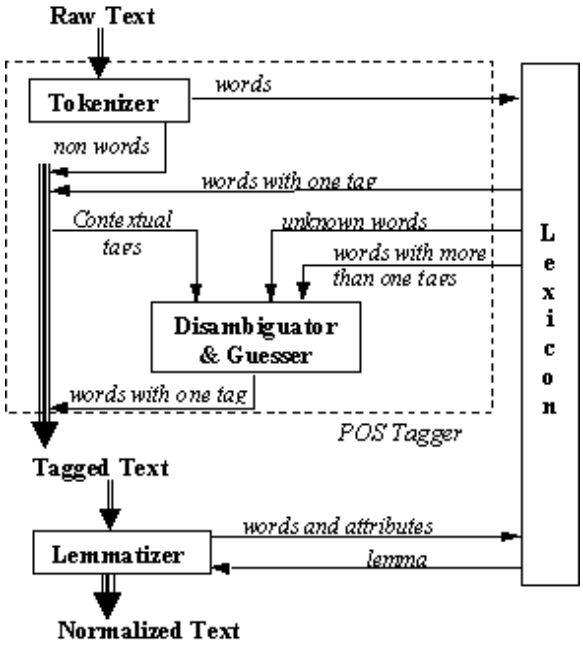


***Figure 1****.Architecture of the Normalizer*

## 4 Incorporating the Normalizer into the Search Engine

The search engine[2] we used indexes the full text of 700,000 Web pages with continuous update frequencies. It supports wildcards, Boolean searching, term as well as phrase searching, field searching(e.g.title:governmental, url:home.html, keywords) and case insensitive searching. The engine provides two modes: plain and normalized, thus two different indices are kept, each corresponding to a mode. The plain version indexes the pages fetched by the spider (including stop words), while the normalized one passes the pages through the normalizer, where stop words are excluded and the remaining tokens are induced to their first inflected form. Since the engine is case insensitive all tokens prior to indexing are turned to lower case[3]. For each search request end users can either use the default (plain) or the normalization mode of the engine. Whenever the normalization is adopted the query is normalized in order to match with terms from the normalized index. In both cases the query is converted to lower case so that the query terms can be matched towards the respective index.

[2] http://www.anazitisis.gr

[3] Lowercase folding was chosen, instead of Uppercase because of our past experience with stressed symbols of the Greek alphabet, over the Web (e.g.stressed A→ Ά doesn't show up correctly).

The display order of search results is determined by the engine from the location of matching words and occurrence of their frequencies.

## 5 Preliminary Experiments

To investigate the effect normalization has on retrieval performance in a highly inflectional language (Greek) we conducted a series of preliminary experiments by issuing twice to the search engine a set of queries in both normalized and un-normalized form. We evaluated our results firstly by comparing the Recall and Relevance figures returned by the engine for each request, and afterwards by comparing user-assigned Relevance[4] figures. In the following section we present the criteria applied for the selection of the sample queries and we conclude with a discussion of the evaluation results.

### 5.1 Sample Queries

During selection of the query sets we paid particular attention to cover those representing different levels of searching complexity. Moreover, we did not want to omit from our study real reference questions issued by end users. Thus, four of the following eight queries were extracted from real reference questions collected from the engine's log files and the remaining were composed by the authors. The two sets of queries have common characteristics, since they are both too general in relation to the user's search intention.

**Reference Questions**

***1#*** *παιχνίδια λογικής (mind games)*
***2#*** *βιογραφίες (biographies)*
***3#*** *βία και αθλητισμός (violence in athletics)*
***4#****Πανεπιστήμιο Αθηνών (Athens university)*

**Composed queries**

***5#*** *ποιες ταινίες προβάλλονται στους κινηματογράφους (what's on the movies)*
***6#*** *περιοδικά αυτοκινήτου (car magazines)*
***7#*** *καλοκαιρινές διακοπές (summer holidays)*
***8#*** *Δώρα Χριστοδούλου* (a female proper name)

The queries used for our experiments cover a wide variety of question types that users submit to web search engines. In particular, one question is a single term query (**2#**), one constitutes an expression (**7#**) and another tests the field search capability of the engine (**4#**). Some queries require the use of Boolean logic (**1#**, **3#**, **6#**), whereas others are case sensitive (**8#**). Finally, a query (**5#**) was formulated in natural language and a syntactically ambiguous term was issued (**6#**) in order to test the normalizer's capability in ambiguity resolution. According to the specific syntax of the engine, the queries actually typed are listed below:

***1#*** *+παιχνίδια +λογικής*
***2#*** *βιογραφίες*
***3#*** *+βία +αθλητισμός*
***4#*** *title:"Πανεπιστήμιο Αθηνών"*
***5#*** *ποιες ταινίες προβάλλονται στους κινηματογράφους*
***6#*** *+περιοδικά +αυτοκινήτου*
***7#*** *"καλοκαιρινές διακοπές"*
***8#*** *Δώρα Χριστοδούλου*

The first four queries are the most frequent real reference questions issued by end users, whereas the authors selected the rest.

***Criteria applied for the manually composed queries***

For IR, the basic problem is that user requests are often mere sequences of words, without proper internal syntactic structure (Pirkola 1998). This motivated us to formulate a natural language query type (**5#**). Furthermore, we wanted to examine the normalizer's performance on case variants, thus one of the queries selected was a proper name, which is orthographically identical (homograph) to a noun but with a completely different sense[5] (**8#**). The only difference in form of the two terms is the capitalization of the proper name, but since the search engine is case insensitive it was of great interest to test whether the normalized forms of the two terms would be identical or not[6]. We also decided to check the normalizer's performance on a syntactical ambiguous query (**6#**). The query term ***περιοδικά*** can be either the plural of the noun ***περιοδικό*** *(magazine)* or the plural of the neutral gender of the adjective ***περιοδικός*** *(periodic).* The motivation behind issuing an ambiguous query was that such queries are difficult to effectively discriminate relevant from non-relevant documents, thus we wanted to examine whether a normalized wordform could improve retrieval performance.

[4] The term "relevance" is used instead of "precision" since measuring precision in a dynamic collection of data (web) is rather impossible due to continuous updates.

[5] When the term begins with an upper "Δ" (Δώρα) it is the name of a person, whereas when it starts with a lower "δ" (δώρα) it is the plural of the noun *δώρο* (gift).

[6] The normalized form of the proper name is *Θεοδώρα*, whereas for the noun is *δώρο*.

## 5.2 Performance Evaluation

To evaluate the performance of normalization, we examined the effect it has on Recall, using the engine's Recall scores. The Recall list on Table 1 presents Recall scores of each query as provided by the engine in the plain and normalized mode respectively. To evaluate the effect normalization has on the relevance of the retrieved documents we used the scores returned by the engine along with manually assigned relevance scores, given by end users. In particular, relevance of retrieved web records was determined separately by each user on the basis of the up to 10 web records retrieved for every search request. Each user visited all of the 10 pages retrieved for every query without following their internal links due to time considerations and reliability of the links. The relevance scores assigned by the five users, range between 1 (meaning absolutely irrelevant or broken link) and 5 (an exact match).

| | Recall | | Average Relevance | | | |
|---|---|---|---|---|---|---|
| | Unnorm. | Norm. | Unnorm. | | Norm. | |
| | | | Eng. | Usr. | Eng. | Usr. |
| **1#** | 108 | 1722 | 0.0266 | 0.2720 | 0.1444 | 0.2200 |
| **2#** | 243 | 1660 | 0.8200 | 0.5280 | 1.0000 | 0.4320 |
| **3#** | 45 | 244 | 0.0489 | 0.4000 | 0.0770 | 0.3880 |
| **4#** | 188 | 460 | 1.0000 | 0.8240 | 1.0000 | 0.8480 |
| **5#** | 76146 | 36105 | 0.0703 | 0.4920 | 0.3894 | 0.5480 |
| **6#** | 171 | 2152 | 0.2495 | 0.6000 | 0.5390 | 0.7680 |
| **7#** | 286 | 641 | 0.3400 | 0.3920 | 0.7800 | 0.5920 |
| **8#** | 3436 | 7852 | 0.0611 | 0.2560 | 0.1222 | 0.4400 |

***Table 1**.The evaluation results*

We then calculated the average user Relevance score for each query and converted it to a scale compatible with the scores returned by the engine, where maximum relevance=1.0. In order to delineate the normalization performance we computed recall and relevance scores for each query twice, with and without the adoption of the normalization mode. The results obtained from our experiment after examination of the first top ten ranked web resources are listed in Table 1. The precision numbers under the "Usr." column represent the mean scores on relevance of the retrieved data provided by end users, while "Eng." column contains the mean scores on relevance provided by the engine.

## 6 Evaluation of Results

Results show that linguistic normalization restricted to inflection yields improvement in recall of the retrieved data, whereas relevance depends on the search requests issued to the engine. Figure 2 shows the Relevance graphs for the results obtained by each version of the engine.

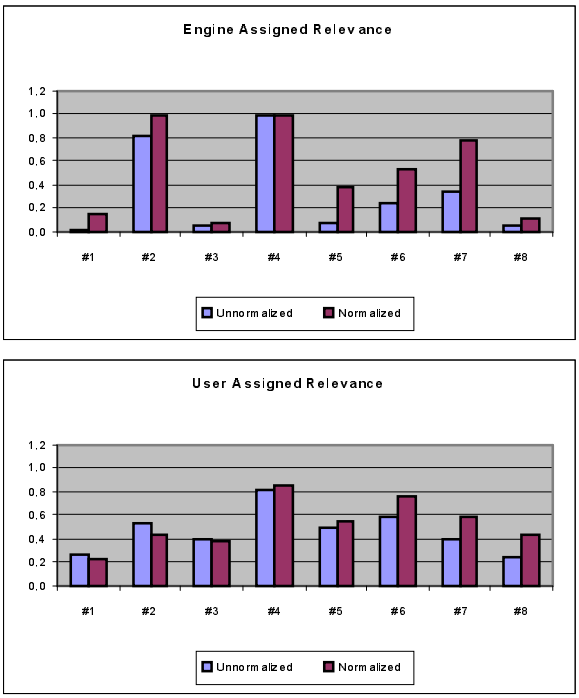


***Figure 2**.Engine-Assigned & User-Assigned Relevance Comparison.*

Compared with the results obtained from the plain version, normalization achieved a remarkable amelioration (of up to 1494% in **1#**) on recall, while its relevance is slightly decreased. On the other hand in the case of (**5#**) the recall was reduced due to the fact that (**5#**) is a "Boolean-Or" phrase query containing stop words, which were excluded from the normalized index. Its relevance however, is slightly enhanced, due to the fact that searching is conducted with morphological variants. Normalization handles better "Boolean-And" queries (**3#**) with regard to the relevance of the results and has better performance for POS ambiguous terms (**6#**) due to the POS Disambiguator. On the other hand, adoption of normalization does not affect the engine's filed search capabilities with regards to the Relevance (**4#**). Of the two versions, the normalized one, better supports phrase searching (**7#**), and it also yields Relevance improvement over single term queries (**2#**). In the case of homographs (**8#**) the

normalizer assigned the correct tag to the query term "*Δώρα*" thus resulting in a better relevance score. However, this is not always the case since the assigned tags strongly depend on the lexicon coverage. In general, the aforementioned results imply that correlating morphologically related words slightly enhances relevance in most cases.

## 7 Conclusions and Future Work

The work reported in this paper examines the application of normalization in a commercial Greek web search engine. The impact of normalization on Relevance strongly depends on the queries involved, whereas it yields a significant improvement on Recall of the retrieved data, since the search is conducted with all wordforms and not exclusively with the ones issued by users. Since normalization performance strongly depends on the lexicon supported by the normalizer, our future plans concentrate on the lexicon enrichment and enhancement so that there are no conflations with terms that might have some wordfroms in common. We also examine the possibility of applying other search facilities to the engine in order to help users issue their queries. Towards this direction we plan to apply query refinement methods by automatically generating term suggestions employed by the users as a guide for improving their queries. Our aim is the users' understanding of our approach to retrieval in order to use it more effectively for expressing their information needs.

## Acknowledgements

The authors gratefully acknowledge the contribution of Giorgos Orphanos in providing the POS tagger & the normalizer and Prof. Dimitris Christodoulakis for his support and valuable contribution. Many thanks to the anonymous reviewers for valuable comments and to Kemal Oflazer for his suggestions.

## References

Arampatzis A, Van der Weide Th, Van Bommel P, Koster C.H.A 2000 "Linguistically-Motivated Information Retrieval". To appear in Encyclopedia of Library and Information Science

Harman D 1991 "How effective is suffixing". Journal of the Am. Soc. for Information Science, 42, 1

Hull D 1996 "Stemming algorithms: A case study for detailed evaluation". Journal of the Am. Soc. for Information Science, 47, 1

Kraaij W, Pohlmann R 1996 "Viewing Stemming as Recall Enhancement" Proceedings of the 19th ACM SIGIR Conference, ACM, Zurich

Krovetz R, Croft B 1992 "Lexical Ambiguity and Information Retrieval" ACM Transactions on Information Systems, Vol. 10(2)

Krovetz R 1993 "Viewing Morphology as an Inference process" Proceedings of the 16th ACM SIGIR Conference, ACM, Pittsburgh, USA

Krovetz R, 1997 "Homonymy and Polysemy in Information Retrieval" Proceedings of ACL/ EACL'97 Conference.

Tzoukermann E., Klavans J, Jacquemin C 1997 "Effective Use of Natural Language processing Techniques for Automatic Conflation of Multi-Word terms: The Role of Derivational Morphology, Part of Speech Tagging, and Shallow Parsing". Proceedings of the 20th ACM SIGIR Conference, Philadelphia, Pennsylvania, USA

Oflazer K., Tur G. 1996 "Combining Hand-crafted Rules and Unsupervised Learning in Constraint-based Morph. Disambiguation". Proceedings of ACL-SIGDAT Conference, Philadelphia, USA

Orphanos G., Christodoulakis D. 1999. "Part-of-speech Disambiguation and Unknown Word Guessing with Decision Trees", Proceedings of EACL'99, Bergen, Norway

Pirkola A 1998 "The Effects of Query structure and Dictionary Setups in Dictionary-Based Cross-Language Information Retrieval" Proceedings of the 21st ACM SIGIR Conference, Melbourne.

Popovic M, Willet P 1992 "The Effectiveness of Stemming for Natural Language Access to Slovene Textual Data" Journal of the Am. Soc. for Information Science, Vol.3, No.5

Porter M 1980 "An Algorithm for Suffix Stripping" Program, Vol. 14 (3)

Strzalkowksi T, Lin F, Perez-Carballo J 1997 "Natural Language Information Retrieval" TREC-6 Report.

Xu J, Croft B 1998 "Corpus-Based Stemming Using co occurrence of Word Variants", ACM Transactions of Information Systems, 16, 1, pp.61-81