Keyword Identification within Greek URLs

Maria-Alexandra Vonitsanou, Lefteris Kozanidis, and Sofia Stamou

Abstract—**In this paper we propose a method that identifies and extracts keywords within URLs, focusing on the Greek Web and especially on URLs containing Greek terms. Although there are previous works on how to process Greek online content, none of them focuses on keyword identification within URLs of the Greek web domain. In addition, there are many known techniques for web page categorization based on URLs but, none addresses the case of URLs containing transliterated Greek terms. The proposed method integrates two components; a URL tokenizer that segments URL tokens into meaningful words and a Latin-to-Greek script transliteration engine that relies on a dictionary and a set of orthographic and syntactic rules for converting Latin verbalized word tokens into Greek terms. The experimental evaluation of our method against a sample of 1,000 Greek URLs reveals that it can be fruitfully exploited towards automatic keyword identification within Greek URLs.**

Index terms—**Greek to Latin character set transliteration, Greeklish to Greek transliteration, keyword extraction, Uniform Resource Locator, word segmentation.**

I. Introduction

THE large volume of information that is available over the web increases at prodigious rates with the current size of the surface web reaching to nearly 15 billion pages [1]. This, coupled with the need for accurate and effective identification of useful information within this huge network of data sources, has made imperative the need to come up with efficient methods for organizing, processing and structuring the plentiful web content. Towards this direction, several researchers have proposed methods for classifying the web content thematically so as to facilitate the data storage and the information seeking processes. The most commonly employed approaches towards web data classification focus on the examination of three main features extracted from web pages, namely their textual content [2], their anchor text and internal link distribution [3] and their URL features [4], [5], [6], [7], [8]. Although there exist several techniques for each of the above approaches, there is still ample room for improvements as none of the existing tools and methods can successfully detect the topic of every single page on the web and thus be able to assign it to a suitable thematic category.

In this paper, we address the problem of URL-based keyword extraction for web page classification from the perspective of a Greek Web search engine. In particular, we study the problem of URL features analysis in order to capture web resource's thematic orientation. The extracted features can be used in order to automatically categorize Greek web content based entirely on URLs and without the need to perform content analysis which is a time-consuming and laborious process. The motive for carrying out our study is the observation that Greek URLs are articulated via the use of Latin characters and as such even if they encapsulate useful information within their elements, we have to translate this information to Greek in order to be able to interpret it.

In this paper, we address the problem of URL-based keyword extraction for web page classification from the perspective of a Greek Web search engine. In particular, we study the problem of URL features analysis in order to capture web resource's thematic orientation. The extracted features can be used in order to automatically categorize Greek web content based entirely on URLs and without the need to perform content analysis which is a time-consuming and laborious process. The motive for carrying out our study is the observation that Greek URLs are articulated via the use of Latin characters and as such even if they encapsulate useful information within their elements, we have to translate this information to Greek in order to be able to interpret it.

It is common knowledge that the URL is a string that specifies the mechanism to retrieve the identified sources, providing a scheme, a host or IP address and a path. Relying on pages' URLs rather than their textual content for web data classification is more time and cost effective since working with strings (i.e. URLs) instead of full documents diminishes the computational complexity and the network overheads associated with data processing. Extracting keywords from a URL can be useful because when no anchor text exists or the web resource is not a web page, it is the only available information about the web resource of interest. URL-based web page classification mainly concerns the identification of keywords within URLs that could serve as terminological descriptors of the corresponding pages' topics. But keyword extraction from URLs is not an obvious straight forward process, because URLs may or may not contain valid terms, they might contain symbols, special characters, they may conflate alphanumerics to abbreviate a phrase or a name and so forth. Conversely to the URLs' content, which is difficult to capture and interpret as this does not follow any specific guidelines, the URLs' structure is indicative of the topology of their corresponding web pages on the web graph. Therefore, the majority of works that try to capture the properties of web

Manuscript received November 1, 2010. Manuscript accepted for publication January21, 2011.

M.-A. Vonitsanou is with the Computer Engineering and Informatics Department, Patras University, 26500, Greece (e-mail: bonitsan@ ceid.upatras.gr).

L.Kozanidis is with the Computer Engineering and Informatics Department, Patras University, 26500, Greece (e-mail: kozanid@ ceid.upatras.gr).

S.Stamou is with the Computer Engineering and Informatics Department, Patras University, 26500, Greece and the Department of Archives and Library Science, Ionian University, 49100, Greece (e-mail: stamou@ ceid.upatras.gr).

pages based on the analysis of their URLs mainly focus on building URL parsers that could interpret the URL syntax. The few reported attempts that try to identify the topic of a web page based on the interpretation of the keywords identified with the page's URL, generally focus on English URLs. In this paper, we focus on the analysis of Greek URLs in order to identify and extract meaningful terms from their elements. The main challenge we need to confront is the fact that Greek URLs contain terms written in Latin and that Greek words can be transliterated in many different ways such as phonetic, orthographic or visual, depending on personal references. In addition, apart from the syntactically valid work combinations within URLs, the Greek language being a free word order one, allows multiple combinations of terms, some of which result to word phrases not necessarily encoded in general-purpose dictionaries.

Although there are previous works on how to process Greek online content [9], [10], [11] none of the reported attempts has focused on the problem of keyword identification within URLs of the Greek Web domain. In our work, we address the problem of keyword extraction from Greek URLs by implementing a system that integrates a transliteration engine and a URL tokenizer. In brief, the URL tokenizer segments an input URL into tokens which are given as input to the transliteration engine, which in turn produces all possible variations of the URLs' Greek tokens. For generating the transliterations, our engine relies on a Greek morphological dictionary, a Greek grammar and embodies a set of orthographic rules. After a brief introduction to relevant works, we describe in detail our Greek URL-based keyword extraction method, we discuss the results of a preliminary study we carried out and we sketch our plans for future work.

II. Related Work

There exist large volumes of works on URL processing for web page classification. Among the existing studies, researchers proposed methods for segmenting URLs into meaningful chunks to which one could add components, sequential and orthographic features for modeling salient patterns and rely on them for web data organization [6].

From a different perspective, researchers suggested ways for categorizing web pages based on URL elements, metadata descriptors and text extraction techniques via three-phase pipeline of word segmentation, abbreviation expansion and eventually classification [4]. A slightly different approach [5] employs a two-phase pipeline (e.g. URL word segmentation/ expansion and classification) for reducing the content of web data sources and be able to classify pages from academic hosts into the following predefined categories: course, faculty, project and student. More recently in [7] researchers proposed a machine learning technique for identifying the topical subject of a page based on its URL feature analysis. Feature identification within URLs entailed the combination of token and n-gram representation models. From a different viewpoint in [8] URL-based web page classification relies on language detection methods and the resulting classification is according to the pages' language rather than theme. Still, the case of Greek has not been investigated in any of the related works.

Related work falls also within the subject of Greek transliterations using the Latin alphabet for enabling Greek web content management. In this direction the work of [9], [10] focuses on the identification of query keywords from Greek web content and the subsequent handling of web queries verbalized in Latin characters. In [11] the authors study ways for classifying web sites of Greek vendors based on the identification of entities within their contextual elements. With respect to Greek transliterations of Latin-scripted texts, researchers mainly rely on the application of probabilistic models [12], spell-checking techniques [13] and regular expressions approaches [14] which they unify into common transliteration platforms. Despite the availability of such tools and methods none of them has been tailored to handle transliterations embedded within URLs in which lexical boundaries are absent and there is a lack of consensus with respect to what could or should a URL contain so as to reflect the content of its hosting page.

III. The Greek Web

This section provides a brief description of the Greek Web, the Greek language and the characteristics of Greek transliterations using Latin characters

A. The Greek Web

The main difficulty in defining the properties and characteristics of the Greek web arises from the fact that the exact limits of the Greek web are vague and imprecise. A naive approach would be that the Greek Web consists of the sites registered in the .gr top-level domain. This claim would lead to incorrect results as many Greek Web sites are hosted under the .net, .com, or .org top-level domains and reverse many sites in the *.gr* domain verbalize their Greek-oriented content in via the use of English [15]. In the course of our study, we define the Greek web as the web content written in Greek. Although we are aware of the fact that our definition of the Greek Web is incomplete, we rely on that in the course of this study essentially because our research objective is to identify valid keywords within Greek URLs rather than determine and capture the boundaries of the Greek Web.

B. The Greek Language

Like most Indo-European languages, Greek is highly inflected. The Greek alphabet consists of 24 letters each with a capital and a lowercase form plus an extra form for the letter s when used in the final position. Greek demonstrates a mixed syllable structure, permitting complex syllabic onsets, but very restricted codes.

Greek is a language distinguished by an extraordinarily rich vocabulary and a powerful compound-constructing ability. Another distinctive characteristic of the Greek language is its rich inflectional morphology which may deliver for a single lemma between 7 (for nouns) to 150 (for verbs) distinct inflected forms. In addition, due to the existence of diphthongs and digraphs, spelling is significantly complicated.

Based on the above characteristics of the Greek language, we may naturally conclude that computationally processing Greek texts is a complex and laborious process that requires extensive linguistic knowledge and the availability of several resources such as dictionaries, grammars, sets of rules, corpora, etc.

C. *Greeklish*

The transliteration of Greek to Latin characters, a frequent practice on the web, has formed a hybrid language known as Greeklish.

Greeklish became widely known in the 1990's since not all operating systems and applications, especially web browsers, had support for the Greek character set. Nowadays, they are commonly used in blogs and forums because they are typed easily and users do not have to follow any orthographic rules.

Greeklish is not standardized, thus Greek words can be transliterated to Latin script in many different ways. Briefly, there are two generic types of transliterations, namely [16]:

a) Phonetic transliterations: based on how words are pronounced. For example «καλημέρα» meaning "*goodmorning*" is transliterated to "*kalimera*".

b) Orthographic transliterations: based on how the words are written. The aforementioned example is transliterated to "*kalhmera*".

Yet, there still exist quite a few variations in both orthographic and phonetic transliterations of certain Greek characters. For instance, the Greek letter *θ* (theta) may be written as *8, 9, 0, q, u* in the orthographic use of Greeklish and *th* in the phonetic use. What makes things more complicated is that oftentimes people switch between phonetic and orthographic transliterations, therefore increasing the heterogeneity of Greeklish writing.

D. *Challenges*

Given the variety of Greek to Latin characters' transliterations it becomes evident that being able to accurately reproduce Greek lemmas from Latin-scripted words becomes cumbersome and error-prone. This is not only due to multiple mappings that hold between the Greek and the Latin character sets (e.g. Greek diphthongs may match a single Latin character) but also due to the absence of specific guidelines about how Greek words are transliterated in Latin and vice versa. In addition, the lack of punctuation marks in the Latin alphabet imports an additional burden in the process of the identification of the correct Greek term as there exist many homonyms in the Greek vocabulary. For example the Greek term «γέρος» meaning "old man" and «γερός» meaning "strong man" produce the same transliteration "geros". Similar error-prone situations emerge in the case of homophones, i.e. words pronounced the same way, but spelled differently. In those cases, the word interaction should be considered.

IV. Methodology

Given the lack of a standard transliteration for Greeklish, it is extremely difficult to automatically process Greeklish data. Because of that, research on keyword extraction from URLs has not addressed the case of Greek. In this section we present in detail our method for identifying keywords within Greek URLs. In particular, we introduce the main components our method incorporates, namely the URL tokenizer and the Latin-to-Greek transliteration engine. Alongside, we introduce the resources upon which our proposed components operate and we demonstrate via examples the URL keyword identification process.

A. *URL Tokenization*

Based on the observation that a significant fraction of the URLs contain two or more word-tokens that are not delimited by non alphanumeric characters [5], it is obvious that the implementation of a tokenizer is required. Briefly, the tokenizer searches within the substring of the input token (i.e. URL), for meaningful keywords. To tackle tokenization for Greek URLs, we implemented two distinct yet complementary tokenizers, namely a surface keyword tokenizer and a hidden keyword tokenizer, both of which operate upon dictionary lookups.

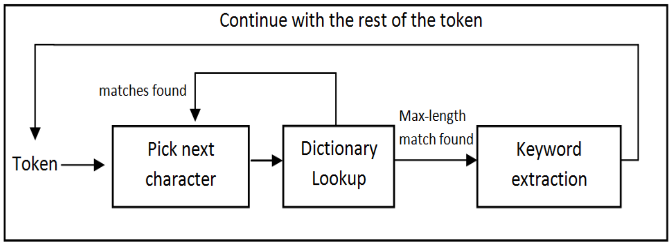

Fig. 1. Tokenization.

The surface keyword tokenizer parses the URL tokens from beginning to end, terminates upon the detection of an explicit keyword token. Unless the latter is identified, the tokenizer proceeds until reaching to a maximum size of keyword tokens and runs recursively by employing successive tokens as starting points simulating an n-gram examination process. For example, consider the URL http://www.contrastsensitivity.com where "contrastsensitivity" has to be split into "contrast" and "sensitivity". The tokenizer starting from position = 0 identifies "contrast" and starting from position = 8 identifies the keyword "sensitivity". The end of the token is identified, so the tokenizer terminates its function and returns the above keywords. This tokenizer works also when keywords are between unknown words. Figure 1 illustrates the tokenization process.

Consider now a typical property of URLs, i.e. that keywords are nested inside other keywords or unknown words. Obviously, using the aforementioned tokenization technique would not accurately identify the hidden URL keywords. To tackle this problem we implemented a hidden-keyword tokenizer, which begins from every character and searches for keywords continuously until reaching the end of the token. For example, consider the word “certifications” *from which we want to extract the keyword “cat”. The tokenizer first searches within “*certifications*”, then moves to position = 1 and searches “*ertifications*”, and so forth until it extracts all of the hidden keywords*.

B. Latin-to-Greek Transliteration Script Engine

The transliteration engine we implemented relies on recursive look-ups against a Greek dictionary and incorporates with a set of transliteration rules in order to effectively address the problem of the variety of Greeklish forms. In addition, to reduce the number of possible representations every Greeklish word might entail, we have integrated into our transliteration engine a set of grammatical and orthographical rules.

Based on Greeklish literature, large Greeklish corpora as well as our own experience, two different sets of transliteration rules are created, depending on whether the character is ambiguous or not. The dictionary is available in a trie structure in order to efficiently assume whether a substring is word-prefix or not.

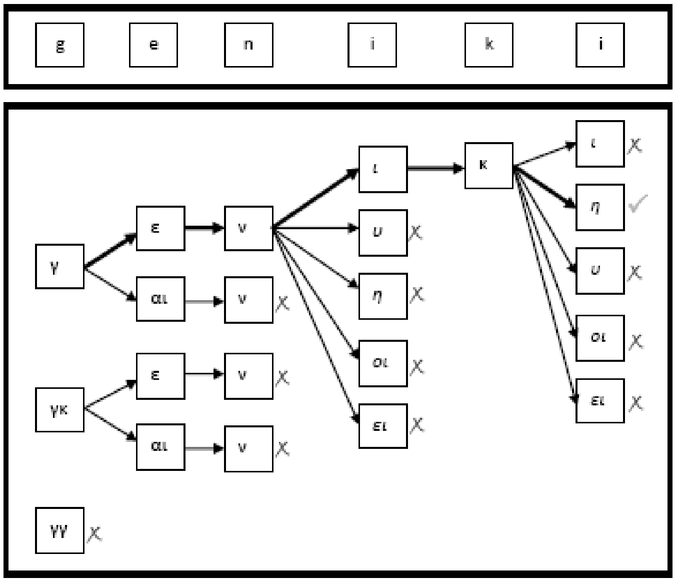

Fig. 2. Greeklish-to-Greek transliteration example.

The transliteration engine takes the following steps:

a) It replaces the unambiguous characters of a word with the corresponding characters of the Greek alphabet. For example: “geniki” (meaning general) is semi-translitered to “gevικi”.

b) The output of the previous operation is processed letter by letter, in ascending order, in a way that a tree-like structure (Figure 2) is produced. Using the rules for ambiguous characters every letter is represented in all possible ways.

c) In every level (letter) a dictionary look-up is performed. If the given substring is word prefix, we move on to the next level. Otherwise, this branch is terminated and deleted.

d) As soon as the above process terminates, the output words are returned. In most cases only one word is returned.

A special issue that has to be addressed is the transliterations of two Latin characters to one Greek (“ph”→ “φ”) and reverse (“ι”→ “οι”). Towards this direction we use a similar double-letter processing only on specific positions depending on whether the word contains such characters. Using the above method, it is obvious that computational burden is considerably reduced compared to methods that require the production of all the possible transliterations. Figure 2 schematically illustrates a transliteration example demonstrating the significant reduction of the required transliterations from 150 to 7. The complexity depends on the input string’s length and the number of possible transliterations in every step.

Additionally, orthographic rules are also applied in order to extract misspelled keywords, like trying with double consonants. For example, gramata (meaning letters) is also tried as grammata in order to extract its correct form (γράμματα). In that way, we avoid the use of a Speller. Moreover, in order to create a Greeklish-to-Greek dictionary, containing usual URL keywords, we store locally every successfully one-way transliterated word. Thus, before applying the above method, a word is searched in the transliteration dictionary and the computational burden is reduced further.

C. URL Keywords’ Extraction

Having presented the functionality of every sub-system that our URL keyword identification module integrates, we now proceed with the description of the keyword identification process. The keyword extraction system consists of the following steps:

a) A URL is divided into its basic components, according to URI protocol (scheme:// host / path-elements / document . extension).

b) The host-domain part is split on the appearance of punctuation marks.

c) For each token, transliteration is applied. Firstly a look up in the produced Greeklish-to-Greek dictionary is performed and if the word is not found, the transliteration machine is activated.

d) Parallel transliteration, tokenizing is performed. Every exact match is returned and the process continues to the next character.

Figure 3 schematically illustrates the URL keyword identification process.

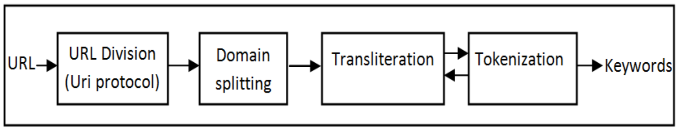

Fig. 3. URL’s Keywords identification process.

TABLE I
EXAMPLES OF SPECIAL CASES

URL	Keywords Extracted	Correct Keywords	Explanation
http://www.pamediakopes.gr/	πάμε, διακοπές	πάμε, διακοπές	Two-keyword URL
http://www.politis-chios.gr/	πολίτης, πωλητής, Χίος	πολίτης, Χίος	Ambiguous Greeklish term
http://www.mila-elefthera.gr/	μήλα, μίλα, ελεύθερα	μίλα ελεύθερα	Homophones
http://www.ellinikospiti.gr	ελληνικός	ελληνικό, σπίτι	Inflection
http://www.iatridis.gr/	ιατροί	Ιατρίδης	Named entity
http://www.gatospito.com/	γάτος	γατόσπιτο	Unknown compound
http://www. skeftomastellinika.com/	σκεφτόμαστε, νίκα	σκεφτόμαστε, ελληνικά	Word Overlapping
http://www.tapote.gr/	ποτέ	ΤΑΠΟΤΕ	Unknown abbreviation

V. EXPERIMENTAL EVALUATION

In this section we outline a preliminary experimental study we carried out in order to assess the performance of our method in accurately identifying meaningful keywords within Greek URLs.

A. Experimental Setup

To assess our study objective, we applied our proposed technique on a set of sample URLs. This dataset was collected manually and consists of 1,000 distinct URLs, which belong to different domain names, containing Greek terms. Via the use of a general purpose web crawler we downloaded every web page corresponding to the above URLs and we extracted its title and the meta-keywords, i.e. keywords that the webpages' authors have identified for describing the respective pages' content and are found as values to html meta-tags. From the initial dataset, we could only download 958 web pages of which only 942 had an associated title, 547 contained meta-words and only 400 of them had both an associated title and meta-words. In cases that the page title or meta-words were missing the only information we had at our disposal besides the page content was the page URL. For our experiments we used only the 400 URLs that contained both title and meta-words.

We compared the title and meta-words extracted against the keywords our method identified in the corresponding URLs. To assess our method's effectiveness in detecting valid keywords within Greek URLs we carried out two experiments:

a) One using the surface keyword tokenizer, and
b) One using the hidden-keyword-tokenizer.

B. Experimental Results

We perform exact match assessment; each extracted keyword is searched in the title or meta-keywords of the web page.

Using the surface keyword tokenizer, 49% of the extracted keywords per URL were found in the URL's title, 43% in the meta-words, 33% in both title and meta-words and 60% in title or meta-words. Using the hidden-keyword-tokenizer, the results reduced significantly, due to the large number of keywords extracted per URL.

The main weakness of our proposed system it that when named entities are contained within Greek URLs, it fails to recognize them as such and therefore it is ineffective in extracting keywords from them. As a consequence, obtained results might be misleading especially when dealing with named entities not lexicalized in the dictionary. In addition, several web sites contain in their titles or meta-keywords terms such as "Home", "Introduction", the domain itself or non-Greek words, instead of containing topic keywords.

Moreover, we have to consider that Greek words are highly inflected. Thus, as every word might occur in many different forms, the exact matching would not recognize the word similarity, and the results can be misleading. In light of this observation, a Greek lemmatizer or stemmer should be incorporated in the comparison task.

Nevertheless, despite the above few error-prone situations, results demonstrate that in overall, our proposed technique can effectively capture a considerable amount of valid keywords within URLs. This coupled with the acknowledged lack of existing keyword detection techniques from Greek URLs validates the usefulness and the potential of our proposed method towards organizing Greek web content based entirely on the analysis of their URLs.

VI. CONCLUSION

In this paper we introduced a keyword extraction technique focusing on Greek URLs. Our proposed technique consists of two main subsystems: a transliteration engine and a tokenizer. The transliteration engine produces the possible reconstructions of the Greeklish tokens using a dictionary in order to reduce them. After transliteration, if a token consists of more than one keyword, the tokenizer segments the Greek tokens into meaningful keywords. Based on our experimental results, this paper shows that quality keyword extractors for Greek web pages can be built based on URLs alone.

The innovative aspect of our work is that we process non-English URLs, particularly URLs that contain keywords written in Greek using Latin characters.

Future work concentrates on, but is not limited to the following issues. Abbreviation handling is a significant issue in URL processing. For this study we used a common-abbreviations list. However we are working on an abbreviations' identifier based on Greek URLs. In addition, like previous attempts, we will process the path and the query part of the URL, adding a system that recognizes and decompiles percent encoding. Moreover, we are planning to improve the hidden-keyword tokenizer in order to reduce the extracted keywords that are not related to the URL. We are also working on adding a Greek stemmer to each extracted keyword and obtain keywords synonyms using Wordnet [16] in order to receive an improved match between URL keywords and meta- or title- keywords. Finally, the experiments will be repeated using a larger data set and a language detector, in order to recognize English keywords that may be contained within a Greek URL.

REFERENCES

[1] The size of the World Wide Web. Available: http://www.worldwidewebsize.com.

[2] S. Dumays and H. Chen, "Hierarchical classification of web content," in *Proceedings of the 23rd annual international ACM SIGIR Conference on Research and development in information retrieval*, Velingrad, Bulgaria, 2000, pp. 256–263.

[3] S. Chakrabarti, K. Punera, and M. Subramanyam, "Accelerated focused crawling through online relevance feedback," in *Proceedings of the International World Wide Web Conference (WWW2002)*, Honolulu, 2002, pp. 251–262.

[4] M.-Y. Kan, "*Metadata extraction and text categorization using Universal Resource Locator expansions*", National University of Singapore, Department of Computer Science, Technical Report, TR 10/03, 2003.

[5] M.-Y. Kan, "Web page classification without the web page," in *Proceedings of the 13 th. International World Wide Web Conference (WWW2004)*, New York, USA, 2004, pp. 262–263.

[6] M.-Y. Kan and H.-O.-N. Thi, "Fast webpage classification using url features," in *CIKM 2005: Proceedings of the 14th ACM international conference on Information and knowledge management.* New York, USA: ACM, 2004, p. 325-326.

[7] E. Baykan, M. Henzigner, L. Marian, and I. Weber, "Purely url-based topic classification," in *Proceedings of the 18th international World Wide Web Conference (WWW2009)*, Madrid, Spain, 2009, pp. 1109–1110.

[8] E. Baykan, M. Henzigner, and I. Weber, "Web page language identification based on URLs," in *Proceedings of the VLDB Endowment 1(1)*, Auckland, New Zealand, 2008, pp. 176–188.

[9] S. Stamou, L. Kozanidis, P. Tzekou, and N. Zotos, "Query selection for improved Greek web searches." in *Proceedings of the 2nd International CIKM Workshop on Improving Web Retrieval for non-English Queries*, CA, USA, 2008, pp. 63–70.

[10] P. Tzekou, S. Stamou, N. Zotos, and L. Kozanidis, "Querying the Greek web in greek-lish," in *Proceedings of the SIGIR Workshop on Improving Web Retrieval for non-English Queries*, Amsterdam, Netherlands, 2007, pp. 29–38.

[11] D. Farmakiotou, V. Karkaletsis, G. Samaritakis, G. Petasis, and D. Spyropoulos, "Named entity recognition in Greek web pages," in *Proceedings Companion Volume of 2nd Hellenic Conference on Artificial Intelligence (SETN-02)*, Thessaloniki, Greece, 2002, pp. 91–102.

[12] A. Chalamandaris, A. Protopapas, P. Tsiakoulis, and S. Raptis, "All greek to me! an automatic greeklish to greek transliteration system," in *Proceedings of 5th International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy, 2006, pp. 1226–1229.

[13] Aspell, spell checker for Greek. Available: http://aspel.source.gr.

[14] A. Karakos, "Greeklish: An experimental interface for automatic transliteration," *Journal of the American Society for Information Science and Technology*, vol. 54, pp. 1069–1074, 2003.

[15] C. Lampos, M. Eirinaki, D. Jevtuchova, and M. Varzigiannis, "Archiving the greek web," in *Proceedings of the 4th Intl. Web Archiving Workshop*, Bath, UK, 2004.

[16] WordNet. Available: http://www.cogsci.princeton.edu/~wn.