ACM SIGIR 2007 Workshop

Improving Non English Web Searching (iNEWS'07)

held in conjunction with the 30th Annual International
ACM SIGIR Conference

27 July 2007, Amsterdam

Organizers

Fotis Lazarinis
Technological Educational Institute of Mesolongli, Greece
Jesus Vilares
University of A Coruna, Spain
John I. Tait
University of Sunderland, UK

ISBN: 978-84-690-6978-3

iNEWS'07

First Workshop on

Improving Non English Web Searching

Edited by Fotis Lazarinis, Jesus Vilares and John I. Tait

Sponsors

This workshop has been partially funded by:

XUNTA DE GALICIA

"Rede Galega de Procesamento da Linguaxe e Recuperacion de Informacion (*Galician Network for Language Processing and Information Retrieval*)"

Volume Editors:

Fotis Lazarinis
Technological Educational Institute of Mesolongli, Greece

Jesus Vilares
University of A Coruna, Spain

John I. Tait
University of Sunderland, UK

ISBN: 978-84-690-6978-3

Preface

The First Workshop on Improving Non English Web Searching (iNEWS’07) took place on July 27 in Amsterdam (The Netherlands) in conjunction with the 30th Annual International ACM SIGIR Conference (SIGIR’07) aiming at bringing together researchers interested in non-English web searching.

Nowadays, over 60% of the online population are non-English speakers and it is probable the number of non-English speakers is growing faster than English speakers. Recent studies showed that non-English queries and unclassifiable queries have nearly tripled since 1997. Most search engines were originally engineered for English. They do not take full account of inflectional semantics nor, for example, diacritics or the use of capitals.

The main conclusion from the literature is that searching using non-English and non-Latin based queries results in lower success and requires additional user effort so as to achieve acceptable recall and precision. Furthermore, international search engines (like Yahoo and Google) are relatively weaker with monolingual non-English queries.

So, new tools and resources are needed to support researchers in non-English retrieval, new methodologies need to be proposed which will help the identification of problems in existing search engines and new teaching strategies should be formed aiding users to become more efficient in formulating their queries.

Taking into account these needs, the main objectives of this workshop are the proposal of techniques and the evaluation of tools which improve the effectiveness of the existing search engines. This way, the specific aims of the workshop have been:

- Evaluate search engines in non-English queries and measure the additional user effort.
- Define methodologies for evaluating the effectiveness of search engines in non-English queries.
- Study the user query patterns in non-English Web retrieval.
- Identify the factors that influence utilization of search engines in a multicultural world.
- Propose extensions to the search engines to improve non-English Web retrieval.
- Propose teaching strategies for helping users improve their searching behaviour.
- Identify how standard IR techniques (Indexing, Query representation, Query reformulation, etc) can be adapted in Web retrieval for non-English languages.
- Discuss the application of natural language processing techniques for non-English Web IR.

In response to our call, 13 papers were submitted. After a triple blind reviewing process, 4 papers were selected by the Program Committee for presentation as full papers and 6 more as short papers.

Finally, we wish to thank SIGIR organizers, the program committee and our sponsor, the "Rede Galega de Procesamento da Linguaxe e Recuperacion de Informacion (Galician Network for Language Processing and Information Retrieval)", funded by Xunta de Galicia government, for its support.

July 2007

Fotis Lazarinis
Jesus Vilares
John I. Tait

iNEWS’07 Organization

Workshop Chairs

Fotis Lazarinis, Technological Educational Institute of Mesolongli, Greece
Jesus Vilares, University of A Coruna, Spain
John I. Tait, University of Sunderland, UK

Program Committee

Miguel Alonso, University of A Coruna, Spain
Kuang-hua Chen, National Taiwan University, Taiwan
Theodore Dalamagas,National Technical University of Athens, Greece
Chu-Ren Huang, Academia Sinica, Taiwan
Ghassan Kanaan, Yarmouk University, Jordan
Noriko Kando, National Institute of Informatics, Japan
Fotis Lazarinis, Technological Educational Institute of Mesolongli, Greece
David Losada, University of Santiago de Compostela, Spain
Alexandros Ntoulas, Microsoft Search Labs, USA
Doug Oard, University of Maryland, USA
Gabriel Pereira, Universidade Nova de Lisboa, Portugal
Carol Peters, ISTI-CNR, Italy
Owen Rambow, Columbia University, USA
Mark Sanderson, University of Sheffield, UK
Jacques Savoy, University of Neuchatel, Switzerland
Nasredine Semmar, LIC2M/CEA-LIST, France
Min Song, New Jersey Institute of Technology, USA
Sofia Stamou, University of Patras, Greece
Richard Sutcliffe, University of Limerick, Ireland
John I. Tait, University of Sunderland, UK
Jesus Vilares, University of A Coruna, Spain
Manuel Vilares, University of Vigo, Spain

Table of contents

Full papers

Short papers

Invited Talk

“Who's the user? Who's the researcher?”

Maarten de Rijke

Informatics Institute, University of Amsterdam

Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

mdr@science.uva.nl

Abstract

Over the past few years there has been a lot of progress in technology used for addressing monolingual or multilingual web queries in languages other than English. Nevertheless, a great deal of work still remains to be done, e.g., on the morphological analysis of non-English web queries, before the retrieval performance on English and non-English are on a par. There's another pressing issue, however, that's at least as important: we know very little about users of monolingual or multilingual (non-English) web search facilities. Who are they? What do they search for? What are their intents? At WebCLEF --- the multlingual web retrieval track run at CLEF --- these questions and concerns have led to a very explicit definition of the retrieval task, where various assumption are being recorded as part of the topic statement. In the talk I will review the choices made at WebCLEF over the past few years and detail (and motivate) the current set-up.

Another important aspect of the talk concerns the lack of user data that most academic research groups have to work with. I discuss various ways around this, one example being the use of publicly available and usable showcases and demonstrators. We (the University of Amsterdam) have run and continue to run a small number of Dutch language online search and browsing tools. At the workshop I will discuss a number of findings of this strategy, based on a brief log analysis together with both quanitative and qualitative analyses.

This talk is based on joint work with Leif Azzopardi (Glasgow), Krisztian Balog, Valentin Jijoun, Jaap Kamps (Amsterdam), and Borkur Sigurbjornsson (Barcelona).

How do Search Engines Handle Greek Queries?

Efthimis N. Efthimiadis
Information School
University of Washington
Seattle, WA, USA
+1 (206) 616-6077
efthimis@u.washington.edu

Nicos Malevris, Apostolos Kousaridas, Alexandra Lepeniotou, and Nikos Loutas
Department of Informatics
Athens University of Economics and Business
Athens, Greece
+30 (210) 820-3126
{ngm, akousar, alex, nloutas}@aueb.gr

ABSTRACT

General or Global Search Engines maintain that have indexed over 20 billion pages worldwide [10]. But, how well do they respond to non-English queries? And, how well do they index the content of specific domains? To address this we selected the Greek web (.gr) domain and conducted an evaluation using Greek language queries involving ten search engines. These were five “global” namely A9, AltaVista, Google, MSN Search, and Yahoo!, and five Greek search engines, namely Anazitisi, Ano-Kato, Phantis. Trinity, and Visto. In the evaluation we used the methodology of searching for the name of a known organization. Eighty (80) organizations were selected and used for searching. These organizations were divided into ten (10) categories: government departments, universities, colleges, travel agencies, museums, media (TV, radio, newspapers), transportation, and banks. A table was created with the names of the organizations and their corresponding URL that uniquely identifies them. Searches were run using the Greek and English names of each organization. The ideal retrieval would be to get the website of that organization ranked first in the result set. We present the results of this evaluation, reporting on how the engines respond to Greek and Romanized or Anglicized queries, and on the best performing global and Greek search engines.

Categories and Subject Descriptors

H.3.3 Information Search and Retrieval, *query formulation, search process*; H.3.4 Systems and Software, *Performance evaluation (efficiency and effectiveness)*

General Terms

Measurement, Performance, Experimentation.

Keywords

Web Search Evaluation, Greek Queries, Greek Web, Search Engine Evaluation.

SIGIR'07 iNEWS07 workshop, July 27, 2007, Amsterdam, The Netherlands.

1. INTRODUCTION

The web continues to expand and the dominant search engines, Google and Yahoo! claim to have indexed more than 20 billion pages [10]. Recent statistics on Internet usage by language show that 29.5% is English and 70.5% is non-English [7]. As the non-English web usage increases there are an increasing number of non-English queries that need to be handled by the search engines.

The goals of this research are (a) to evaluate how well search engines respond to Greek language queries; and, (b) to assess whether the Greek or global search engines are more effective in satisfying the user requests.

2. RELATED WORK

Bar-Ilan and Gutman [2] explored how three search engines, AltaVista, FAST and Google, respond to four non-English languages, Russian, French, Hungarian and Hebrew. They found that the search engines ignored the special language characteristics and do not handle diacritics well. Moukdad [11] studied how AltaVista, AllTheWeb, and Google handle Arabic queries compared to three Arabic engines (Al bahhar, Ayna, and Morfix). He found that the former had shortcomings in handling Arabic. Lazarinis [9] used five Greek language queries to evaluate the performance of eight search engines, six global and two Greek. He noted that there were variations in the handling of Greek. Moukdad and Cui [12] investigated how Chinese language queries are handled by Google and AlltheWeb, as well as Sohu and Baidu, the Chinese search engines. They found that the “global” search engines were not able to process the Chinese queries satisfactorily, thus introducing unexpected results.

3. THE GREEK LANGUAGE

The Greek language uses a different script to that of Latin-based languages. The Greek alphabet set has twenty four upper case letters, twenty five lower case letters and a number of diacritics or accent marks depending on the form used. The most commonly known forms of the Greek language are ancient or classical Greek, Katharevousa, and Demotic Greek (Dhimotiki). Depending on the system of accents used Greek is either polytonic or monotonic. The polytonic orthography system for Greek uses three accents, two breathings, iota subscripts and diaeresis. The polytonic system was used since the ancient times and was simplified into the monotonic system in 1982. The monotonic Greek language system uses one accent and the diaeresis, in order to signify that two adjacent vowels are pronounced separately and not as a diphthong.

Transliteration of Greek to Latin letters is common but adds to the complexity of processing Greek because of the different transliteration standards. Furthermore, individuals often ignore the standards and apply their own phonetic interpretation. The widespread use of computers and the Internet coupled with the slow progress in adopting non-Latin-based scripts has given rise to Greeklish, which is a form of transliteration used to exchange email messages and post to discussion fora.

Alevizos *et al.* [1] discuss the challenges faced by search systems in handling Greek. Kalamboukis [8] introduces the inflectional aspects of Greek and presents a stemming approach.

4. METHODOLOGY

4.1 Selecting the Search Engines

For the study we selected ten search engines based on their popularity and market share. These were divided into two groups, five global or international in scope, and five Greek search engines. The global search engines are: A9, AltaVista, Google, MSN (Live) Search, and Yahoo!. The Greek engines are: Anazitisis, Ano-Kato, Phantis, Trinity, and Visto. Appendix 8.1 lists the engines and their corresponding URLs.

4.2 User Needs and Task Definition

There has been a three fold increase in the numbers of Greeks using the Internet between 2000 and 2006, jumping from 9.1% to 33.5% respectively [6]. Similarly, the Greek web has proliferated with an increasing presence of governmental and commercial entities. In 2004, most of the Greek web pages (63.5%) were in the Greek language [4]. Though most Greeks learn a second language to some degree of proficiency, it is reasonable to assume that they would search in Greek to find information in the Greek web. Following the Broder [3] classification of web queries we selected the "navigational" class as the basis of a user task definition. We assume that a user will search to find the specific site of an organization. To that respect our methodology relates to that of Hawking et al. [4].

4.3 Queries and Subject Categories

We identified ten popular broad categories in which we selected organizations to search for. The categories are: government departments, universities, colleges, travel agencies, museums, media (TV, radio, newspapers), transportation, and banks. Using professional and business directories we selected two hundred and seventeen (217) organizations that had a web presence. For each organization we established the formal name in Greek, its non-Greek equivalent if available (usually in English or other Latin-based language) and the URL(s) of the web site.

Table 1 lists the subject categories and the corresponding numbers of Greek organizations. There were a total of 217 organizations, of which 92 had a corresponding English or other non-Greek equivalent name, thus, resulting in 309 queries.

Searches were submitted automatically to the engines in August 2006. The queries searched were the Greek and English or Romanized names of each organization. Thus, the Greek and English queries are equivalent. Examples of the queries are given in Table 2.

Table 1: Subject categories searched and number of queries.

Subject Categories		**Organizations in:**	
(in English)	(in Greek)	Greek	English
Government Departments	Υπουργεία	18	14
Universities	Πανεπιστήμια	21	20
Colleges	TEI	14	8
Travel Agencies	Ταξιδιωτικά Γραφεία	39	4
Museum	Μουσεία	19	0
Transportation & Communication Services	Μέσα Μεταφοράς, Επικοινωνίες	12	7
Banks	Τράπεζες	28	13
Newspapers	Εφημερίδες	17	16
Television Stations	Τηλεόραση	12	3
Radio Stations	Ραδιόφωνο	37	7
Total / Σύνολο		217	92

The queries were submitted for search in the typical format of typing out the keyword separated by spaces. No advance search techniques were employed in order to simulate the input of a non-expert searcher. The ideal retrieval would be to get the website of that organization ranked first in the result set.

Table 2: Examples of queries

In Greek	Equivalent in English or in a transliterated form
Υπουργείο Αγροτικής Ανάπτυξης και Τροφίμων	Ministry of rural development and food
Υπουργείο Εξωτερικών	Ministry of Foreign Affairs
Υπουργείο Μακεδονίας Θράκης	Ministry of Macedonia Thrace
Εθνική Τράπεζα της Ελλάδος	National Bank Of Greece
Λαική Τράπεζα	Laiki Bank
Πανεπιστημιο Αιγαίου	University of the Aegean
TEI Σερρών	Technological Education Institute (TEI) of Serres
KM Ταξίδι & Τουρισμός	KM Travel and tourism
Η Καθημερινή	Kathimerini
Νέα Ελληνική Τηλεόραση	NET
Εκκλησία της Ελλάδος	Ecclesia
Οργανισμός Σιδηροδρόμων Ελλάδος	Hellenic Railways Organisation

4.4 Evaluation Criteria

For every search we recorded the top ten results and their rank order. Then we evaluated whether the organization's URL was found in the results set, and, if so, recorded the rank position and the number of times. The evaluation also counted whether there was an exact or partial match of the desired URL.

The score includes two components, the rank position, and the depth of the page as indicated in the URL. For example, if the correct URL were found in rank 1, then the score assigned was 100, if in rank two 90, and so on. If the URL were a partial match, that is, it came from a page in the website but not the top page, then, the score was adjusted depending on the depth of the page retrieved. The latter gives some credit for partial matches, assuming that the searcher will be able to identify that the returned result is related to the desired result. This way the search engine is penalized for the additional navigational effort that will be required by the user.

5. RESULTS

5.1 Qualitative aspects of searching

Table 3 presents how search engines respond to Greek queries that either have or do not have accent marks. It also shows whether the engines handle articles, prepositions, pronouns, etc. The five global search engines and one Greek return different results. The differences observed in the top ten results vary from providing totally different results, to having some small overlap in the results, but with differences in rank order.

Table 3: How search engines handle Greek accent marks

Search Engine	**Greek with or without accents produce:**	**Handling of articles, prepositions, etc.**	
		Greek	**English**
Anazitisis	different results	No	No
AnoKato	same results	No	Yes
Phantis	same results	Yes	Yes
Trinity	same results	Yes	Yes
Visto	same results	Yes	Yes
A9	different results	No	Yes
Altavista	different results	No	Yes
Google	different results	No	Yes
MSN	different results	No	Yes
Yahoo	different results	No	No

The search results should be the same with or without the accent marks. For example, in Greek the meaning of the words is usually not affected by punctuation. Since in the majority of the cases the meaning of the word does not change the search results should be the same. For example, the query "Πανεπιστημιο" (university) or "Πανεπιστήμιο" should return the same results.

5.2 Search Results by Rank Order

The 309 queries were submitted to each of the 10 search engines for a total of 3090 searches. Of those 276 queries or 2760 searches returned valid results, while 33 queries or 330 searches did not return any results at all.

Table 4: Rank distribution of all search results by search engine.

		Rank												
	Search Engines	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Missed**	**Total Found**	**% success rate**
Ελληνικές Greek	anokato	53	16	5	2	2	2	2	0	1	0	226	83	26.86%
	anazitisis	17	7	4	0	2	1	1	2	0	1	274	35	11.33%
	phantis	23	5	2	1	1	0	0	0	0	1	276	33	**10.68%**
	trinity	142	10	5	3	0	1	1	0	0	0	147	162	52.43%
	visto	78	20	6	4	4	2	1	1	1	0	192	117	37.86%
Διεθνείς Global	a9	106	17	11	3	4	5	3	2	1	0	157	152	49.19%
	altavista	166	30	10	2	2	2	4	2	0	3	88	221	71.52%
	google	199	11	8	1	2	3	1	1	1	1	81	228	**73.79%**
	msn	101	18	12	6	5	3	3	1	1	0	159	150	48.54%
	yahoo	133	30	13	7	3	3	3	1	2	0	114	195	63.11%

Table 4 presents the rank distribution of the results for both the Greek and English queries by search engine. The table lists also the number of organizations missed by each engine, and their success rate. Of the organizations found it appears that most results were presented in the first three ranks. The global search engines have higher success rates, ranging from 48.54% to 73.79%, than the Greek engines which range from 10.68% to 52.43%. Google is the best performing global engine and Trinity is the best Greek engine.

The above results give an overall performance rate for the search engines but do not show how the engines respond to Greek or non-Greek queries. Table 5 and Table 6 present the rank distributions of the results by language. In Table 5 we see that AltaVista and Google handle Greek queries better than all the other engines with a success rate of 72,81%, and 70,96% respectively, whereas MSN and A9 are almost tied in last rank with about 50%. The best performance of the Greek engines was recorded by Trinity with 49.3%.

Table 5: Rank distribution of results for Greek queries.

	Search	**Rank**										**Total**	**%**
	Engines	1	2	3	4	5	6	7	8	9	10	**Found**	**success**
Ελληνικές Greek	Anokato	32	11	3	1	2	2	1	0	1	0	53	24,42%
	Anazitisis	12	5	3	0	2	1	1	2	0	1	27	12,44%
	Phantis	18	1	1	1	1	0	0	0	0	1	23	**10.59**
	Trinity	94	6	3	2	0	1	1	0	0	0	107	49,3%
	Visto	63	16	4	3	4	0	1	1	1	0	93	42.86%
Διεθνείς Global	a9	82	7	9	3	3	1	2	1	1	0	109	50.23%
	AltaVista	118	23	8	1	2	0	3	2	0	1	158	**72.81%**
	Google	131	10	5	1	2	2	1	1	0	1	154	70.96%
	msn	79	9	8	6	3	2	1	1	1	0	110	50.69%
	Yahoo	104	12	8	5	3	1	2	0	2	0	137	63,13%
	total queries	217											

The rank distribution of the results from the queries in English or in a transliterated form is given in Table 6. These show mixed results, as we observe variations in performance for almost all the search engines. When compared to results from the Greek queries (Table 5) Google (80.43%) has increased its performance by about 10%, Yahoo!'s performance remained the same (~63%), whereas MSN, AltaVista, and A9 decreased theirs. Of the Greek search engines Trinity's performance increased to 59.78%, whereas the performance of all other engines decreased.

Table 6: Rank distribution of results for English queries.

	Search	Rank										Total	%
	Engines	1	2	3	4	5	6	7	8	9	10	Found	success
Greek Ελληνικές	Anokato	21	5	2	1	0	0	1	0	0	0	30	32.61%
	Anazitisis	5	2	1	0	0	0	0	0	0	0	8	**8.69%**
	Phantis	5	4	1	0	0	0	0	0	0	0	10	10,87%
	Trinity	48	4	2	1	0	0	0	0	0	0	55	59,78%
	Visto	15	4	2	1	0	2	0	0	0	0	24	26,09%
Global Διεθνείς	a9	24	10	2	0	1	4	1	1	0	0	43	46.73%
	Altavista	48	7	2	1	0	2	1	0	0	2	63	68.48%
	Google	68	1	3	0	0	1	0	0	1	0	74	**80.43%**
	msn	22	9	4	0	2	1	2	0	0	0	40	43,48%
	Yahoo	29	18	5	2	0	2	1	1	0	0	58	63,04%
	total queries	92											

For a small web domain size like the Greek web neither Greek nor Global search engines perform well. The percentages can be improved taking into consideration that the best performance on Greek is about 70% and on English or Romanized queries is about 80%. In other words, one in three Greek language queries are not answered correctly.

5.3 Search results by subject category.

Using the method discussed in the section evaluation criteria all queries were scored and then grouped by category. This enables a finer evaluation of the performance of the search engines in the study. Table 7 shows the results of this evaluation grouped by language and by subject category. Based on the scoring the larger the number the better the performance of a search engine. Google from the global engines and Trinity from the Greek engines outperformed the other engines in their respective groups. But, this is not to say that Trinity's performance is good. On the contrary when comparing the Greek and global engines the Greek engines failed miserably.

Based on the aggregate results for all search engines per category for Greek queries the coverage of the categories is in the following rank order: travel agencies, universities, banks, government departments, newspapers, colleges (TEI), radio stations, museums, transportation & communication services, TV stations. Similarly, the aggregate results for all search engines for English queries show that the rank order of the coverage of the categories is: universities, newspapers, banks, government departments, colleges, transportation & communication services, travel agents, radio stations, and TV stations. Travel agencies is the category with most variation in rank amongst Greek and English, positions 1 and 7 respectively. Newspapers also ranged from rank 5 for Greek queries to rank 2 for English queries.

6. CONCLUSIONS

This study aimed at evaluating how search engines handle Greek language queries. The study evaluated ten search engines, five Greek and five global. Our results corroborate and extend the findings of [7]. The analysis shows that the global search engines ignore the characteristics of the Greek language, hence treating Greek queries differently. Despite this finding the performance of the global search engines outperforms that of the Greek engines. A set of 309 navigational queries was used in the evaluation. The rank distribution of all search results indicates that on average the search engines retrieved the desired document in the first three rank positions. However, the rate of success leaves much to be desired as the most successful engine, Google, was able to find the correct answer to only 73.91% of the English and 70.96% of the Greek queries. The engines seem to have poor coverage of the

Table 7: Sum of the scores of the top ten results by subject category, language, and search engine.

	search engines	Government Departments Υπουργεία	Newspapers Εφημερίδες	Transportation & Comn. Services Μέσα Μεταφοράς-Επικοινωνιών	Banks Τράπεζες	Universities Πανεπιστήμια	Radio Stations Ραδιόφωνο	Colleges TEI	Travel Agents Ταξιδιωτικά Γραφεία	TV stations Τηλεόραση	Museums Μουσεία
	Greek - Ελληνικά										
Greek Ελληνικές	anazitisis	**407**	**63**	**321**	702	**776**	529	243	1375	**396**	950
	anokato	767	502	200	580	1221	460	**190**	2268	634	200
	phantis	891	412	483	**532**	978	652	459	**1065**	502	**80**
	trinity	**2338**	1488	**1429**	2386	2614	**271**	**1900**	**3641**	806	672
	visto	1403	850	930	1340	2073	930	316	1650	400	300
Global Διεθνείς	a9	2206	1334	1030	2044	2304	1306	1478	2094	524	1304
	AltaVista	2145	1776	1311	2666	2385	**2046**	1493	3192	1049	1602
	google	2289	**1866**	1312	**2953**	**3039**	1841	1817	3100	**1169**	**1712**
	Msn	2206	1262	1030	2126	2418	1242	1430	2114	524	1260
	yahoo	1848	1435	1073	1985	1953	1519	1527	2827	818	1396
	Totals:	16500	10988	9119	17314	19761	10796	10853	23326	6822	9476
	English - Αγγλικά										
Greek Ελληνικές	anazitisis	413	897	170	381	958	180	106	149	7	
	anokato	865	1376	**70**	**261**	1371	**642**	**0**	290	**0**	
	phantis	437	**554**	90	495	1197	152	242	271	81	
	trinity	1528	**1957**	**849**	1519	2818	363	950	**298**	298	
	visto	**300**	1243	130	290	**75**	290	**0**	280	**0**	
Global Διεθνείς	a9	822	703	541	1244	2656	**96**	1091	**135**	172	
	AltaVista	1275	1109	494	1458	2844	136	1085	245	**361**	
	google	**1609**	1875	688	**1650**	**2926**	352	**1169**	281	181	
	Msn	822	688	531	1244	2391	**96**	1047	**135**	100	
	yahoo	1224	1057	495	1119	2785	130	1044	254	100	
	Totals:	9295	11459	4058	9661	20021	2437	6734	2338	1300	

Greek web, and the results returned by the engines are different depending on how the searcher has typed the Greek query, e.g., with or without accents. Therefore, the implications for Greek users are many as they need to be aware of the nuances to searching using Greek.

7. REFERENCES

[1] Alevizos T., Galiotou E., Skourlas C. (1988) Information retrieval and Greek-Latin text. International online information meeting. 12, London (06/12/1988). Learned Information Europe, Oxford, UK. pp. 791-801.

[2] Bar-Ilan, J., Gutman, T. (2005) How do search engines respond to some non-English queries. Journal of Information Science, 31(1), Pages: 13-28, 2005.

[3] Broder, A. (2002) A taxonomy of web search. *SIGIR Forum* 36, 2 (Sep. 2002), 3-10. http://doi.acm.org/10.1145/792550.792552

[4] Efthimiadis, E.N. and Castillo, C. (2004) Charting the Greek Web. In: ASIST'04: American Society for Information Science and Technology Annual Conference. Providence, Rhode Island, November 13-18, 2004.

[5] Hawking, D., Craswell, N., and Griffiths, K. Which search engine is best at finding airline site home pages? CMIS Technical Report 01/45 (March, 2001). http://es.csiro.au/pubs/craswell_tr01.pdf

[6] Internet World Statistics. (2007) Greece: Internet Usage and Marketing report. Retrieved May 15, 2007, available at http://www.internetworldstats.com/eu/gr.htm.

[7] Internet World Statistics. (2007) Internet World Users By Language (3/19/07). Retrieved May 15, 2007, available at http://www.internetworldstats.com/stats7.htm

[8] Kalamboukis, T. Z. (1995). Suffix Stripping with Modern Greek. *Program, 29*(3), 313-321.

[9] Lazarinis, F. (2005) Do search engines understand Greek or user requests "sound Greek" to them? In Open Source Web Information Retrieval Workshop (IEEE/WIC/ACM International Conference on Web Intelligence & Intelligent Agent Technology, France), 2005, 43-46.

[10] Mayer, T. (2005) Our Blog is Growing Up, And So Has Our Index. Yahoo! Search Blog, posted 8/8/05, retrieved May 15, 2007, http://www.ysearchblog.com/archives/000172.html

[11] Moukdad, H. (2004) Lost In Cyberspace: How Do Search Engines Handle Arabic Queries? In: Access to Information: Technologies, Skills, and Socio-Political Context. University of Manitoba, Winnipeg, Manitoba. June 3 - 5, 2004. Proceedings Editors: H. Julien and S. Thompson. Available at: www.cais-acsi.ca/proceedings/2004/moukdad_2004.pdf

[12] Moukdad, H., Cui, H. (2005) How do search engines handle Chinese queries? Webology, Volume 2, Number 3, October, 2005. http://www.webology.ir/2005/v2n3/a17.html

8. Appendix

8.1 List of search engines used in the study

Global Search Engines:

A9: (http:// www.a9.com/)

Google (http://www.google.com.gr/)

Yahoo (http://www.yahoo.com/)

Altavista (http://www.altavista.com/)

MSN (http://www.msn.com/)

Greek Search Engines:

Anazitisis (http://www.anazitisis.gr/)

Ano-Kato (http://www.ano-kato.com/)

Phantis (http://www.phantis.gr/)

Trinity (http://www.trinity.gr/)

Visto (http://www.visto.gr/)

A Fine-Grained Model for Language Identification

Harald Hammarstrøm
Department of IT
Chalmers University
S-412 96 Gothenburg
Sweden
harald2@cs.chalmers.se

ABSTRACT

Existing state-of-the-art techniques to identify the language of a written text most often use a 3-gram frequency table as basis for 'fingerprinting' a language. While this approach performs very well in practice (99%-ish accuracy) if the text to be classified is of size, say, 100 characters or more, it cannot be used reliably to classify even shorter input, nor can it detect if the input is a concatenation of text from several languages. The present paper describes a more fine-grained model which aims at reliable classification of input as short as one word. It is heavier than the classic classifiers in that it stores a large frequency dictionary as well as an affix table, but with significant gains in elegance since the classifier is entirely unsupervised. Classifying a short input query in multilingual information retrieval is the target application for which the method was developed, but also tools such as spell-checkers will benefit from recognising occasional interspersed foreign words. It is also acknowledged that a lot of practical applications do not need this fine level of granularity, and thus remain largely unbenefited by the new model. Not having access to real-world multi-lingual query data, we evaluate rigorously, using a 32-language parallel bible corpus, that accuracy is competitive on short input as well as multi-lingual input, and not only for a set of European languages with similar morphological typology.

1. INTRODUCTION

The language identification problem is to decide for a natural language text which language it is written in. The usual setting is to assume that one has access to training corpora beforehand for the languages to be considered. Some language fingerprint model is built from the training corpora and then classification of unseen text (belonging to one of the languages at hand) is performed through this model.

Existing state-of-the-art techniques rely on a surprisingly simple model, namely, a frequency table of character 3-grams for each language, read off directly from the training corpora. The corresponding 3-gram frequency table for the text to be classified is then compared to each stored language by some rank-frequency metric. In practice, this approach performs very well (99%-ish accuracy) if the text to be classified is of size, say, 100 characters or more [12]. Thus the language identification problem is a solved problem for most practical applications.

However, the crude 3-character gram method has a certain drawback (which may or may not be practical problem), in that it is not monotone. That is, if two texts s_1, s_2 are classified as l_1, l_2 respectively, then it is not certain that the concatenation of s_1 and s_2 is classified as either l_1 or l_2.

We will present an alternative model which aims at reliable classification of new text as short as one word. This model combines a frequency dictionary from each training corpus and a component that tries to recognize completely unseen words by looking at affixes (which would e.g. identify a word like *jihading* 'fighting the jihad' correctly as English). This latter component is crucial, not only for languages which make more use of affixes than English, but because there will always pop up completely novel words for any natural language no matter what size the training data. The affix detection technique implemented also builds from the same training corpora and requires no extra supervision or work by a human.

There are certainly practical applications which do require reliable classification of small segments and autodetection of language switches. These include spell checkers that wish to disregard interspersed foreign words, text-to-speech systems that make intermediate use of grapheme-to-phoneme conversion likewise wish to indentify interspersed foreign words, and multilingual information retrieval systems would benefit from knowing the language(s) of the words of a short query. For a lot of other practical applications, the granularity of the proposed new model is superfluous. For these applications, the only advantage of the proposed model is elegance and absolute lack of training supervision.

The resultant language identifier is evaluated using bible corpora for 32 languages, spanning the full range of morphological typology of languages of the world [7]. Both its ability to classify short segments into one language and to autodetect short segments that may be composed of several languages, are evaluated. However, we do not compare these figures to existing systems, because they were not designed for classifying short segments accurately (and thus perform very poorly)[1]. On longer segments, i.e. 100 char-

[1] There would also have been practical problems in doing justice as many descriptions of existing systems hide information on parameter tweaking. Online systems we have

acters, performance is near perfect, and it is presumed that the state-of-the-art systems would also perform near perfect if tested on the same set.

With the improved accuracy on short segments and wide typological testing range, we hope to have met the challenges for written language identification set out in a recent survey article by [11].

All the training corpora used in this paper are bible corpora, since they are the only sufficiently large corpora available for a reasonably varied set of languages.

2. PREVIOUS WORK

My full bibliography of works dealing narrowly with written language identification spans over 100 articles, a handful of technical reports and one PhD thesis [25] – it is therefore not possible to review them all here. Many pointers to older work and language identification of speech signals are given in [19, 2]. [22] is an excellent review and comparison of techniques used in early work.

For the language identification problem in the setting as in this paper, namely, written language identification trained on reference language data, two different feature models have been prevalent. One that looks at common words and one based on character n-grams [9, 3, 6, 8] – see [15, 13] for refinements of the n. The classification can then be done by comparing input text features to reference language features using rank-order statistics. More recent work in this direction has aimed at trimming overweight feature models [20, 23] or at combining n-gram and whole word features [21]. See, however [1] for a novel, completely different approach based on words clustered on sentence-co-occurrence. (The accuracy of this identifier is comparable to the older approaches, but it is not, as claimed therein, unsupervised, because there is a very large number of manually set parametres/thresholds and word-frequency statistics are gathered from curated corpora.) There is also more recent work targeting web pages specifically [24, 16, 14], that address the proper treatment of HTML tags.

Whereas the language identification problem has variously been labelled 'easy' and 'solved' [17], it depends on whether one sets the goal higher than distinguishing non-minimal noise-free samples of European languages. Some recent articles [18, 5, 4] identify practical problems where this is not so. For instance, as far as we can ascertain, the best systems in van Noord's Online Summary[2] minimally require some 20 characters of text to make a judgment at all. Nor are they capable of realizing that a sample text is a concatenation of two languages. For example, The Xerox MLTT Language Identifier[3] classifies the sentence 'good fish prefer their snake' correctly as English, the sentence 'fina fiskar sprattlar inte ofta' correctly as Swedish, but the concatenation of the two is classified as Norwegian (even though there is actually no legal Norwegian word in either sentence).

As indicated already, the present method seeks to tackle also smaller sample texts, which is crucial in order to be able to track whether a text is a composition of words from several languages. While the classic n-gram approaches have found that a good $n = 3$, i.e. that salient morphemes can be approximated as being exactly 3 characters, a more elegant alternative is to hold this variable, so that salient affixes can have any length in any language. Furthermore, we wish to extend the testing scope, as present published testing has been only on a rather small set of European languages.

3. DEFINITIONS AND PRELIMINARIES

Start with a finite non-empty alphabet Σ. The following terminology and notation will be used.

word: a non-empty finite string over Σ. Thus the set of all possible words can be denoted Σ^+. Lowercase w with subscripts will be used for variables over words. A word will be enclosed in quotes if confusion could arise otherwise.

sentence: a finite non-empty tuple of words $\langle w_1, w_2, \ldots, w_n \rangle$. Commas and brackets will be omitted when no confusion can arise. However, variables that range over tuples, e.g. $\langle l \rangle$, will always be written with brackets.

S_Σ : let $S_\Sigma = \{\langle w_1 w_2 \ldots w_n \rangle | w_i \in \Sigma^+, n \in \mathbf{N}\}$ denote the set of all possible sentences.

language: a probability distribution over sentences $L : S_\Sigma \rightarrow [0, 1]$ such that $\sum_{\langle s \rangle} L(s) = 1$.

training corpus: a finite sequence of sentences. However, we will never make use of the order of sentences, or order or words in the sentences, so a training corpus may be equated with its bag of words. Thus, if T is a training corpus, let $f_T(w)$ denote the frequency of the word w in T. Also, use $W_T = \{w | f_T(w) \geq 1\}$ for the *set* of words in the training corpus.

names and variables: Unless we are talking about existing natural languages, e.g. English, natural numbers $1, 2, \ldots$ will be used for language names. $\Sigma_1, \Sigma_2, \ldots$ will be used for their corresponding alphabets, with $\Sigma = \bigcup_i \Sigma_i$ for the mother alphabet. $L_1, L_2, \ldots$ will be used for languages, i.e. probability distributions, and coindexed $T_1, T_2, \ldots$ for training corpora (where T_i is assumed to be sampled from L_i).

The idea is of course that sentences which are illegal or ill-formed in some natural language will have zero probability and legal sentences will have a non-zero probability corresponding to their relative frequency. A natural way to see how a natural language should correspond to such a formal probabilistic language is to consider ever increasing amounts of natural language text and let the probability of each sentence be its limiting relative frequency. This correspondence requires that this limit actually exists for all sentences. If there are natural languages that do not live up to this, or which cannot be modelled so with an acceptable level of discrepancy, they should not be thought of as languages in our terminology.

Our notion of language is a generalization of the more common formalization of natural language as a *set* of sentences. We actually need this greater flexibility in order for language identifiers to exploit the fact that some words (and thus some sentences) which are legal in several natural languages may be distinguished by their different levels of frequency. It also provides a framework for gracious treatment

found do not allow uploading the training/test set we use, which is crucial in order to assess language-dependentness.

[2] `http://odur.let.rug.nl/~vannoord/TextCat/competitors.html` accessed the 25th of May 2005.

[3] `http://www.xrce.xerox.com/competencies/content-analysis/tools/guesser` accessed 20 Jan 2007.

of new words and proper names which are so ubiquitous in open domain natural language text (such as newspaper text) that they cannot be "abstracted away". With the probability model we have the power to say that any word is possible in any language, for example as a proper name, but it is more probable that an instance of e.g. 'the' is from English than in some other language where it may have occurred as a proper name.

4. A FINE-GRAINED MODEL OF LANGUAGE IDENTIFICATION

From the input of a training corpus, the proposed model characterizes a language using the following two components:

Frequency dictionary: Stores each seen word and its (relative) frequency. The frequency of seen words is a very powerful predictor of a language.

Unsupervised affix detection: Salient affixes are extracted (in an unsupervised manner), which form the basis for a probabilistic guessing of previously unseen words.

These two components are combined into a *word emission probability* distribution that aims to predict how likely a language is to have emitted a given word. In principle, a collection of such probability distributions are sufficient to make up a standard case of language identifier that always outputs exactly one language. However, we shall also use another component, a *language holdback bias*, to enable intuitively correct identification of text that is concatenated from several languages.

4.1 Word Emission Probability

A frequency dictionary FD_l is built simply as:

$$FD_l(w) = \frac{f_{T_l}(w)}{\sum_{w' \in \Sigma} f_{T_l}(w')}$$

Following [10] we use an unsupervised algorithm to gather information on the salient affixes for a given language. The algorithm uses W_l as its input and outputs a probability distribution on character strings that aims to say whether a given segment is likely to be a characteristic prefix or suffix for the language at hand. To be more precise, the probability distribution aims to capture the notion of morpheme probability that one arrives at if: 1. A linguist does a morphemic segmentation of the word types (not words tokens) occurring in a corpus, 2. The frequencies of the individual morphemes, in prefix or suffix position, are interpreted as probabilities. For example, *-qvj* would likely get zero probability in an English corpus. An example output, adapted from [10], is given in Table 1, sorted on highest probability. The outcome of the algorithm for languages which do not have any morphology at all is a fairly even spread of probability mass over initial and final characters of the words of the language in question. For reasons of space, the reader is referred to the said paper for a discussion of the inner workings and alternative algorithms.

As mentioned, the output from the affix extraction is a probability distribution over affixes. What we need is a probability distribution over words, in which any word ending in some salient suffix should have nonzero probability. One quite reasonable way to achieve this is to assign geometrically decreasing probabilities for longer and longer words. Thinking in this way, we let all observed (in W_l) word lengths get the probability mass proportional to the number of observed words with such lengths, and unseen word lengths get geometrically decreasing probability. Thus, to get a well-defined probability distribution over words based on the affix probability distribution, we multiply together the word-length mass for w with the highest (not necessarily longest!) matching, if any, affix probability, for a given word w. The details aren't interesting, but use $A_l(w)$ to denote the just described affix-based probability distribution.

Table 1: Comparative figures for prefix vs. suffix detection for three sample languages.

Swedish		English		Swahili	
för-	0.097	*-ed*	0.132	*-a*	0.100
-en	0.086	*-eth*	0.109	*wa-*	0.095
-na	0.036	*-iah*	0.099	*ali-*	0.065
-ade	0.035	*-ly*	0.090	*nita-*	0.059
-a	0.034	*-ings*	0.068	*aka-*	0.049
-ar	0.033	*-ing*	0.062	*ni-*	0.046
-er	0.033	*-ity*	0.059	*ku-*	0.044
-as	0.032	*-edst*	0.058	*ata-*	0.042
-s	0.031	*-ites*	0.046	*ha-*	0.032
-de	0.031	*-s'*	0.036	*a-*	0.031
...	...	...	...	...	...

Table 2: Some indications as to the widely differing identification cues for three languages; the polysynthetic Greenlandic versus the almost isolating Haitian creole.

Language	$\|T\|$	$\|W\|$	α	$argmax_w(FD(w))$	
Greenlandic	382188	107918	0.706	*taava* (then)	0.00857
Swedish	758773	26825	0.407	*och* (and)	0.05566
Haitian creole	904915	7796	0.335	*yo* (PL/they)	0.05531

Putting the affix detection together with the frequency dictionary to make an emission probability involves a related kind of estimatate. How much probability mass should be assigned to seen vs. unseen words? There are probably many similar alternatives, but here we have simply guessed that unseen words are like hapax words, and assigned the probability mass proportions to be like the proportion of hapax words: $\alpha_l = \frac{|\{w \in W_l | f_{T_l}(w)=1\}|}{|W_l|}$.

We are now ready to define emission probability:

$$P_l(w) = \begin{cases} (1 - \alpha_l) \cdot FD_l(w) & \text{if } w \in W_l \\ \alpha_l \cdot A_l(w) & \text{if } w \notin W_l \end{cases}$$

It can happen that there is more mass given to an unseen word than to a (rare) seen word, even within one particular language. In fact, proportions vary quite wildly between languages, as can be seen in Table 2 with figures computed on the translations of the same bible text.

4.2 Language Holdback Bias

If we have $L_1, \ldots, L_n$ languages, the previous section shows how to construct the corresponding $P_1, \ldots, P_n$ probability distributions over words. Next, we shall define a family of probability measures over *sequences of words*. There will be one probability distribution for each language tuple of the same length as the sequence to be measured:

$$P_{l_1 l_2 \ldots l_m}(w_1 w_2 \ldots w_m) = \prod_i P_{l_i}(w_i)$$

Given a sequence of words we could then naïvely decide which language(s) it most probably belonged to by listing each tuple of the appropriate length and computing which tuple has the highest probability of having generated the sequence of words. However, for several reasons, such an approach is not advisable. First, with n languages there are n^m language tuples so it would not be tractable to enumerate them all. Second, the probability measures so defined, the output will be the concatenation of the most probable language for each word individually. This is probably not what we want since many words that are legal in several languages differ in frequency. Consider a sequence of a million words indisputably belonging to language L_1, and, interspersed inside, a word that is legal in both L_1 and L_2 but slightly more common in L_2. The naïve language identifier would yield L_2 disregarding the suggestive surrounding million words of L_1. While it is technically not impossible that it is a concatenation of the two languages, a human would never see it as that. Third, it's not clear how to see if an input sequence is non-trivially legal in more than one way (i.e. there are several satisfactory language tuples). Either we insert some kind of threshold which would be hard to know how to set, or we have to say that pretty much all tuples are satisfactory identification of the sequence only with some degree variation.

For the first problem, it is easy to see that not all tuples need to be enumerated to get the maximally probable one (if we want only this one, rather than the probabilities for all). As defined, the emission probabilities depend only on a particular word, not anything else in the sequence, so maximas can be computed locally in the sequence and glued together as in any standard application of dynamic programming. For the second and third problem, we shall propose a refinement of the strategy that obviates the need for any thresholds.

We propose that a machine language identifier like ours should have a *bias* towards minimizing the number of times we change languages in an identification sequence. To be more precise, the prior probability that a sequence should switch language c times should decrease exponentially in c. Also, other things being equal, the longer the sequence the stronger the bias should be, i.e. it should not be less likely that a million word sequence should switch language once somewhere within it, than that a two-word sequence should switch language (once) within it. This is the way to say that having seen a million words of language L_1 counts for more than having seen just one word of L_1. We do not see any basis for this to be a sequential property, e.g. that language switches are significantly more (or less) likely after or before certain words, wherefore a (H)MM-modeling technique offers no advantage.

Formally, let $C(l_1 l_2 \ldots l_m) = |\{i | l_i \neq l_{i+1}\}|$ denote the number of times a change in language occurs in a language sequence. Clearly, we have $0 \leq c \leq m-1$. Let $\langle l \rangle = l_1 l_2 \ldots l_m$ be an arbitrary language tuple under consideration and $c = C(\langle l \rangle)$ its number of switches. Now, for any language identifier parametrized on c and m, we wish the bias, regardless of the particular languages at hand, to ensure that:

$$\tfrac{P(c,m)}{P(c+k,m)} \geq 2^k \qquad \text{for all } k \geq 0, m$$

$$P(c,m) > P(c,m+k) \qquad \text{for all } k \geq 1, c$$

A simple fulfilment of these is the following **Language Holdback Bias** function $B(c,m)$:

$$B(c,m) = \frac{1}{m^c} \cdot \frac{1}{\sum_{0 \leq i \leq m-1} \frac{1}{m^i}}$$

There of course alternative bias functions that also fulfill the desiderata, but this is the simplest one. Now, with the bias function defined we are ready to present our full definition of the output of the now rather sophisticated language identifier.

$$ID(w_1 \ldots w_m) = \begin{array}{l} \text{the set of all tuples } \langle l \rangle = l_1 \ldots l_m \\ \text{such that for all } \langle l' \rangle \\ B(C(\langle l \rangle), m) \cdot P_{\langle l \rangle}(w_1 \ldots w_m) \geq \\ B(C(\langle l' \rangle), m) \cdot P_{\langle l' \rangle}(w_1 \ldots w_m) \end{array}$$

The formula conveys the following: look for tuples with as few cuts (i.e. minimal c) as possible, that are such that they have higher probability, the bias respected, than any other tuple with *more* cuts. This is the key feature which eliminates the need for a threshold. Thus, for example, a word sequence will be said to be of language L_l iff it has higher probability than any division of the sequence into two parts of different languages (or three parts etc). There may be several such languages, but hardly all, so the yield will be a strong prediction.

The following more procedural reformulation of the identification function may be easier to understand. It should also make it clear that language identification is still polynomial in the sequence length, since there are still no dependencies between the word-probabilities.

1. Find minimal c such that there exists a tuple $\langle l \rangle$ with $C(\langle l \rangle) = c$ and:

$$\begin{array}{l} B(c,m) \cdot P_{\langle l \rangle}(w_1 \ldots w_m) \geq \\ B(C(\langle l' \rangle), m) \cdot P_{\langle l' \rangle}(w_1 \ldots w_m) \\ \text{for all } \langle l' \rangle \text{ with } C(\langle l \rangle) > c \end{array}$$

2. Output all tuples $\langle l \rangle$ with $C(\langle l \rangle) = c$ and:

$$\begin{array}{l} B(c,m) \cdot P_{\langle l \rangle}(w_1 \ldots w_m) \geq \\ B(C(\langle l' \rangle), m) \cdot P_{\langle l' \rangle}(w_1 \ldots w_m) \\ \text{for all } \langle l' \rangle \text{ with } C(\langle l \rangle) > c \end{array}$$

4.3 Examples

4.3.1 Example 1: The kings hon walikusoma

Consider the sequence *the kings hon walikusoma* which consists of *the*, which is of course the English definite article; *kings* is the well-known English lexical item which does occur in the training corpus – it also happens to end in *-s* which is a very common Swedish inflectional ending (but there is no lexical item 'king' or 'kings' in Swedish); *hon* is a Swedish personal pronoun, abundantly occurring in the Swedish training corpus; and *walikusoma* is a well formed Swahili word whose individual morphemes all individually occur abundantly in the Swahili training corpus – but the

Table 3: Example 1: $P_l(w)$ for a set of languages and some interesting words, followed by a selection of the more interesting tuple-probabilities.

	'the'	'kings'	'hon'	'walikusoma'
English	0.051522	0.000286	0.000003	0.000004
Swedish	0.000002	0.000040	0.000916	0.000043
Swahili	0.000218	0.000000	0.000000	0.000317

All one-language tuples	
$P_{eng,eng,eng,eng}$	1.350e-016
$P_{swe,swe,swe,swe}$	2.468e-018
$P_{swa,swa,swa,swa}$	1.878e-025
Some top one-switch tuples	
$P_{eng,swe,swe,swe}$	2.034e-014
$P_{eng,eng,swe,swe}$	1.465e-013
$P_{eng,eng,eng,swa}$	3.008e-015
The top two-switch tuple	
$P_{eng,eng,swe,swa}$	2.701e-013

Table 4: Example 2: $P_l(w)$ for a set of languages and some words that are very easy to classify, followed by examples to indicate that the dominance of a certain zero-switch tuple over some others.

	'the'	'kings'	'are'	'there'
English	0.051522	0.000286	0.002812	0.002065
Swedish	0.000002	0.000040	0.000006	0.000035
Swahili	0.000218	0.000000	0.000004	0.000006

$P_{eng,eng,eng,eng}$	8.5467629403443202e-011
$P_{swe,swe,swe,swe}$	1.2961894211016589e-020
$P_{swa,swa,swa,swa}$	2.5363460513704776e-023
...	

perfectly well-formed word 'walikusoma' does not occur in the training corpus (it would mean 'they read you').

The individual word-probabilities as well as a selection of the more interesting tuple-probabilities for the sequence as a whole, are shown in Table 3. As can be seen, the $P_{eng,eng,swe,swa}$ value beats all tuples with zero or one switches. It also happens to beat all tuples with three switches and it is the only such tuple. Therefore, in this case, the output will be exactly English, English, Swedish, Swahili.

4.3.2 *Example 2: The kings are there*

The complicated interaction seen in the previous example does not disturb the "normal" easy class of classifications. Table 4 shows the word-probabilities for the almost trivial sentence *the kings are there*. There is a certain zero-switch tuple which is way ahead of the others. As it also beats all one-switch tuples (and no other zero-switch tuple does), it will be the output of the identifier.

4.3.3 *Example 3: De la*

Table 5: Example 3: $P_l(w)$ for a set of languages and two words, followed by a selection of the more interesting tuple-probabilities.

	'de'	'la'
French	0.029172	0.016325
English	0.000000	0.000000
Swedish	0.008400	0.000001
Swahili	0.000000	0.001517
Spanish	0.033905	0.014280

$P_{fre,fre}$	0.0003174886
$P_{spa,spa}$	0.0003227756
$P_{spa,fre}$	0.0001844997
...	...

There are instances where there are several "winning" tuples, though informal tests show that this is not achieved very often. The sequence *de la* is very common to both Spanish and French. In English it is not common at all. In Swedish *de* is a personal pronoun so it enjoys a certain frequency, whereas *la* is not a word in (bible) Swedish. Similarly, *la* is a negator in Swahili and is therefore fairly frequent. Table 5 shows the relevant probabilities. The output will be only the tuples spa, spa and fre, fre, because tuples like swe, swa and spa, fre lose out because of the bias, favouring few switches.

5. EVALUATION AND DISCUSSION

Three extensive tests were performed using a parallel corpus of the bible in 32 languages, which contains languages from the isolating Maori to the record holding polysynthetic Greenlandic [7]. In order to get a sufficiently cross-language comparable evaluation, size and randomness were equalized between languages the following way. A random verse from each chapter was selected (there are 1209 chapters in the bible). This was done once for the whole language set. Of course, these verses were removed from the training data. A random word from each selected verse was selected. This word-selection was done separately for each language. For each language, we thus get a set of randomly selected words E_l. Though 1209 word-selections were made for each language, many selections happened to select the same word. Thus the size of the E_l-sets varied from 350 (for Maori) to 974 (for Greenlandic). The descrepancy is not disturbing. Words are not entities of the same kind across languages, but our classifier operates on the granularity of words, and the desiderata is an evaluation of 'accuracy per (randomly selected) word'. An alternative, e.g. selecting 1000 unique words of each language would have made interpretation of the result difficult, because for Maori, it is likely that most of the 1000 words would have been *seen* words, occurring in other verses, whereas the opposite is the case for Greenlandic.

If E is a set of tuples (possibly one-word tuples), drawn for language l, we define the accuracy $R_E(l)$ of a language identifier ID:

$$R_E(l) = \frac{|\{\langle x\rangle | ID(\langle x\rangle) = l \text{ and } \langle x\rangle \in E\}|}{|E|}$$

One-word classification: The R_{E_l} was calculated for each of the 32 languages. Since the input sequence is of length 1, there will never be any cuts, so the language identifier was set to output the language with highest probability of having emitted the input word. The E_l-sets as defined above may contain words that are "impossible" predict where they were taken from, on the basis of the word alone. For example, let's say a word w is legal in two languages but much more common in l_1 than l_2. If it happened to be drawn from L_{l_2}, it is hard to see how this can be predicted. However, we computed figures on the possible influence of this issue, and it turned out to be minor. Therefore, the results in Table 6 stand, but could be adjusted upwards by very small percentages.

Verse classification: To check how accurate the identifier was on longer segments, we chose to test on segments of roughly the length of a verse. Verses, in fact, happen to be around 100 characters long on average. From the 1209 verses selected (as above), those 100 verses thereof whose number of characters were closest to the average verse length of that language, were selected for testing. Denoting these 100-verse sets by V_l, the verse-classification accuracy R_{V_l} was calculated. This score, as well as data on average verse length, can be seen in Table 6.

4-tuple multilingual classification: A set of 1000 mixed language 4-tuples were built from $E_1, \ldots, E_{32}$ as follows.

1. Pick a random language l and pick two random words from that E_l.
2. Precede it with a random word from a random language $E_{l'}$.
3. Add a random word from a random language $E_{l''}$ at the end.

The results of this test was 193 (**19.3%**) fully correctly identified tuples and 204 (**20.4%**) with exactly one word misclassified.

Some figures are low, not surprisingly for languages with a lot of morphology, but overall we hold the results are very reasonable given the exceedingly difficult test problems of one-word and multi-language classification. It is very easy to make mistakes on single words when there are so many languages in the pool – the results are much higher if the number of competing languages is halved.

Unfortunately, we cannot contrast the verse-test with figures from competing state-of-the-art systems, as none of the systems known to us give enough details (on thresholds and such) to reconstruct a fair version of the classifier.

A matter requiring further commentary is the use of a bias function to do the job a scalar threshold value does in related work. (Human language identifiers, having the ability to assess syntactic and semantic coherence, need not use either.) Conceptually the bias function employed is nothing other than a complex system of thresholds, in terms of growth behaviour (exponential, linear etc.) rather than scalar values. Arguably, this is an elegance improvement, although it comes with the cost of being harder to understand, compute and analyse. Also, in the experiments reported above, the bias function approach experimentally outperforms a simple systems of scalar threshold values. For example, through supervised training we have tried tuning one single threshold value for all experiments, one threshold value individually for each language, different threshold values for different classification tasks (i.e. one for multi-language classification and one for single language classification) and so on, resulting in generally lower accuracy on the same test set (obviously, there is little room for presenting and discussing figures from these tests here). Nevertheless, it remains possible that some other, yet undiscovered, system of scalar thresholds is superior to the bias function.

Table 6: Accuracies for the one-word and verse tests plus average verse length in characters ($\overline{V}$).

Language	1-word	Verse	$\overline{V}$
Haitian Creole	0.839	1.00	101.79
Zarma	0.781	1.00	99.45
Kekchi	0.720	1.00	148.78
English	0.678	1.00	104.19
Maori	0.665	1.00	107.73
Hindi	0.607	1.00	119.50
Hausa	0.605	1.00	94.10
Afrikaans	0.594	1.00	103.34
Danish	0.580	1.00	89.30
Cebuano	0.573	1.00	129.48
Icelandic	0.550	1.00	95.58
Swedish	0.547	1.00	107.20
Adamawa Fulfulde	0.539	1.00	96.57
German	0.533	1.00	103.52
Albanian	0.523	1.00	114.80
Spanish	0.511	1.00	95.83
French	0.507	1.00	101.83
Swahili	0.494	1.00	105.03
Slovene	0.488	1.00	100.12
Polish	0.487	1.00	144.52
Portuguese	0.481	1.00	98.41
Esperanto	0.473	1.00	97.80
Italian	0.473	1.00	116.80
Catalan	0.450	1.00	109.70
Dutch	0.415	1.00	109.36
Lithuanian	0.396	1.00	104.99
Hungarian	0.386	1.00	102.10
Latin	0.366	0.99	112.54
Turkish	0.348	0.95	93.43
Finnish	0.345	0.99	107.88
Malayalam	0.276	0.88	128.65
Greenlandic	0.222	0.87	126.99

6. CONCLUSIONS

We have described a new model with considerable elegance for language identification on small, possibly mixed languages segments. We have also added significantly to the set of published evaluations of a language identification system with a balanced cross-language test. For larger input texts the new model has excellent accuracy, but it is bigger and slower in practice than the existing state-of-the-art systems.

7. REFERENCES

[1] C. Biemann and S. Teresniak. Disentangling from babylonian confusion - unsupervised language identification. In A. F. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing, 6th International Conference, CICLing 2005, Mexico City, Mexico, February 13-19, 2005, Proceedings*, volume 3406 of *Lecture Notes in Computer Science*, pages 773–784. Springer, 2005.

[2] D. Caseiro. Automatic language identification bibliography. `http://www.phys.uni.torun.pl/kmk/projects/ali-bib.html` accessed the 25th of May 2005., 1999.

[3] W. B. Cavnar and J. M. Trenkle. N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, US, 1994.

[4] J. F. da Silva and G. P. Lopes. Identification of

document language is not yet a completely solved problem. In *CIMCA '06: Proceedings of the International Conference on Computational Inteligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce*, pages 212–219, Washington, DC, USA, 2006. IEEE Computer Society.

[5] J. F. da Silva and J. G. P. Lopes. Identification of document language in hard contexts. In *Proceedings of the SIGIR 2006 Workshop on New Directions in Multilingual Information Access, Seattle, USA*, 2006.

[6] M. Damashek. Gauging Similarity with n-Grams: Language-Independent Categorization of Text. *Science*, 267(5199):843–848, 1995.

[7] M. S. Dryer. Prefixing versus suffixing in inflectional morphology. In B. Comrie, M. S. Dryer, D. Gil, and M. Haspelmath, editors, *World Atlas of Language Structures*, pages 110–113. Oxford University Press, 2005.

[8] T. Dunning. Statistical identification of language. Technical report, Techical Report MCCS-94-273, Computing Research Lab (CRL), New Mexico State University, 1994.

[9] G. Grefenstette. Comparing two language identification schemes. In S. Bolasco, L. Lebart, and A. Salem, editors, *The proceedings of 3rd International Conference on Statistical Analysis of Textual Data (JADT 95), Rome, Italy, Dec. 1995*, 1995.

[10] H. Hammarström. A naive theory of morphology and an algorithm for extraction. In R. Wicentowski and G. Kondrak, editors, *SIGPHON 2006: Eighth Meeting of the Proceedings of the ACL Special Interest Group on Computational Phonology, 8 June 2006, New York City, USA*, pages 79–88. Association for Computational Linguistics, 2006. `http://www.cs.chalmers.se/~harald2/sigphon06.pdf`.

[11] B. Hughes, T. Baldwin, S. Bird, J. Nicholson, and A. MacKinlay. Reconsidering language identification for written language resources. In *Proceedings 5th International Conference on Language Resources and Evaluation (LREC2006)*, pages 485–488. Genoa, Italy, 2006.

[12] P. Juola. Language identification, automatic. In K. Brown, editor, *Encyclopedia of Language and Linguistics*, volume 6, pages 508–510. Elsevier, Amsterdam, 2 edition, 2006.

[13] C. Kruengkrai, V. Srichaivattana, P. andSornlertlamvanich, and H. Isahara. Language identification based on string kernels. In *IEEE International Symposium on Communications and Information Technology, 2005. ISCIT 2005*, volume 2, pages 926–929, 2005.

[14] R. D. Lins and P. Gonçalves, Jr. Automatic language identification of written texts. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 1128–1133, New York, NY, USA, 2004. ACM Press.

[15] T. Martin, B. Baker, E. Wong, and S. Sridharan. A syllable-scale framework for language identification. *Computer Speech & Language*, 20(2-3):276–302, 2006.

[16] B. Martins and M. J. Silva. Language identification in web pages. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 764–768, New York, NY, USA, 2005. ACM Press.

[17] P. McNamee. Language identification: a solved problem suitable for undergraduate instruction. *Journal of Computing Sciences in Colleges*, 20(3):94–101, 2005.

[18] K. N. Murthy and G. B. Kumar. Language identification from small text samples. *Journal of Quantitative Linguistics*, 13(1):57–80, 2006.

[19] Y. K. Muthusamy and L. A. Spitz. Automatic language identification. In R. A. Cole, editor, *Survey of the State of the Art in Human Language Technology*, chapter 8.7. Center for Spoken Language Understanding CSLU, Carnegie Mellon University, Pittsburgh, PA, 1997.

[20] A. Poutsma. Applying monte carlo techniques to language identification. In T. Mariët, A. Nijholt, and H. Hondorp, editors, *Computational Linguistics in the Netherlands 2001: Selected Papers from the Twelfth CLIN Meeting*, volume 45 of *Language and Computers - Studies in Practical Linguistics*, pages 179–189. Rodopi, Amsterdam/New York, NY, 2002.

[21] J. M. Prager. Linguini: Language identification for multilingual documents. *Journal of Management Information Systems*, 16(3):71–102, 2000.

[22] P. Sibun and J. C. Reynar. Language identification: Examining the issues. In *5th Symposium on Document Analysis and Information Retrieval*, pages 125–135, Las Vegas, Nevada, U.S.A., 1996.

[23] H. Takci and I. Sogukpinar. Centroid-based language identification using letter feature set. In A. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing: 5th International Conference, CICLing 2004 Seoul, Korea, February 15-21, 2004 Proceedings*, volume 2945 of *Lecture Notes in Computer Science*, pages 640–648. Springer-Verlag, Berlin, 2004.

[24] A. Xafopoulos, C. Kotropoulos, G. Almpanidis, and I. Pitas. Language identification in web documents using discrete HMMs. *Pattern Recognition*, 37(3):583–594(12), 2004.

[25] D.-V. Ziegler. *The Automatic Identification of Languages Using Linguistic Recognition Signals.* PhD thesis, University of New York at Buffalo, 1991.

Terrier takes on the non-English Web

Craig Macdonald
University of Glasgow
Computing Science
Glasgow G12 8QQ, U.K.
craigm@dcs.gla.ac.uk

Christina Lioma
University of Glasgow
Computing Science
Glasgow G12 8QQ, U.K.
xristina@dcs.gla.ac.uk

Iadh Ounis
University of Glasgow
Computing Science
Glasgow G12 8QQ, U.K.
ounis@dcs.gla.ac.uk

ABSTRACT

The aim of this work is to identify how standard Information Retrieval (IR) techniques can be adapted in Web retrieval for non-English queries. In particular, we address the challenge of stemming queries and documents in a multilingual setting. Experiments with a multilingual collection of over 20 languages, more than 800 queries, and various stemming strategies in these languages reveal that using no stemming results in satisfactory Web retrieval performance, that is overall stable. Moreover, we show that language-specific stemming requires an accurate identification of the language of each query.

Categories and Subject Descriptors

H.4 [**Information Storage and retrieval**]: Information Search and Retrieval

General Terms

Measurement, Performance, Experimentation

Keywords

Web retrieval, Known-item retrieval, Multilingual retrieval, Language-specific stemming

1. INTRODUCTION

The field of Information Retrieval (IR) addresses the general problem of how to retrieve information, which is relevant to a user need, from a large repository of information, such as a collection of documents. Information in the document collection is represented in the form of an index, which contains statistics of term frequencies in each document and in the whole collection. Typically, using these statistics, term weighting models compute weights for individual terms, which capture the importance of the terms to the content of each document. A matching function then estimates the likely relevance of a document to a query, on the basis of these term weights, and the most relevant documents are identified and retrieved [26].

In brief, IR systems typically contain an *indexing* component, which stores a collection of information, and a *matching* component, which retrieves relevant information in response to a user query. This very basic architecture is typically enriched with a variety of retrieval-enhancing techniques, aiming to facilitate the system's efficiency and effectiveness. Examples of such techniques are removing stopwords or reducing variants of the same word to a single form (*stemming*). These IR system techniques were originally engineered for English collections of documents and queries.

Nowadays, it is reported that the majority of Web users are non-native English speakers. This means that most people wishing to retrieve information relevant to their need from the Web are likely to do so in a language other than English [4]. It is estimated that non-English queries and unclassifiable queries are not only numerous, but also that they grow increasingly bigger in number. This fact creates a problem for most search engines, which are typically optimised to process mainly English queries. For example, most search engines do not take full account of diacritics or the use of capitals in a user query. Such limitations in processing non-English queries make multilingual retrieval less effective [9]. Consequently, it is usually acknowledged that international search engines (like Yahoo! and Google) are less effective with monolingual non-English queries. In fact, Google has only very recently announced the upcoming launch of a cross-lingual functionality.

In this paper, we investigate how the Terrier retrieval platform [19] can deal with non-English queries. Terrier is a robust and modular IR engine, with an established track record of solid high performance for English retrieval [14, 15]. By testing it on non-English queries, we aim to identify whether standard IR techniques implemented in it are appropriate for non-English retrieval. Specifically, the IR technique we investigate is the application of appropriate stemming in a multilingual Web IR environment.

Stemming consists of reducing morphological variants of a word to a single form (or stem). This technique has been popular with IR systems, because it allows for different word forms to be represented under a single entry. For example, by stemming singular and plural forms of a word to a common form, the occurrence of that word in a document is represented more accurately, and hence retrieval performance and system efficiency improves [10].

Nevertheless, in a multilingual Web IR setting, stemming is not a straightforward process. Firstly, before stemming is applied, the language of the query/document needs to be known, so that the correct stemmer is used. Secondly, morphological complexity varies greatly per language, from the relatively simple (e.g. English), to the relatively more elaborate (e.g. Hungarian). This practically means that, whereas stemming might work for some languages, it might not work for others. Finally, as with other types of lan-

SIGIR '07 iNEWS07 workshop, July 27, 2007, Amsterdam, The Netherlands.

guage resources (e.g. part-of-speech taggers, named entity extractors, and so on), the availability of stemmers for many languages is sparse. In such cases, what is the best strategy: applying no stemming, or using stemmers designed for other languages?

These are the main issues we address in this paper. By doing so, we seek to gain insights into what is the most appropriate way for an IR system to process words in many languages, so that they are accurately indexed and efficiently matched to user queries.

The remainder of this paper is organised as follows. Section 2 gives an overview of studies relating to this work. Section 3 describes how we adapt Terrier to multilingual retrieval. Section 4 presents our experiments and discusses the experimental results. Section 5 concludes this paper with lessons learnt and opted future research directions.

2. RELATED STUDIES

The Web is an heterogeneous environment, in which information may appear in a great variety of different languages. The workshops on the evaluation of multilingual Web IR (WebCLEF) [4, 24] constitute an organised effort into looking at how Web IR systems can scale up to retrieval in a multilingual setting. These workshops have produced literature on a variety of techniques that can extend standard English IR systems to perform multilingual retrieval. One such reported technique is the extension of Web-based features (for example document structure) for retrieval in a multilingual setting [1, 8, 16, 17, 18, 25]. Another technique is applying language-specific stemming when retrieving documents in different languages [16, 17, 25]. An alternative to stemming in a multilingual environment is the use of character n-grams to represent the terms in the index [12]. Further techniques used with retrieval in different languages include normalising diacritics and accents [13]. Encoding issues, one of the biggest problems with non-English retrieval, have been dealt with either by adapting the retrieval system to process specific encodings, such as UTF-8 for example [16], or by transliterating characters into encodings that the system can process [13].

Overall, the above work draws an encouraging yet incomplete picture of multilingual Web IR: encouraging, because the community addresses the problem with organised efforts for standard evaluation. Incomplete, because these efforts reveal that technical difficulties, such as character encoding, are not yet overcome, while there is not a clear consensus on whether standard IR techniques, such as stemming, are beneficial to multilingual IR.

It is this last point that motivates the work in this paper: we address the technical difficulties in doing Web IR across languages by extending the modular Terrier platform, and we investigate the usability of stemming by experimenting with different combinations of stemmers and languages.

3. ADAPTING TERRIER TO MULTILINGUAL RETRIEVAL

In this section, we present how we adapt Terrier's functionalities for non-English retrieval. There are two main components in the overall architecture of the Terrier platform, namely *indexing* (described in Section 3.1), and *matching* (described in Section 3.2). *Indexing* describes the process during which Terrier parses a document collection and represents the information in the collection in the form of an index that contains statistics on term frequency in each document and in the whole collection. Term weights are generated for each term based on these statistics. *Retrieval* describes the process during which Terrier weights each document term and estimates the likely relevance of a document to a query, on the basis of these term weights.

In order to adapt Terrier into a multilingual environment, we focus on the application of appropriate stemming strategies. This technique is part of the system's indexing process, which is presented next.

3.1 Indexing

Indexing consists in parsing a document collection and *appropriately* indexing the information contained in it. In a multilingual setting, indexing collections in an *appropriate* way means being able to support retrieval in different languages, so that the IR system can accurately and uniquely represent each term in the corpus. To meet this requirement, we use a Terrier version that supports multiple character set encodings[1], ensuring that we have a robust representation of the collection.

Terrier achieves modularity in indexing collections of documents by splitting the process into four stages, where, at each stage, plugins can be added to alter the indexing process. The four stages of indexing with Terrier are [19]:

- handling a collection of documents,
- handling and parsing each individual document,
- processing terms from documents, and
- writing the index data structures.

During indexing, Terrier assigns to each term extracted from a document three fundamental properties, namely

- the actual string textual form of the term,
- the position of the term in the document, and
- the document fields in which the term occurs (fields can be arbitrarily defined by the document plugin, but typically relate to HTML/XML tags).

During indexing, the terms pass through a configurable 'Term Pipeline', which transforms them in various ways, using plugins such as stemming, removing stopwords in various languages, expanding acronyms, and so on. The outcome of the Term Pipeline is passed to the Indexer, which writes the data structures of the final index.

We adapt Terrier's indexing component as follows: during the parsing of the collection, we use heuristics to identify the correct character set encoding of each document. In particular, we examine the Content-Type HTTP header of the request, and any equivalent META tag in the header of the HTML document. If neither of these are found, then a default encoding is assumed based on the language of the document (as described below). For example a Czech document is likely to be encoded in ISO8859-2. Once the correct encoding for each document is determined, the collection

[1] The latest open source release of Terrier (version 1.1.0) supports various encodings of documents, and the use of non-Latin character sets. More details can be found at: `http://ir.dcs.gla.ac.uk/terrier/`

is parsed, each term being read and converted into UTF-8 representation. Hence, we ensure that terms from different languages encoded using different character sets are accurately represented in the index.

Terrier's modular architecture allows for any number of different stemmers to be easily applied at this stage. In particular, to determine the language of each document, we use the language identification tool TextCat [5], combined with evidence from the URL and the HTML of each document. For instance, if the identifier fails to identify the language of a document, then we can assume that documents from the .fr domain are likely to be in French. Alternatively, the `HTML` tag of an HTML document can have a `lang` attribute describing the language of the document. In this work, in addition to English stemming, we use several language-specific stemmers, appropriately selected using the language identification data. The application of stemmers is detailed in Sections 4.1 and 4.2.

Because in this paper we investigate the effect of different combinations of stemming upon multilingual retrieval performance, we create different indices of the collection, so that each index applies a different type of stemming strategy. This point is further detailed in Sections 4.1 and 4.2. Overall, we apply several stemming combinations to index the collection. This means that we create different indices of the collection. In each index, we keep field information for each term, so that we can identify which terms occur in which fields of the documents. This is motivated by the fact that, for Web IR, knowing where in a document terms occur may help retrieval performance [6]. In this work, we use different document fields when matching relevant documents to queries, as explained next.

3.2 Matching

So far we have seen how Terrier indexes a collection, so that terms in different languages are represented accurately, and how information on the location of the terms in the documents is also kept. This positional information for terms takes into account document structure in order to enhance retrieval performance. By document structure we denote specific document sections, also referred to as fields in the literature. It has been shown that using document fields can enhance retrieval performance in a Web IR setting [6, 16, 22]. The specific document fields we use in this work are the *body* of the document, the *title* of the document, and the *anchor text* information for a document (i.e. the text associated with the incoming links of a Web document).

We consider these different sources of evidence when matching a document to a query, using a weighting model that is specifically designed to combine term frequencies from different document fields. Specifically, we use the PL2F weighting model from the Divergence From Randomness (DFR) framework [2]. PL2F is a derivative of the PL2 model, which is specifically adapted to combine evidence from different fields. Using the PL2F model, the relevance score of a document d for a query Q is given by:

$$\begin{aligned} score(d,Q) \;&=\; \sum_{t\in Q} qtw \cdot \frac{1}{tfn+1}\big(tfn \cdot \log_2 \frac{tfn}{\lambda} \\ &\quad +(\lambda - tfn)\cdot \log_2 e + 0.5 \cdot \log_2(2\pi \cdot tfn)\big) \end{aligned} \qquad (1)$$

where λ is the mean and variance of a Poisson distribution, given by $\lambda = F/N$; F is the frequency of the query term t in the collection, and N is the number of documents in the whole collection. The query term weight qtw is given by qtf/qtf_{max}; qtf is the query term frequency. qtf_{max} is the maximum query term frequency among the query terms.

tfn corresponds to the weighted sum of the normalised term frequencies tf_f for each used field f, known as *Normalisation 2F* [16]:

$$tfn = \sum_f \left(w_f \cdot tf_f \cdot \log_2(1 + c_f \cdot \frac{avg\_l_f}{l_f}) \right), (c_f > 0) \quad (2)$$

where tf_f is the frequency of term t in field f of document d; l_f is the length in tokens of field f in document d, and avg_l_f is the average length of the field across all documents. The contribution of the field is controlled by the weight w_f; c_f is a hyper-parameter for each field, which can be set automatically [11], and which controls the term frequency normalisation. The c_f and w_f values used in this work are given in Section 4.1, along with the rest of the experimental settings.

4. EVALUATION

The aim of our experiments is to investigate whether the standard IR techniques implemented in Terrier are appropriate for non-English retrieval, with special focus on the use of stemming in a multilingual setting. Section 4.1 describes the datasets and resources used, while Section 4.2 describes how we organise our experiments. Experimental results are presented in Section 4.3.

4.1 Experimental Settings

We adapt Terrier for multilingual Web IR (as presented in Section 3), and we evaluate it on the *mixed monolingual* task from WebCLEF 2005 and 2006. The mixed monolingual task simulates a user searching for a known-item page in a European language. This task uses known-item topics, namely homepage finding and named page finding queries. The homepage topics are names of a site that the user wants to reach, and named page topics concern non-homepages that the user wants to reach. The mixed monolingual retrieval task is based on a stream of known-item topics in a range of languages.

The mixed-monolingual retrieval task uses the EuroGOV test collection [23], and more than 800 monolingual known-item topics in various languages.

EuroGOV consists of Web documents crawled from European governmental sites. As such, it is a multilingual Web corpus, containing 3.5 million pages from 27 primary domains, and covering over 20 languages. Specifically, EuroGOV contains documents from the following (top-level) domains:

```
at(=austria)            cy(=cyprus)
de(=germany)            ee(=estonia)
eu(=european union)     fr(=france)
hu(=hungary)            it(=italy)
lu(=luxemburg)          mt(=malta)
pl(=poland)             ru(=russia)
si(=slovenia)           uk(=united kingdom)
be(=belgium)            cz(=czech republic)
dk(=denmark)            es(=spain)
fi(=finland)            gr(=greece)
ie(=ireland)            lt(=lithuania)
lv(=latvia)             nl(=the netherlands)
```

```
pt(=portugal)       se(=sweden)
sk(=slovakia)
```

There is no single language that dominates the corpus, and its linguistic diversity provides a natural setting for multilingual Web search. Files in EuroGOV have the following format:

```
 <EuroGOV:bin
domain="" <!-- The top level domain -->
id=""> <!-- The name of the file -->
<EuroGOV:doc
url="" <!-- URL of the page -->
id="" <!-- DocID of the format Exx-yyy-z -->
<!-- E is E and stands for EuroGOV -->
<!-- xx is the top level domain -->
<!-- yyy is the file name -->
<!-- z is the character offset of the document
-->
md5="" <!-- MD5 checksum of the content of
the page -->
fetchDate="" <!-- Fetch date of the page -->
contentType=""> <!-- contentType as given by
the web server -->
<EuroGOV:content>
<![CDATA[
...  content ...  <!-- This is the actual page
-->
]]>
</EuroGOV:content>
</EuroGOV:doc>
...
</EuroGOV:bin>
```

The structure of documents in EuroGOV is clearly marked by the annotation shown above.

An example of the topic format used at WebCLEF 2005 is:

```
 <topic>
<num>WC0006<\num>
<title>Minister van buitenlandse zaken<\title>
<metadata>
<topicprofile>
<language language="NL"/>
<translation language="EN">
dutch minister of foreign affairs </translation>
</topicprofile>
<targetprofile>
<language language="NL"/>
<domain domain="nl"/>
</targeprofile>
<userprofile>
<native language="IS"/>
<active language="EN"/>
<active language="DA"/>
<active language="NL"/>
<passive language="NO"/>
<passive language="SV"/>
<passive language="DE"/>
<passive_other>Faroese</passive_other>
<countryofbirth country="IS"/>
<countryofresidence country="NL"/>
</userprofile>
</metadata>
</topic>
```

The topics used in WebCLEF include a large amount of metadata, as can be seen above. Real-life user queries on the Web do not come with such a variety of metadata. In fact, they typically consist of very few keywords [20]. In order to simulate as much as we can real user queries, in our experiments we only use the title field of the topics.

There is a significant amount of queries available for the 2005 and 2006 mixed-monolingual task. Specifically, the 2005 topics contain 547 queries, consisting of 242 homepage finding queries, and 305 named page finding queries. These queries have been created manually by humans and target pages in 11 different languages: Spanish, English, Dutch, Portuguese, German, Hungarian, Danish, Russian, Greek, Icelandic, and French. The 2006 topics differ from the 2005 topics as follows: a great part of the 2006 topics has been created automatically, using Azzopardi and de Rijke's technique for automatically generating known-item topics [3]. The 2006 topic set also includes a number of manual (human-generated) topics. Specifically, there is a total of 1120 new topics for 2006, 817 of which are automatic, and 303 of which are manual. The 2006 manual queries cover only languages for which human expertise was available (Dutch, English, German, Hungarian, and Spanish) and are supplemented by including some of the queries from the 2005 topic set, while the 2006 automatic queries cover almost all languages. However, in this work, we consider only the manual queries, as the evaluation using the automatic queries did not correlate highly with the true performance of the IR systems as measured by the manual queries [4].

Section 3 presented how we extend Terrier's indexing component to take into account various stemmers, and how we match documents to queries using a field-based weighting model. Specifically, we apply the following stemmers:

- For English, we use Porter's English stemmer;
- For all other languages, we use their corresponding Snowball stemmer[2], with the exception of languages for which there was no stemmer available:
  - For Icelandic, we use the Danish Snowball stemmer; our reasoning is that Danish is 'linguistically' relatively close to Icelandic.
  - For Hungarian, we use Hunstem[3] as the Snowball stemmer for Hungarian was not available at the time of our experiments.

We do not remove stopwords during indexing, because we do not have stopword lists for all languages, and we do not wish to give an unfair advantage to some languages over others. For retrieval, we use the language topic metadata to select the appropriate stemmer and stopword list for that language. Moreover, we use the body, title, and anchor text[4] fields of documents, which we weight using the PL2F model (Section 3.2). The setting of the parameters c_f and field

[2] http://snowball.tartarus.org/

[3] http://magyarispell.sourceforge.net

[4] During indexing, anchor text from a document with a different language to the target document is stemmed using the stemmer of the language of the source document.

weights w_f presented in Section 3.2 is taken from [16], and is the following:

- $c = 4.10$ & $w = 1$ for the body of the document;
- $c = 100$ & $w = 40$ for the title and anchor text of the document.

Finally, we mentioned earlier that the WebCLEF topics are known-item topics, where a unique URL is targeted. This means that an early precision measure is more suitable to evaluate retrieval in this case. We use the metric also used in WebCLEF, namely the *mean reciprocal rank* (MRR). The reciprocal rank is calculated as 1 divided by the rank at which the (first) relevant page is found. The mean reciprocal rank is obtained by averaging the reciprocal ranks of a set of topics.

4.2 Experimental Methodology

We hypothesise that being able to apply the correct stemmer to a document and a topic can increase retrieval performance. To test this hypothesis, we create three indices of the EuroGOV collection:

1. we index the collection without applying stemming;
2. we index the collection by applying Porter's English stemmer to all documents, regardless of their domain and language;
3. we index the collection by applying stemming to each document according to the language of the document. The language of each document is determined by the language identification data provided by the TextCat utility described in Section 3.1.

We organise our experiments as follows:

- **NoStem**: retrieval without stemming the documents or the queries. This is our baseline.
- **PorStem**: retrieval using Porter's English stemmer for all documents and queries, regardless of their language. This run is a simple baseline showing the effects of applying an English-oriented IR system. For languages not in the Latin character set, Porter's stemming should have no effect.
- **AllStem**: retrieval using language-specific stemming, where the language of the query is defined by the topic-metadata.
- **SelStem**: retrieval using language-specific stemming, where the language of the query is guessed using the TextCat language identifier. When the language identifier fails to identify a language, no stemming is applied to the query and the the unstemmed index is used.

While the run **AllStem** is not realistic in the sense that users would likely not state the language of their query at submission time, it allows us to determine the extent to which the language identification of the queries adds noise to the **SelStem** run. In addition to the runs described above, we compare the system's retrieval performance on a per-language basis, so that we may distinguish between 'harder' and 'easier' languages. The next section details the findings of our experiments.

4.3 Experimental Results

Table 1 displays the retrieval performance of Terrier on the 2005 topic set. We display the MRR scores according to the topic language, the named-page (NP) and home-page (HP) topics, and for all topics in total (All). In Table 1 we observe the following:

- Applying no stemming is generally the most effective approach. This is the general conclusion for all languages. However, on a per-language basis, stemming helps retrieval for German.

- Applying Porter's English stemmer for all languages results in the most stable retrieval performance (the deviation in MRR across all topics is the smallest of all, σ=0.426). However, applying Porter's stemming to all languages significantly harms retrieval performance, yet less than using language-specific stemming. This is the general conclusion for all languages. On a per-language basis, language-specific stemming is better for Danish, German, and Greek. The particularly low performance when applying the correct stemmer to the Hungarian topics (**AllStem**) implies that the Hungarian stemmer is not effective.

- There exists a considerable amount of variation across languages. This point is also displayed graphically in Figure 1(a). This observation is consistent with the general trend observed in WebCLEF 2005 [24], namely that some languages were hard for all systems. Specifically, in WebCLEF 2005, it was reported that most systems scored relatively high for Dutch, relatively low for Russian and Greek, and close to average for German. We observe that Terrier is not only consistent with this, but also generally robust across different languages, including Russian.

- Named page runs score higher than home page runs. This is consistent with the general trend reported in WebCLEF 2005 [24], and also the English monolingual experiments of the Text REtrieval Conference (TREC)[5] for the Web track of 2003 and 2004 [6, 7].

- As expected, the selective application of stemming using the language identifier (**SelStem**) normally decreases in performance compared to the **AllStem** run. This happens when the inaccuracy of the language identifier has caused the wrong stemmer to be selected. For some languages the performance of **SelStem** is better than when the correct stemmer is used (**AllStem**); we suggest that this is mostly the case when the language identifier fails to guess a language, and in these cases the system used the unstemmed query with the unstemmed index was used (which has a better performance).

Table 2 displays the retrieval performance of Terrier on the 2006 topic set. From the table, we observe the following:

- Similarly to before, applying no stemming is the most effective approach, overall, and for both NP and HP tasks, as well as for most languages.

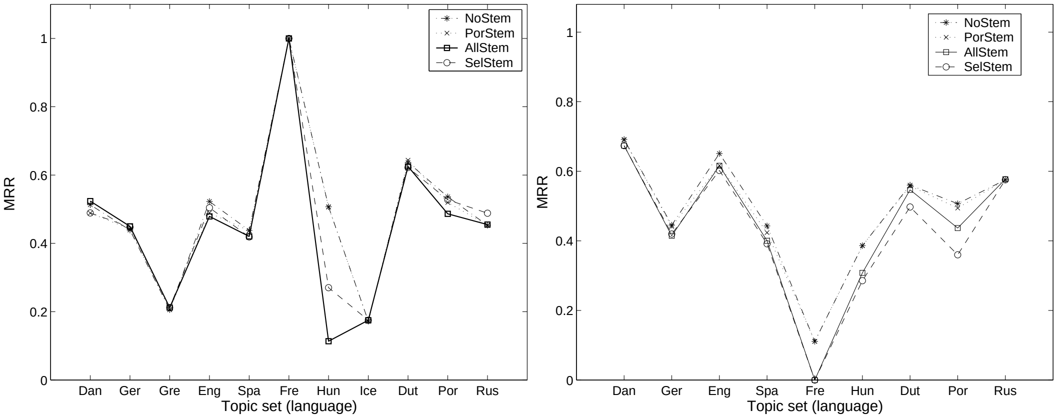
[5] `http://trec.nist.gov/`

Lang.	NoStem	PorStem	(Δ%)	AllStem	(Δ%)	SelStem	(Δ%)
Dan	0.5130	0.4886	(–4.8%)	0.5263	(+2.6%)	0.4891	(–4.7%)
Ger	0.4389	0.4421	(+0.7%)	0.4498	(+2.5%)	0.4476	(+2.0%)
Gre	0.2056	0.2056	(0.0%)	0.2119	(+3.1%)	0.2119	(+3.1%)
Eng	0.5226	0.4892	(–6.4%)	0.4789	(–8.4%)	0.5045	(–3.5%)
Spa	0.4381	0.4370	(–0.3%)	0.4203	(–4.1%)	0.4188	(–4.4%)
Fre	1.0000	1.0000	(0.0%)	1.0000	(0.0%)	1.0000	(0.0%)
Hun	0.5071	0.5062	(–0.2%)	0.1137	(–77.6%)	0.2702	(–46.7%)
Ice	0.1722	0.1722	(0.0%)	0.1750	(+1.6%)	0.1750	(+1.6%)
Dut	0.6371	0.6433	(+1.0%)	0.6251	(–1.9%)	0.6222	(–2.3%)
Por	0.5361	0.5197	(–3.1%)	0.4866	(–9.2%)	0.5277	(–0.2%)
Rus	0.4530	0.4530	(0.0%)	0.4549	(+0.4%)	0.4883	(+7.8%)
σ	0.429	0.426		0.428		0.430	
All NP	0.5142	0.4928	(–4.2%)	0.4630	(–10.0%)	0.4909	(–4.5%)
All HP	0.4597	0.4643	(+1.0%)	0.4254	(–7.5%)	0.4320	(–6.0%)
All	0.4900	0.4802**	(–2.0%)	0.4464**	(–8.9%)	0.4648**	(–5.1%)

Table 1: Mean Reciprocal Rank (MRR) of WebCLEF 2005 mixed monolingual runs. Statistically significant differences on All from the NoStem baseline (Wilcoxon Signed Rank Test) are denoted * and ** for ($p < 0.05$) and ($p < 0.01$) respectively. Lang. = topic language. (Δ%) = % diff. from NoStem. σ=st. deviation. NP = named page. HP = homepage.

Lang.	NoStem	PorStem	(Δ%)	AllStem	(Δ%)	SelStem	(Δ%)
Dan	0.6914	0.6901	(–0.2%)	0.6735	(–2.6%)	0.6735	(–2.6%)
Ger	0.4451	0.4415	(–0.8%)	0.4145	(–6.9%)	0.4196	(–5.7%)
Eng	0.6509	0.6167	(–5.3%)	0.6158	(–5.4%)	0.6024	(–7.5%)
Spa	0.4428	0.4237	(–4.3%)	0.4002	(–9.6%)	0.3916	(–11.6%)
Fre	0.1111	0.1111	(0.0%)	0.0000	(n/a)	0.0000	(n/a)
Hun	0.3862	0.3862	(0.0%)	0.3080	(–20.2%)	0.2855	(–26.1%)
Dut	0.5601	0.5573	(–0.5%)	0.5467	(–2.4%)	0.4974	(–11.2%)
Por	0.5068	0.4942	(–2.5%)	0.4367	(–13.8%)	0.3600	(–29.0%)
Rus	0.5755	0.5755	(0.0%)	0.5772	(+0.3%)	0.5755	(0%)
σ	0.423	0.418		0.425		0.425	
All	0.5150	0.5031*	(–2.3%)	0.4733**	(–8.1%)	0.4530**	(–12.0%)

Table 2: Mean Reciprocal Rank (MRR) of the WebCLEF 2006 mixed monolingual runs (manual topics). Statistically significant differences on All from the NoStem baseline (Wilcoxon Signed Rank Test) are denoted * and ** for ($p < 0.05$) and ($p < 0.01$) respectively. Lang. = topic language. (Δ%) = % diff. from NoStem. σ = st. deviation. n/a = non applicable.



(a) WebCLEF 2005 Topics

(b) WebCLEF 2006 (Manual) Topics

Figure 1: MRR per language with different stemming combinations for the WebCLEF 2005 and 2006 topics.

- Similarly to before, applying Porter's stemming gives the most stable retrieval performance throughout (smallest deviation among languages throughout), which is however not the best performance in terms of retrieval effectiveness.

- The considerable amount of variation across languages reported for the 2005 topics is observed here as well (see Figure 1(b)). This trend was also reported in WebCLEF 2006 [4], namely that some languages were hard for all systems.

- The **SelStem** run never outperforms the **AllStem** run for any language. This suggests that, unlike for the 2005 topics, the language identifier has failed to suggest a language for only a few queries, meaning that there has been insufficient fallback (cf **NoStem**) to increase the overall performance for those languages. This is confirmed by Table 3, which is described below.

Overall, we can summarise the observations drawn from Tables 1 and 2 as follows:

- In a multilingual Web IR environment, applying no stemming at all is generally the most effective approach. As predicted, applying Porter's English stemming to all languages results in a signficant decrease compared to applying no stemming. However, unexpectedly, applying Porter's English stemmer does achieve the most stable retrieval performance across both tasks. Applying language-specific stemming is neither the most stable, nor the most effective retrieval approach, and in particular, always results in a statistically significant degradation in overall MRR.

- In a realistic Web IR environment, the languages of each query are not available. However, using modern language identification tools to select an appropriate stemmer can affect the performance of a selective stemming system. In particular, Table 3 shows the accuracy and the number of unknowns generated by the language identification tool for the topic and documents respectively. While 94% accuracy is achievable for the language identification of the documents, due to the much shorter nature of the queries, only 50% accuracy is achieved in query language identification. This explains the difference in performance exhibited between the **AllStem** and **SelStem** runs in Tables 1 and 2.

This conclusion is not entirely generalisable, but subject to the quality of the stemming resources used. The different stemmers used for various languages are not necessarily of the same quality. For example, the performance of the Hungarian stemmer is not entirely satisfactory; the stemmer used for Icelandic is in fact designed to stem Danish. On the contrary, Porter's stemmer for English is a generally popular and well-established stemmer, the performance of which can be expected to be relatively reliable. More and better resources are needed in order to have a more accurate idea of whether language-specific stemming is indeed not beneficial for multilingual Web IR. Additionally, the accuracy of language-specific stemming is partly depicted by the extent to which the language of the queries can be identified, and hence we believe that it is in this area that future research should also be directed.

Language Identification				
	Accuracy		Unknown	
	2005	2006	2005	2006
Topics	55.9%	51.5%	43.3%	13.2%
Relevant Documents	94.4%	94.7%	2.5%	1.7%
All Documents	n/a	n/a	2.8%	

Table 3: Accuracy of the language identification for the language of the topics, and the language of the target documents of the topics. Unknown is the fraction that the classifier failed to suggest any languages. Note that there is only a language identification ground truth available for the relevant documents, not all documents in the collection.

WebCLEF Year	
2005	2006
0.5135	**0.5150**
0.4900	0.3145
0.4780	0.1396
0.2860	0.0923

Table 4: Terrier's best runs (bold) versus top 3 submitted runs for WebCLEF 2005 & 2006 (mixed monolingual task).

Finally, Table 4 displays the best MRR scores reported in our experiments next to the top three runs on the manual queries submitted to WebCLEF 2005 and 2006 from all participating groups. However, because these are the official submitted runs of participating groups, they all use more than baseline settings: for example, they make use of retrieval-enhancing techniques, such as some knowledge about the document URL, query expansion, Natural Language Processing (NLP) functionalities, and so on. In fact, the best scoring run for the manual runs of 2005 (MRR of 0.5135) uses the same retrieval system and weighting model on fields as our reported runs. Nevertheless, that run outperforms our equivalent run (MRR of 0.4900), because it uses URL evidence and acronym expansion, while we only use the baseline weighting model with document fields. Note that for the 2006 manual topics, our reported run obtains the best overall performance. Naturally, the retrieval performance reported here could be improved by using retrieval-enhancing techniques, such as the ones mentioned above, and by further optimising the system's settings.

5. CONCLUSIONS

We investigated whether the standard IR techniques implemented in Terrier are appropriate for non-English retrieval, with special focus on the use of stemming in a multilingual setting. The bare-system approach of applying no stemming at all is very effective, and in addition is a safe and stable option, where the results are significantly better than those produced by the best stemming approach for that language. It is not clear that stemming with respect to a language can assist retrieval performance, and in particular the performance of such is partly depicted by the accuracy of the language identifier tool used for the documents and the queries.

With regards to the retrieval platform used, we have shown how Terrier's modular configuration allows for some simple extensions that easily solve some well-noted technical problems in the field (e.g. character encoding). Experiments in a mixed monolingual environment show that the platform is thoroughly robust in dealing with queries in 11 European languages.

Future work includes using more realistic settings as well as more and better quality resources (e.g. non-English stemmers). Moreover, we will aim to adapt Terrier to non-European languages with different writing systems, such as Chinese or Japanese, where the tokenisation performed is much more important. In particular, the success of Terrier on retrieval in a Japanese content can be evaluated using collections from the NTCIR evaluation forum[6].

6. REFERENCES

[1] M. Adriani and R. Pandugita. Using the Web information structure for retrieving Web pages. In Peters et al. [21], pages 892–897.

[2] G. Amati. *Probabilistic Models for Information Retrieval based on Divergence from Randomness*. PhD thesis, University of Glasgow, Glasgow, 2003.

[3] L. Azzopardi and M. de Rijke. Automatic construction of known-item finding test beds. In *Proceedings of SIGIR 2006*, pages 603–604, 2006.

[4] K. Balog, L. Azzopardi, J. Kamps, and M. de Rijke. Overview of WebCLEF 2006. In A. Nardi, C. Peters, and J. Vicedo, editors, *Working Notes CLEF 2006*, 2006.

[5] W. B. Cavnar and J. M. Trenkle, N-Gram-Based Text Categorization. In Proceedings of SDAIR'94, pages 161–175, 1994.

[6] N. Craswell and D. Hawking. Overview of the TREC-2004 Web track. In *Proceedings of TREC-2004.*, 2005.

[7] N. Craswell, D. Hawking, R. Wilkinson, and M. Wu. Overview of the TREC 2003 Web track. In *Proceedings of TREC-2003*, 2004.

[8] C. G. Figuerola, J. L. A. Berrocal, Á. F. Z. Rodríguez, and E. R. V. de Aldana. Web page retrieval by combining evidence. In Peters et al. [21], pages 880–887.

[9] F. C. Gey, N. Kando, and C. Peters. Cross language information retrieval: a research roadmap. *SIGIR Forum*, 36(2):72–80, 2002.

[10] D. Harman. A failure analysis on the limitations of suffixing in an online environment. In *Proceedings of SIGIR 1987*, pages 102–108, 1987.

[11] B. He and I. Ounis. A study of the dirichlet priors for term frequency normalisation. In *Proceedings of SIGIR 2005*, pages 465–471, 2005.

[12] N. Jensen, R. Hackl, T. Mandl, and R. Strötgen. Web retrieval experiments with the EuroGOV corpus at the University of Hildesheim. In Peters et al. [21], pages 837–845.

[13] J. Kamps, M. de Rijke, and B. Sigurbjörnsson. Combination methods for crosslingual Web retrieval. In Peters et al. [21], pages 856–864.

[14] C. Lioma, C. Macdonald, B. He, V. Plachouras, and I. Ounis. University of Glasgow at TREC 2006: Experiments in Terabyte and Enterprise Tracks with Terrier. In *Proceedings of TREC-2006*, 2007.

[15] C. Macdonald, B. He, V. Plachouras, and I. Ounis. University of Glasgow at TREC 2005: Experiments in Terabyte and Enterprise Tracks with Terrier. In *Proceedings TREC-2005*, 2006.

[16] C. Macdonald, V. Plachouras, B. He, C. Lioma, and I. Ounis. University of Glasgow at WebCLEF 2005: Experiments in per-field normalisation and language specific stemming. In Peters et al. [21], pages 898–907.

[17] T. Martínez, E. Noguera, R. Muñoz, and F. Llopis. University of Alicante at the CLEF 2005 WebCLEF track. In Peters et al. [21], pages 865–868.

[18] Á. Martínez-González, J. L. Martínez-Fernández, C. de Pablo-Sánchez, and J. Villena-Román. MIRACLE at WebCLEF 2005: Combining Web specific and linguistic information. In Peters et al. [21], pages 869–872.

[19] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In *Proceedings of OSIR 2006*, 2006.

[20] S. Ozmutlu, A. Spink, and H. C. Ozmutlu. A day in the life of Web searching: an exploratory study. *Inf. Process. Manage.*, 40(2):319–345, 2004.

[21] C. Peters, F. C. Gey, J. Gonzalo, H. Müller, G. J. F. Jones, M. Kluck, B. Magnini, and M. de Rijke, editors. *Accessing Multilingual Information Repositories, 6th Workshop of the Cross-Language Evalution Forum, CLEF 2005, Vienna, Austria, 21-23 September, 2005, Revised Selected Papers*, volume 4022 of *Lecture Notes in Computer Science*. Springer, 2006.

[22] S. E. Robertson, H. Zaragoza, and M. J. Taylor. Simple BM25 extension to multiple weighted fields. In Proceedings of CIKM 2004, pages 42–49, 2004.

[23] B. Sigurbjörnsson, J. Kamps, and M. de Rijke. EuroGOV: Engineering a multilingual Web corpus. In Peters et al. [21], pages 825–836.

[24] B. Sigurbjörnsson, J. Kamps, and M. de Rijke. Overview of WebCLEF 2005. In Peters et al. [21], pages 810–824.

[25] S. Tomlinson. Danish and Greek Web search experiments with Hummingbird SearchServer$^{\mathrm{TM}}$ at CLEF 2005. In Peters et al. [21], pages 846–855.

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

[6] http://research.nii.ac.jp/ntcir/

Querying the Greek Web in Greeklish

Paraskevi Tzekou Sofia Stamou Nikos Zotos Lefteris Kozanidis
Computer Engineering Department, Patras University 26500 Greece
{tzekou, stamou, zotosn, kozanid} @ceid.upatras.gr

ABSTRACT

In this paper, we experimentally study the problem of querying the web in a hybrid language, namely Greeklish. Greeklish is the transliteration of Greek in Latin characters of the ASCII code. Although Greeklish emerged as a convenient mean for the creation and distribution of digital data at a time when Unicode Transformation Format was not supported for the Greek alphabet, nevertheless it is still being utilized as a matter of habit or need. Today, a considerable amount of the Greek web data contains pages written in Greeklish. Although, these are less *official* web pages and they appear mainly in blogs or forums, their contents may be of good quality and usefulness to the Greek online information seekers. However, the paradox of searching the Greek web is that search engines perceive Greeklish as a totally different language form Greek and as such they do not return Greek pages in response to Greeklish queries. As a consequence, users who issue Greeklish queries (sometimes for technical reasons) are systematically deprived of information that would otherwise be valuable to their search intentions. In an analogous manner, searching the web via Greek queries excludes from the search results pages of valuable content simply because they are written in Greeklish. In this paper, we study the phenomenon of Greeklish web searches and we propose a model that treats Greek and Greeklish web data in a uniform manner. Our aim is to improve the usability of Greek search engines and ameliorate the user experience, regardless of the preferred query alphabet.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: search process, query formulation; H.3.1 [**Content Analysis and Indexing**]: linguistic processing.

General Terms

Measurement, Performance, Experimentation.

Keywords

Greeklish, query transliteration, web search.

1. INTRODUCTION

The most prominent way for finding information on the web is go to a search engine, submit keyword queries that describe an information need and receive a list of results that satisfy the information sought. Although English is the lingua franca of the web, the majority of the web users are non-English speakers[1]. As the size of the non-English speaking online population grows rapidly, and the amount of non-English web data increases, it is increasingly important to support web searches in languages other than English. In this direction, there have been previous studies that investigate the problem of searching the web through non-English queries (cf. to [4] for a recent overview). The striking majority of existing studies concentrate on either searching the web in a particular natural language (other than English) [18] [12] [13] [6] [15], or on multilingual web information retrieval [16] [9] [7]. However, one aspect that none of the reported studies addresses is the phenomenon of querying the web via transliterated queries. Transliteration, as defined in Wikipedia, is:

> "*the practice of transcribing a word or text written in one writing system into another writing system.*"

In this paper, we investigate the problem of searching the Greek web using a hybrid language, namely Greeklish. Greeklish, a blend of the words Greek and English, is the representation of Greek textual data with the Latin script. The use of Greeklish as a means of writing Greek via the Latin alphabet dates back to the 17th century, when Greek merchandisers living abroad used the Latin script in their writings to communicate with other expatriate Greeks. For a thorough understanding in the history of Greeklish, we refer the interested reader to the works of [2] and [11].

Greeklish became widely known in the 1990's because of the spread of computer-mediated communication across the Greek society. In the digital era, Greeklish revived as a convenient mean for verbalizing Greek, since not all operating systems and applications back then had support for Greek. Today, modern software supports Greek but still it is much easier for Greek computer literates to e-write in Greeklish because it is faster to type and they do not have to worry for orthography issues. Moreover, Greeklish is being used for practical reasons since some people might not have access to the Greek character set. For instance it is impossible to send an SMS text message from a web-based interface using Greek characters to a cell phone. Despite the long official debates on whether Greeklish is threatening the cultural integrity of the Greek language, and letting aside the recent (2004) movement against Greeklish, the current literacy practices in Greek cyberspace demonstrate that users still communicate, search, write and receive information in Greeklish.

When it comes to the web searching paradigm, Greeklish imposes a number of challenges to the search engine community, which to the best of our knowledge have not been formally addressed inso-

SIGIR'07 iNEWS workshop, July 27 2007, Amsterdam, Netherlands

[1] According to the data provided by Global Reach, nearly 64.8% of the web users are non-English speakers.

far. One challenge is to understand the users' search behavior when querying the Greek web through Greeklish queries. That is, to find out whether there is any purposeful reason for issuing Greeklish queries, besides that of convenience and practicality (such as the lack of Greek fonts). Another challenge is to study whether Greeklish queries aim at the retrieval of data written in Greeklish only, or do they aim at locating data written in both Greeklish and Greek? Yet a more stimulating challenge is to investigate whether it would be useful for Greek web users to equip search engines with applications that automatically convert Greeklish pages and queries to Greek, in order to support global searches on the Greek web, regardless of the alphabet in use. In such case, a number of modifications would be required at the search engines' indexing modules, in order to be able to maintain stored pages in both their original and transliterated scripts. Moreover, there should be modifications at the engines' ranking functions in order to account for the Greeklish web data while ordering search results. Finally, the engines' query processing modules should integrate a Greeklish to Greek converter for automatically transcribing a query of the Latin alphabet into the Greek alphabet.

In this paper, we address the above challenges and we try to plug in the missing information about the impact that Greeklish may have on the effectiveness of Greek web searching. In particular, we experimentally study the difference between searching in Greek and searching in Greeklish, in terms of text-based retrieval performance. In the course of our study, we have developed a Greeklish-to-Greek translator that converts the contents of Greeklish pages into Greek. Moreover, our system converts Greeklish queries into Greek, so as to enable searching in the Greek web space via transliterated queries. We applied our translator to a number of experimental queries that we issued to Google Greece[2] search engine that indexes pages in both Greek and Greeklish and we evaluated the relevance of the returned results. We also carried out a user survey where we study how Greek users select the alphabet of their queries and how their selections exemplify their search pursuits and influence their search experiences. Obtained results demonstrate that there exist several diversifications between the Greek and the Greeklish web data, which inevitably influence retrieval performance. Moreover, our findings indicate that users have different goals in mind when searching in Greeklish compared to searching in Greek. In this respect, the use of Greeklish queries could serve as a useful guide while trying to predict the users' search goals.

The remainder of the paper is organized as follows. We start our discussion with a brief introduction to the Greeklish writing system and we present our approach towards making search engines understand Greeklish. In Section 3, we describe our experimental study and the dataset that we used. Experimental results are presented in Section 4. We conclude the paper in Section 5.

2. UNDERSTANDING GREEKLISH

Greeklish is not a language, but rather an alternative way of writing Greek using non-Greek fonts. For example, the sentence *καμία ερώτηση δεν έμεινε αναπάντητη* (no question was left unanswered) would transliterate in the Latin script as *kamia erotisi den emine anapantiti*. But, this is not the only way of transliterating /transcribing[3] the Greek characters of the sentence into Latin ones. Another way of writing our example sentence would be *kamia erwthsh den emeine anapanthth*.

As our example demonstrates, Greeklish is characterized by spelling variation in which the characters of the Greek alphabet may be transliterated with more that one Latin equivalents. These transliterations can be of two general types, namely orthographic and phonetic [1]. In orthographic transliterations the Greek orthography is generally reproduced in Latin characters as the transliterated terms *erwthsh*, *emeine* and *anapanthth* indicate in our second Greeklish example sentence. Conversely, in phonetic transliterations there is not a one to one mapping between Greek and Latin letters, but rather the pursuit is to phonetically transcribe Greek words with Latin characters, as the terms *erotisi*, *emine* and *anapantiti* in our first Greeklish example sentence illustrate. 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. Recently, it has been attested [2] [21] that the different Greeklish writing styles might be attributed to several factors besides phonetic and visual ones such as psychological, educational or geographical factors.

2.1 Unraveling the Greeklish Web

A fraction of the textual data that is available on the Greek web is written in Greeklish. Although many consider the use of Greeklish in web sites as an indication of the site operators' lacking knowledge of the language, nevertheless Greeklish persist mainly due to technical and ergonomic reasons. With respect to technical issues, Greeklish is a suitable vehicle for getting the message through when the Greek characters are not supported by a system or an Internet Service provider. On the other hand, ergonomic reasons imply that the additional burden of switching between the keyboard settings when writing foreign words in Greek texts is not worth the effort of the user who wants to write fast and communicate instantly.

Based on the above, it is not surprising that Greeklish is endorsed by the online population for social and international communication. Currently, there exist several web sites whose purpose is to enable people communicate in an instant and interactive manner. Most of these sites have a more social than professional character and include blogs, forums, chat rooms, message boards, etc. The wealth of the data stored in such sites is primarily textual and might be of great importance to web users who are interested in finding information about other peoples' comments, experiences and perspectives on a particular subject. With the current growth of the web as a part of our commercial life and the flourishing e-market of online goods, it is more demanding than ever to enable instant access to other peoples' shared viewpoints, opinions and recommendations, through the use of search engines.

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

[3] If the relations between letters and sounds are similar in two languages, a transliteration may be (almost) the same as transcription. Greeklish is the only writing system that mixes transliteration and transcription.

Unfortunately, not all Greek search engines index blogs or forums and those that do, they never return Greeklish pages in response to Greek queries. Nevertheless, besides social sites, there exist quite a few academic (i.e. university) sites that release part of their content in Greeklish. Given the dual nature of the Greek web's script, search engines perceive Greeklish as a totally distinct language from Greek. Therefore Greek pages are retrieved for Greek queries only, and Greeklish pages are returned for Greeklish queries only. But, a search engine indented for a large audience should treat all pages in the Greek web space uniformly regardless of the script in use, and it should never neglect the potential information gain of the users who have global access to the information that exists on the Greek web.

To enable search engine users get the gist of the information that is available on the Greek web, we need to design a sound model that not only manages to download, index and retrieve pages written in Greeklish, but which is also capable of interpreting Greeklish efficiently. By interpretation, we mean that a search engine should be able to understand the subject of a Greeklish page, the degree with which it relates to a given query and the page's importance on the Greek web. Likewise, the engine needs to understand Greeklish queries in order to answer them successfully. Most importantly, the engine should not discriminate between Greek and Greeklish data in the results returned for some query, unless it is otherwise specified by the user.

Given the lack of a standard transliteration for Greeklish, it is extremely difficult to automatically process Greeklish data. Because of that, search engines either prefer not to waste resources for indexing Greeklish pages or they index Greeklish pages but solely retrieve them in response to Greeklish queries by employing string matching techniques. Evidently, in both cases, search engine users are systematically deprived of either the Greeklish or the Greek web data, depending on their preferred query alphabet.

One approach towards enabling the uniform retrieval of the data that is available on the Greek web regardless of the script or writing style is to cast the problem of Greeklish web data processing as a translation problem. That is, to translate Greeklish web pages and queries in Greek and thereafter employ traditional text indexing and retrieval methods for enabling their exploration by the search engine users. The availability of a Greeklish-to-Greek translator would not only facilitate the retrieval of Greek pages through the use of Greeklish queries, but it would also enable the reverse approach, i.e. the retrieval of Greeklish data in response to Greek queries. The latter could be achieved by mapping Greek queries to the translated Greeklish pages and upon the identification of query matching pages, return the latter to the user either in their original (Greeklish) or in their translated (Greek) writing.

To fill this void, we have developed a Greeklish-to-Greek converter that we applied to a number of searches against Google Greece search engine and we experimentally evaluate the impact that the conflation of Greek and Greeklish online data has on retrieval performance. Our goal is to assist Greek web users locate accurate, valuable and interesting information while interacting with search engines. Next, we present our approach towards conflating Greek and Greeklish data at the search engine level.

2.2 Translating Greeklish

The problem of transcribing Greeklish to Greek is not new. Currently, there exist quite a few converters [20] [8] that cope with some Greeklish transliteration patterns and can be either accessed online or downloaded from the web. Moreover there exist some Greek to Greeklish translators [19] [3] that convert Greek characters into Latin ones. Although Greek-to-Greeklish translation is quite straightforward and it can be effectively achieved via a one to one character mapping, the translation of Greeklish to Greek is much more complicated, essentially due to the inconsistency in the Greeklish writing styles. Most of existing Greeklish-to-Greek translators rely on a predefined fixed set of transliteration rules, which simply replace every Latin character with a suitable Greek one. Few of the existing translators utilize regular expressions [10] in order to cope with context-dependent patterns. Currently, the most successful Greeklish translator is the "All Greek to me!" system [5] that automatically transliterates any type of Greeklish. "All Greek to me!" is the first translator to use a set of transliteration rules together with a lexicon, a speller and a language identification module. However, the translator is not freely available and it is a stand alone tool that cannot be readily integrated into a third party application.

Given the lack of an open Greeklish-to-Greek translator that could be easily integrated in a web search engine, we decided to build our own translator for conducting our study on the Greeklish web searches. Although the process for building the translator goes beyond the scope of this work, we briefly present the basic modules that our tool incorporates and we describe how it can be employed in the context of web searching.

Our Greeklish-to-Greek translator incorporates a set of transliteration rules that have been manually determined based on a number of writing patterns that we have extracted from a Greeklish web corpus of nearly 800K words. Given an input Greeklish text, our translator firstly performs all possible conversions of the Latinized terms into their corresponding Greek script. Thereafter, it checks the derived terms against a morphological lexicon of nearly 1,000,000 distinct wordforms [14]. The lexicon entries are organized in an inverted trie structure in order to facilitate dynamic dictionary string matching. Based on the lexicon data files, our translator improves malformed characters and retains only valid transcriptions, i.e. terms identified in the lexicon. Terms not recognized as valid Greek terms in the lexicon are given as input to a spell-checker, which corrects orthographic, intonation and typing errors. Correctly spelled Greek words together with valid transcriptions are utilized for deriving the Greek translation of the input Greeklish text. The remaining terms that cannot be recognized by any of our modules are stored in a separate list which is manually examined by the translator expert. Figure 1, illustrates the overall architecture of our Greeklish-to-Greek translator.

In a similar but much more simplified manner we have developed a Greek to Greeklish translation module, which transcribes Greek text in the Latin script. Having presented our translation module, we now turn our attention to the way in which this could be fruitfully explored in web search applications.

Rendering a search engine with some level of understanding on the correspondence between Greeklish and Greek basically entails the integration of translation services in both the engine's indexing and query processing modules. With respect to Greeklish indexing, one approach might be to parse the Greeklish pages, remove markup and tokenize the pages' textual content. Thereafter, use the pages' Latinized word tokens as input to our Greeklish-to-Greek translator, which will convert them into their corresponding Greek words. Following translation, one might employ traditional indexing techniques to represent the pages' content at the index level. Note however that indexed terms should be maintained in both their Greek and Greeklish representations so as to enable the pages' retrieval through any of the two query languages. For the Greeklish representation of the indexing terms it would be preferable to use not only the writing style of Greeklish in which the terms appear in the pages, but also to use all their possible (or at least common) variations. To enable that, it would be useful to employ a converter, so as to transcribe the indexing terms into all their possible Greeklish variants.

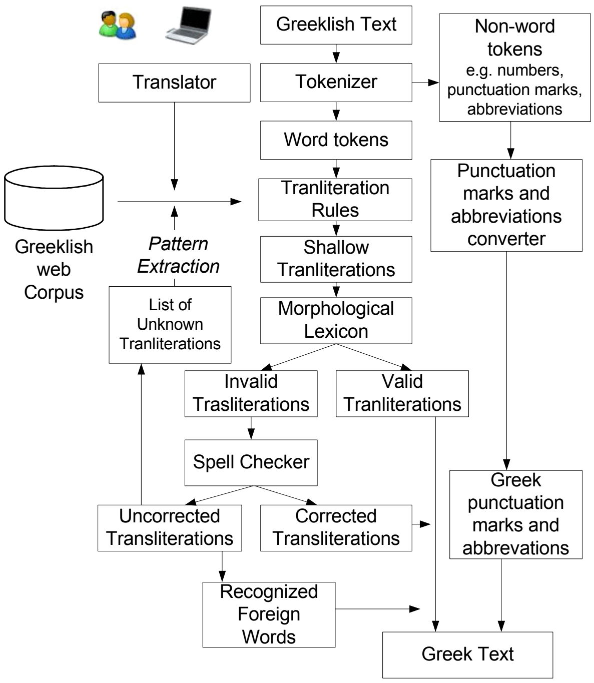


Figure 1. Architecture of the Greeklish-to-Greek translator.

Following the above process, we can represent every indexed Greeklish page as a set of keywords, both Greeklish and Greek, so as to ensure that the underline page will be retrieved in response to a keyword query, irrespectively of the alphabet or the writing style adopted by the user. Queries can be treated in a similar manner and searched against the engine's conflated index. More specifically, Greeklish queries should be converted into their Greek equivalents and Greek queries into all their possible Greeklish variants. To some extend, this approach might be perceived as a query expansion technique, in which all possible alphabetic variants of a query word participate in the search process.

The approach described above can be applied to every page on the Greek web so as to ensure that the search engines will be capable of capturing the complete picture of the Greek web's content. It is important to note that a pre-requisite step that the engine's modules need to take before initializing the translation process it to accurately identify the *language* of the page. To that end, we suggest the utilization of a language identification module that would be able to recognize Greeklish as a potential language in which web pages are written.

Following the translation and the processing of the Greeklish web data as given above, we can easily improve the engine's ability in interpreting both the relevance and the importance of a Greeklish web page in response to some query. In particular, we can explore the translated page's content terms against a semantic resource or a lexical ontology in order to automatically derive the page's topical category. Moreover, we can explore the translated page's content in order to compute the degree with which it relates to a particular query. Query-page relevance estimations may be either statistical or semantic driven, depending on the query matching algorithms the engine employs. Finally, the query-page correlation values could be fruitfully utilized for ranking the pages retrieved for some query. Next we describe how our proposed modules can be applied while searching the Greek web and we experimentally demonstrate the impact that the conflation of Greek and Greeklish might have on retrieval performance.

3. EXPERIMENTAL FRAMEWORK

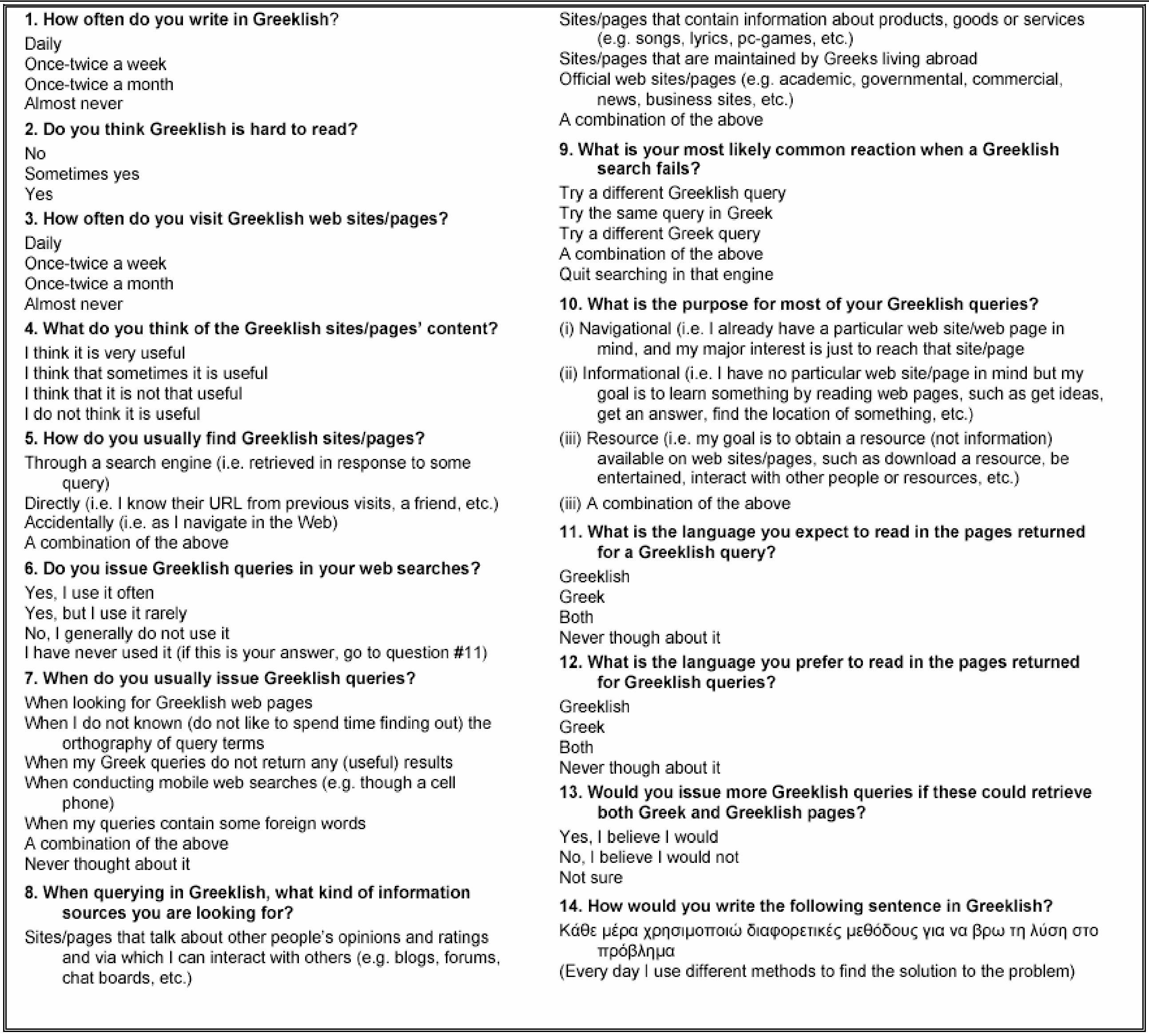
To evaluate the impact that the conflation of Greeklish and Greek online content might have on search engines' retrieval performance, we carried out two distinct, yet complementary, experimental studies. In one experiment, we conducted a user survey in order to collect data about the query patterns of the Greek web users. In particular, we examined the frequency with which Greek users issue Greeklish queries, the search goals hidden behind such queries and the users' perception on the usefulness of the Greeklish web pages. In our second experiment, we applied our Greeklish-to-Greek translator to a number of web searches that we performed to Google Greece search engine that indexes both Greek and Greeklish data, and we compared the performance of our mixed Greeklish and Greek queries in delivering relevant results to the performance of Greek-only and Greeklish-only queries. We start our discussion with the description of our experimental studies and we discuss obtained results in Section 4.

3.1 User Goals in Greeklish Queries

To study the users' search goals and expectations associated with issuing Greeklish queries, we carried out a human survey in which we examined the reasons why people query the Greek web through Greeklish queries, the kind of data that they wish to obtain, the perceived quality of the Greeklish web pages and what in the users' opinion could improve Greeklish web searches. In our survey, we recruited 42 graduate students in our department, with high levels of computer literacy and familiarity with Greeklish and we asked them to fill in a questionnaire that we designed for our study on Greeklish web searches. We decided to limit our survey to computer science graduate students mainly because of their ease of access and their proficiency in searching the web. However, we believe that this restriction does not introduce a significant bias in our results, because our experimental queries (presented next) are also collected from the same department and users. All our study participants had support for Greek characters in their workstations. The exact questions that we presented to our survey subjects are given in Table 1.

While distributing the questionnaire to our participants, we notified them that the purpose of our study was to investigate how users perceive the Greeklish web through both the queries they issue and the pages they visit. Our subjects were given ample of time for completing the questionnaire, but it generally took less than half an hour until all our participants delivered their answers.

As a final note, our participants volunteered to complete the questionnaire and they were encouraged to ask for clarifications in case they could not fully understand a particular question. Before reporting our survey results, we proceed with the description of our second experiment where we evaluated the effectiveness of blending Greeklish and Greek in retrieval performance.



1. How often do you write in Greeklish?
Daily
Once-twice a week
Once-twice a month
Almost never

2. Do you think Greeklish is hard to read?
No
Sometimes yes
Yes

3. How often do you visit Greeklish web sites/pages?
Daily
Once-twice a week
Once-twice a month
Almost never

4. What do you think of the Greeklish sites/pages' content?
I think it is very useful
I think that sometimes it is useful
I think that it is not that useful
I do not think it is useful

5. How do you usually find Greeklish sites/pages?
Through a search engine (i.e. retrieved in response to some query)
Directly (i.e. I know their URL from previous visits, a friend, etc.)
Accidentally (i.e. as I navigate in the Web)
A combination of the above

6. Do you issue Greeklish queries in your web searches?
Yes, I use it often
Yes, but I use it rarely
No, I generally do not use it
I have never used it (if this is your answer, go to question #11)

7. When do you usually issue Greeklish queries?
When looking for Greeklish web pages
When I do not known (do not like to spend time finding out) the orthography of query terms
When my Greek queries do not return any (useful) results
When conducting mobile web searches (e.g. though a cell phone)
When my queries contain some foreign words
A combination of the above
Never thought about it

8. When querying in Greeklish, what kind of information sources you are looking for?
Sites/pages that talk about other people's opinions and ratings and via which I can interact with others (e.g. blogs, forums, chat boards, etc.)
Sites/pages that contain information about products, goods or services (e.g. songs, lyrics, pc-games, etc.)
Sites/pages that are maintained by Greeks living abroad
Official web sites/pages (e.g. academic, governmental, commercial, news, business sites, etc.)
A combination of the above

9. What is your most likely common reaction when a Greeklish search fails?
Try a different Greeklish query
Try the same query in Greek
Try a different Greek query
A combination of the above
Quit searching in that engine

10. What is the purpose for most of your Greeklish queries?
(i) Navigational (i.e. I already have a particular web site/web page in mind, and my major interest is just to reach that site/page
(ii) Informational (i.e. I have no particular web site/page in mind but my goal is to learn something by reading web pages, such as get ideas, get an answer, find the location of something, etc.)
(iii) Resource (i.e. my goal is to obtain a resource (not information) available on web sites/pages, such as download a resource, be entertained, interact with other people or resources, etc.)
(iii) A combination of the above

11. What is the language you expect to read in the pages returned for a Greeklish query?
Greeklish
Greek
Both
Never though about it

12. What is the language you prefer to read in the pages returned for Greeklish queries?
Greeklish
Greek
Both
Never though about it

13. Would you issue more Greeklish queries if these could retrieve both Greek and Greeklish pages?
Yes, I believe I would
No, I believe I would not
Not sure

14. How would you write the following sentence in Greeklish?
Κάθε μέρα χρησιμοποιώ διαφορετικές μεθόδους για να βρω τη λύση στο πρόβλημα
(Every day I use different methods to find the solution to the problem)

Table 1. The questions distributed to our study participants.

3.2 Greeklish Web Information Retrieval

To measure the impact that the conflation of Greek and Greeklish data might have on web retrieval performance, we carried out an experimental study, in which we issued a number of queries in both Greek and Greeklish to Google Greece search engine and we evaluated obtained results.

To collect our experimental queries we asked from each of our study participants to specify a query that mimics a search they had performed earlier that day. For each of the queries, we asked our participants to write it down in both Greek and Greeklish and indicate which of the two writings they had used in their actual submission of the queries. Moreover, we asked them to indicate the search goal of their query by selecting one of the following[4]: (i) navigational, (ii) informational, and (iii) resource. Finally, we advised them to adopt their personal style of Greeklish writing for typing their queries in Greeklish. In total we collected a set of 42 queries, of which 31 were originally submitted in Greek and 11 were originally submitted in Greeklish.

We issued our experimental queries to Google Greece search engine, which indexes both Greek and Greeklish data. We submitted every query three different times: in the first submission every query was issued in Greek, in the second submission que-

[4] The taxonomy of query goals is the one proposed in [17].

ries were issued in Greeklish while in the third submission every query was typed in both Greek and Greeklish. For example, the query *databases* was written as *βάσεις δεδομένων* in its first submission (Greek), as *baseis dedomenwn* in its second submission (Greeklish) and as *βάσεις δεδομένων / baseis dedomenwn* in its third submission (both Greeklish and Greek). Note that the combined Greek and Greeklish queries (i.e. in their third submissions) are processed as Boolean OR queries, in the sense that the pages that are retrieved in their response might be written either in Greek or in Greeklish.

Before the actual submission of our queries, we processed them as follows. All Greek queries went through a spell-checker in order to ensure that they would contain only correctly spelled terms. Moreover, Greek queries passed through our Greek-to-Greeklish converter which returned for every transliterated query all its possible Greeklish variations. On average, for every Greek query our system returned 4.3 Greeklish transliterations. Finally, Greeklish queries were transcribed in Greek through the usage of our Greeklish-to-Greek translator. Thereafter, we submitted each of our experimental queries to the selected search engine three different times: (i) in Greek, (ii) in Greeklish (cf. all variations considered), and (iii) in both Greek and Greeklish. Out of the 42 experimental queries, 37 returned results in both their Greek and Greeklish submissions. Experimental evaluation concerns those 37 queries.

Following query issuing, we collected the first ten results returned for every query in each of the submissions and we asked our study participants to evaluate the results' relevance to the respective queries as follows. Each participant was shown the first ten results returned for her query across all the three query submissions. Retrieved results were displayed to our subjects in a random order. We then asked our participants to read each of the pages returned for every query and rate them using a four-point scale. Results' scoring indicates the degree to which the users perceive retrieval results to be relevant to their query intention and take values from 0, meaning that the result is irrelevant, to 3 meaning that the result is highly relevant.

Based on the users' relevance judgments, we computed the average relevance values of the top ten results delivered for a query across the three submissions, in order to evaluate the impact that the conflation of Greek and Greeklish have on retrieval performance compared to Greek-only and Greeklish-only information retrieval. Experimental results are discussed in Section 4.2.

4. EXPERIMENTAL RESULTS

4.1 Why Greeklish?

In this section, we present our human survey results, which help us improve our understanding in the users' search habits when querying the web in Greeklish. Due to space constraints, we do not graphically illustrate the distribution of the answers that our participants gave to every question. Nevertheless, we report percentage values for all the issues examined in our study. In particular, our results indicate that Greeklish is frequently used by our study participants (62.5% of our users write in Greeklish daily), although most of them (56%) sometimes find Greeklish hard to read. Moreover, 47.5% of our subjects visit Greeklish sites/ pages on a regular basis (i.e. more than once every week) and 65% of our users evaluate the content of Greeklish pages as generally useful. However, the fraction of our subjects' visits to Greeklish sites/pages through the use of search engines accounts to 20%, whereas a significant number of visits (37.5%) are accidental, in the sense that our users come across Greeklish pages as they navigate in the web.

When it comes to Greeklish web searches, 17.5% of our subjects use Greeklish often in their queries, while 37.5% use it sometimes, 37.5% do not generally use it and 7.5% have never tried Greeklish queries. Figure 2 illustrates the breakdown of the reasons behind issuing Greeklish queries as these are determined by our study participants who have used Greeklish in their queries. As we can see, a large number of people (40.5%) query the web in Greeklish as an alternative way of locating information in case their Greek searches fail to return useful data.

With respect to the information sources that people expect to obtain in response to Greeklish queries, 37.8% of our users indicated that they query in Greeklish when looking for sites/pages that contain information about products, goods and services, and 24.4% of them when looking for blogs, forums, chat boards, etc. An interesting finding is that none of our users prefer Greeklish queries to look for pages maintained by *official* sites or by people living abroad. This is in line with the responses that our subjects gave to Question 10 and which indicates that the most common search goal behind Greeklish queries is to obtain resources rather than reach to a particular page or find information on a topic of interest. Figure 3, depicts the distribution of search goals in Greeklish queries. Concerning the users' reaction when a Greeklish query fails to return any useful results, a surprising observation is that 29.8% of our subjects issue a different Greeklish query and 21.6% of them try the same query in Greek, as illustrated in Figure 4.

Another interesting finding is that 25% of our users expect to read Greek in the pages returned for Greeklish queries, while 27.5% of our subjects expect to read Greeklish and 32.5% expect to read both Greek and Greeklish in the pages retrieved for Greeklish requests. This is a quite interesting result that merits further investigation before we can justify the grounds of the participants' answers and before we realize whether the expectation for Greek content in the results of Greeklish queries is attributed to the nature of the queries, (e.g. names of products, proper names) or to the nature of the pages (e.g. pages that blend Greek and Greeklish content)

What is interesting though is that most of the users (67.5%) would like to read Greek in the pages returned for Greeklish queries. If search engines could support the retrieval of Greek content in the results delivered for Greeklish queries, our subjects indicated that they would issue more Greeklish queries; 62.5% of our users gave a positive answer to Question 13 (Yes, I believe I would) and only 22.5% of them answered negatively (No, I believe I would not). This observation justifies the need for our work on Greeklish web searches and we hope that our findings will stimulate the interest of others in assisting Greek web users experience improved searches.

A secondary objective in our human survey was to examine the variety of transliterations exemplified in our users' Greeklish writings. For that, we included Question 14 in our questionnaire in order to obtain perceptible evidence on the different ways in which Greek words can be transliterated in Greeklish. An analysis of the obtained transliterations demonstrates that these can vary to orthographic, phonetic or mixed transliterations, where the latter conflate visual and phonetic transcription of terms.

Table 2 reports our study results on the different transliterations that our subjects projected to their answers in the last question that we gave them.

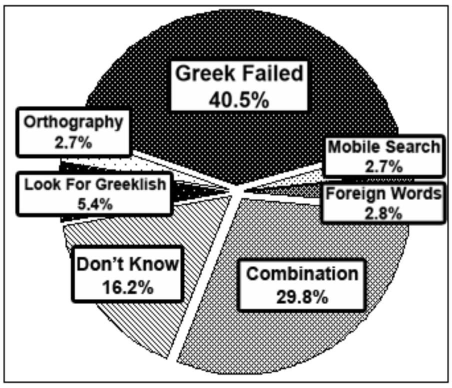


Figure 2. Why Greeklish queries?

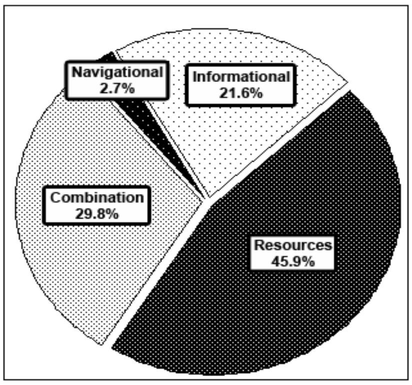


Figure 3. Goals of Greeklish queries.

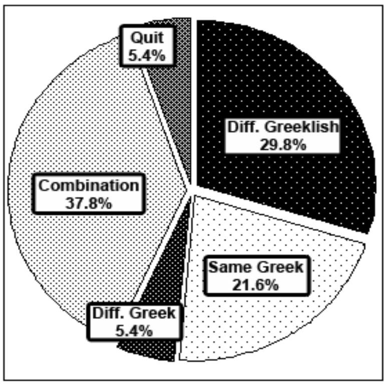


Figure 4. What if Greeklish fails?

Words	Number of transliterations	Type of transliterations		
		Orthographic	Phonetic	Mixed
Κάθε	2	40%	60%	
μέρα*	1	100%	100%	
χρησιμοποιώ	5	47.5%		52.5%
διαφορετικές*	1	100%	100%	
μεθόδους	2	35%	65%	
για*	1	100%	100%	
να*	1	100%	100%	
βρω	5	57.5%	7.5%	35%
τη	2	52.5%	447.5%	
λύση	6	50%	25%	25%
στο*+	2	100%	97.5%+	
πρόβλημα	5	37.5%	20%	42.5%

*the words with asterisks are actually transcribed as there is no variation between their orthographic and phonetic transliterations

*+ although the word in transcribed as "sto" one of our subjects replaced s with 6 as this resembles more the Greek letter 'σ'.

Table 2. Results on Greeklish transliterations.

A close look on the data reported in Table 2 demonstrates the great inconsistency in Greeklish writings as well as the frequent alterations between orthographic and phonetic transliterations. For instance the only difference between the orthographic and the phonetic transcriptions in the terms *κάθε* (every) and *μεθόδους* (methods) concerns the transliteration of the letter **θ** (theta). Although, 60% of our subjects selected a phonetic transliteration for representing **θ** in the first term, this percentage went up to 65% for the transliteration of **θ** in the second term. This practically implies the inconsistency in the personal writings of Greeklish, as the same user may switch between different transliterations in a single sentence.

Another noteworthy observation is that one of our subjects transliterated the term *βρω* (find) as *Bpw*, which is a more visual than strictly orthographic transcription. The above example indicates that people not only have their personal styles in writing Greeklish, but also that they try to make their Greeklish transliterations look as if written in the Greek alphabet, even when the latter is not utilized. This last conclusion is further supported in the transliteration of the term *πρόβλημα* (problem) for which one of our users transcribed the first letter ***π*** (pi) as two consecutive capitalized Latin T (i.e. TT).

Summarizing, the results obtained from our human survey verify that the number of people who prefer Greeklish in their web transactions is non-negligible and it is expected to grow as the commercial usage of the web increases. However, the vast majority of people prefer to read Greek in the obtained results, regardless of their selected query alphabet. In case such option was provided in today's search engines, they would probably issue more Greeklish queries. Lastly, given the remarkable variation in the Greeklish writing, we believe that a search engine capable of understanding Greeklish and the correlation it has to the Greek language would assist information seekers encounter successful web searches. The validity of our argument is experimentally supported in the findings of our second study, discussed next.

4.2 Greeklish Retrieval Performance

In this section we report on the results obtained in our second experimental study where we evaluated the effectiveness that the conflation of Greek and Greeklish alphabet has on retrieval performance. As discussed in Section 3.2 our evaluation was based on a set of 37 real queries that we submitted to Google Greece search engine. Query submissions followed a 3-step approach with a different query alphabet utilized in every step. In the first submission of the queries we used the Greek alphabet, in the second submission we used the Latin alphabet, while in the third submission we used both alphabets, simply by expanding Greeklish queries with their Greek transliterations and vice versa.

Experimental queries are classified into three groups depending on their underlying search goals as these have been determined by our study participants. The first group contains navigational queries such as "*Athens University of Economics and Business*". The second group contains informational queries such as "*mother's day*' and the third group contains resource queries such as "*map of Patras University*". Out of the 37 queries examined, 8 have been associated with a navigational goal, 24 have been associated with an informational goal, and the remaining 5 have been associated with a resource goal.

To evaluate the impact that the query alphabet has on retrieval performance, we relied on the relevance judgments that our study participants indicated for the first ten results retrieved for a query across each of the three query submissions. Figures 5, 6 and 7 show obtained results for resource, informational and navigational queries respectively. In the figures, the x-axis represents experimental queries and the y-axis shows the average relevance scores of the top 10 pages retrieved for every query in each of the submissions. For each query, the first bar represents the average relevance values of the top ten results retrieved for the Greek query, the second bar represents the average relevance scores of the top ten pages returned for the same query in its Greeklish submission, while the third bar represents the average relevance scores of the top ten results delivered for the conflated Greek and Greeklish query.

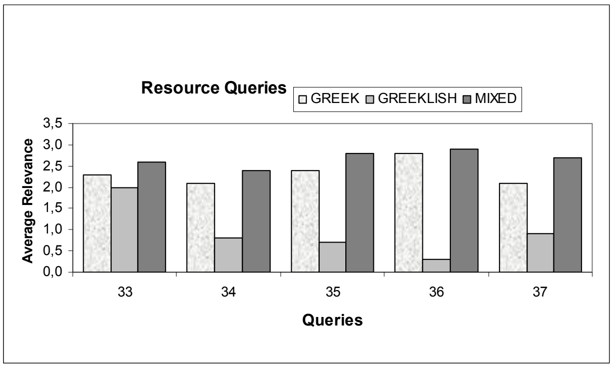


Figure 5. Average relevance of the top 10 results for our resource queries with respect to each of the query alphabets considered.

Results demonstrate that our mixed Greek and Greeklish search can successfully identify query relevant pages, especially when these pertain to resource requests. In particular, based on our the results of our human survey we found that a significant number of Greeklish queries intend to retrieve resources that the user will either download, interact with or save/print them for further utilization (cf. Figure 3).

For such search goals, expanding Greeklish queries with their Greek equivalents increases the likelihood that the resources sought will appear at the top positions in the results list. For instance, consider the case of the Greeklish query **Q36** *syntagh gia patsitsio* (pastitsio recipe) which retrieved results with an average perceived relevance of 0.3 at ranking point 10. A closer look at the first ten obtained results demonstrates that these mainly come from forums where people discuss about recipes, foods that they like, etc. Unfortunately, none of the top ten Greeklish pages contains a recipe for patsitsio, which is the information that the user was hopping to receive. Let's now turn our attention to the results retrieved for the same query following its expansion with the Greek terms. The overall relevance of the first ten pages returned for the expanded query is 2.9, while a closer look at the first few retrieved pages demonstrates that most of them concern pages written in Greek and which they do give a recipe.

Likewise, the first ten pages returned for the Greeklish query **Q35** *isotimia euro dollariou* (euro dollar exchange rate) have an average relevance of 0.7, as they mainly concern pages in forums that discuss users' opinions on the exchange rates. On the other hand, expanding the query by appending the Greek terms yields an average relevance of 2.8 at retrieval point 10 and results include pages in Greek such as the homepage of the Athens stock market as well as financial news articles that give exchange rates. In overall, as Figure 5 illustrates, the expansion of Greeklish queries with their Greek equivalents yields improved retrieval relevance for all our resource queries.

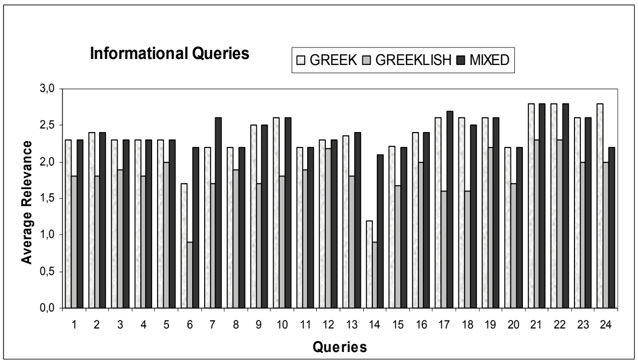


Figure 6. Average relevance of the top 10 results for our informational queries with respect to each of the query alphabets considered.

Conversely, for informational queries where the Greek alphabet is generally preferred, the results obtained for Greek requests generally outperform retrieval relevance for their Greeklish counterparts. In particular for 21 of the 24 informational queries examined, Greek retrieval delivered improved results compared to the results returned for their Greeklish transliterations.

Considering that our participants indicated that a large number of their Greeklish queries follow their unsuccessful Greek searches (cf. Figure 2), we may speculate that it would be useful to return the Greeklish transliterations of the queries together with the search results for the initial (Greek) query so that the user can utilize them in case she wishes to refine her search by adding terms to the initial query.

For instance, the average relevance of the first ten pages retrieved for the Greek query **Q6** *πρωτάθλημα μπάσκετ* (basket championship) has a value of 1.7 and with the first ten results containing pages about basket championships in elementary schools or local communities among others. Following the expansion of the Greek query with its Greeklish variants, the average relevance of the first ten pages goes up to 2.2, as results include also pages from forums and blogs, where people discuss about the games, comment on the teams' scores, etc. For this particular query, expansion yields increased retrieval relevance mainly because the Latin script of the term *μπάσκετ* (basket) is widely used in Greek writings.

Likewise, for the Greek query **Q14** *κάμερα κινητού* (cell phone camera), retrieval relevance has an average value of 1.2 and with the first page containing pictures taken from a cell phone. However, when the query is expanded with its Greeklish variants, average relevance goes up to 2.1 with most of the pages at ranking point 10 discussing cell phone models that incorporate a camera. Again the particularity of the query is that the term *κάμερα* (camera) is used in both Greek and Latin scripts in many Greek pages.

As our examples indicate, expanding Greek queries with their Greeklish transliterations can yield improved search results especially when the transliterations account to a common writing of a Greek term. This is especially true for technical terms most of which are primarily written in the Latin script. Therefore, we argue that recommending query transliterations as additional terms for improving a query can be beneficial to the user who might not consider Greeklish queries as an option for modifying her search.

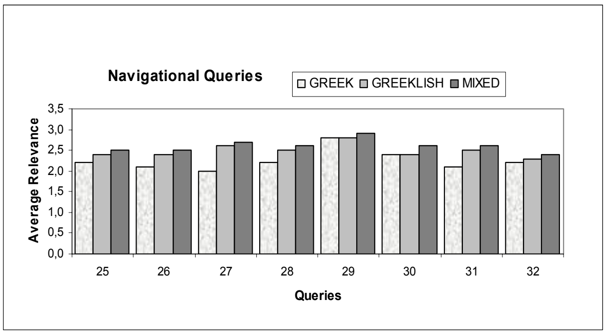


Figure 7. Average relevance of the top 10 results for our navigational queries with respect to each of the query alphabets considered.

Finally with respect to navigational queries, our results indicate that Greeklish can successfully retrieve the desired information especially when the term of the query appears in the URLs of the sought page. Given that page URLs use the Latin script it is reasonable to assume that Greeklish requests have good chances of detecting relevant pages for navigational queries. As our example query **Q27** *τα νέα* (the news) shows Greeklish search has an increased retrieval performance compared to Greek essentially because the query refers to the name of a popular Greek online newspaper that uses its name in the URL.

Based on our findings and considering that 67.5% of the users would like to see Greek pages in the results returned for Greeklish queries (on the provision that these relate to their information need) we may suggest that the conflation of both Greeklish and Greek has a significant potential in improving retrieval performance. Therefore, leaving the option of conflation or not the user can significantly improve the engine's usability and it will definitely assist users gain more control over their searchers, regardless of their preferred query alphabet.

Summarizing, our study is the first reported attempt to understand and evaluate the Greeklish web data from a search engine perspective. Our findings indicate that equipping search engines with mechanisms that can conflate Greek and Greeklish data in a single resource can be beneficial to the web users. We realize that such conflations would increase the computations required for translating the indexed pages from one alphabet to the other and that it would also entail additional storage capacity for maintaining translated pages at the index level, nevertheless it is worth the effort considering that the translation process is performed offline, while processing downloaded pages. Above all, the major goal of the search engine community is to assist users find the information sought in an effortless yet effective manner. With this goal in mind, we argue that Greek information seekers can benefit from the search engines' enhancement with mixed Greek and Greeklish search options.

5. CONCLUDING REMARKS

In this paper, we experimentally studied the phenomenon of querying the web in a hybrid language. In particular, we focused our study on searching the Greek web via Greeklish queries, i.e. Greek language queries that are written with the Latin script. Through a human subject study, we first showed that about 46% of our participants issue Greeklish queries when looking for web resources. This study further suggested that 40.5% of our subjects use Greeklish queries when their Greek searches fail to retrieve the desired information. Moreover, 67.5% of our study participants indicated that they would like to receive both Greek and Greeklish data in the results of their Greeklish queries. We then proposed the conflation of Greek and Greeklish data in the searches performed by Greek information seekers and we experimentally evaluated the impact that the blended Greek and Latin alphabet has on retrieval performance. Our evaluation showed that expanding Greeklish query terms with their Greek equivalents increases the relevance of the search results.

Although querying the web in a hybrid language is not a global phenomenon, nevertheless there exist quite a few writing systems that, either adopt the Latin alphabet for transcribing terms in orthographically complex languages, or they combine elements of different languages in one script. One example is Runglish, a neologism used to denote latinizations of the Cyrillic alphabet or mixing English and Russian grammatical structures. We may not know whether and how such *invented* amalgamate languages are employed when it comes to the web data; however we hope that our work will open up avenues for future research in the direction of both query-based and speech-based web searches.

Finally, our study on Greeklish web searches should be interpreted as neither an endorsement nor a rejection to the use of Greeklish. Rather, it should be perceived as the investigation of a phenomenon that influences peoples' interaction with search engines, a valuable tool for acquiring worldwide knowledge. Given the freedom that characterizes the nature of the web, people creating, using, interacting and searching the web should be given the freedom to choose their personal style of expressing their thoughts. Through our work, we are only giving them the tools to do that efficiently so as to help others benefit from it.

ACKNOWLEDGMENTS

Author Lefetris Kozanidis was funded by the 03ED_413 research project, implemented within the "Reinforcement Programme of Human Research Manpower" (PENED) and co-financed by National and Community Funds (25% from the Greek Ministry of Development-General Secretariat of Research and Technology and 75% from EU-European Social Fund).

REFERENCES

[1] Androutsopoulos J. Latin-Greek spelling in e-mail messages: usage and attitudes. (in Greek). *Studies in Greek Linguistics*, pp. 75-86, 1996.

[2] Androutsopoulos J. Greeklish: Transliteration practice and discourse in a setting of computer-mediated digraphia. *Standard Languages and Language Standards: Greek, Past and Present*, 2006.

[3] ASDA Greek-to-Greeklish converter: http://home.asda.gr /active/ GrLish2.asp

[4] Bar-Ilan J. & Gutman T. How do search engines respond to some non-English queries? *Journal of Information Science*, 31(1): 13-28, 2005.

[5] Chalamandaris A., Protopapas A., Tsiakoulis P. & Raptis S. All Greek to me! An automatic Greeklish to Greek transliteration system. In *Proceedings of the 5th Intl. Conference in Language Resources and Evaluation*, pp.1226-1229, 2006.

[6] Chan M., Fang X. & Yang C.C. Web searching in Chinese: a study of a search engine in Hong Kong. *Journal of the American Society for Information Science and Technology*, 58(7): 1004-1054, 2007.

[7] DeLuca E.W. & Nurnberger A. Adaptive support for cross-language text retrieval. In *Proceedings of the Intl. Conference in Adaptive Hypermedia and Adaptive Web-Based Systems*, pp. 425-429, 2006.

[8] e-Chaos freeware Greeklish converter. Available for download from : http://www.paraschis.gr/files.php

[9] Gey F.C., Kando N. & Peters C. Cross-language information retrieval: the way ahead. *Information Processing and Management*, 41(3): 415-431, 2005.

[10] Karakos A. Greeklish: An experimental interface for automatic transliteration. *Journal of the American Society for Inf. Science & Technology*, 54(11):1069-1074, 2003.

[11] Koutsogiannis D. & Mitsikopoulou B. Greeklish and Greekness: Trends and Discourses of "Glocalness". *Computer-Mediated Communication*, 9(1), 2003.

[12] Moukhad H. & Large A. Information retrieval from full-text Arabic databases: can search engines designed for English do the job? *Libri*, pp. 63-74, 2001.

[13] Neumann G. & Xu F. Mining answers in German web pages. In *Proceedings of the Conference on Web Intelligence*, 2003.

[14] Ntoulas A., Stamou S., Tsakou I., Tsalidis Ch., Tzagarakis M. & Vagelatos A. Use of a Morphosyntactic lexicon as the basis for the implementation of the Greek wordnet. In *the 2nd Intl. Natural Language Conference*, pp. 49-56, 2000.

[15] Lazarinis F. How do Greek Searchers Form their Web Queries? In *Proceedings of the 3rd Intl. WebIST Conference*, pp. 404-407, 2007.

[16] Qin J., Zhou Y., Chan M. & Chen H. Supporting multilingual information retrieval in web applications: an English-Chinese web portal experiment. In *the 6th Intl. Conference on Asian Digital Libraries*, pp.149-152, 2003.

[17] Rose D. & Levinson D. Understanding user goals in web search. In Proceedings of the Intl. World Wide Web Conference, pp. 13-19, 2004.

[18] Sroka M. Web search engines for Polish information retrieval: questions of search capabilities and retrieval performance. *Information and Library Research*, 32, 2000.

[19] Translatum. Greek-Greeklish converted: Available form: http://www.translatum.gr/converter/greeklish-converter.htm

[20] TSIK Greeklish: ttp://www2.cs.ucy.ac.cy/~tsik/others.html

[21] Varouta M. MyGreeklish to standard Greeklish translator needed. Available at: http://www.proz.com/translation-articles/articles/930/.

N-Grams Conflation Approach for Arabic Text

Farag Ahmed
Information Retrieval Group
Faculty of Computer Science
Otto-von-Guericke-University of Magdeburg
Tel. +49.391.67-11399
fahmed@iws.cs.uni-magdeburg.de

Andreas Nürnberger
Information Retrieval Group
Faculty of Computer Science
Otto-n-Guericke-University of Magdeburg
Tel. +49.391.67-18487
nuernb@iws.cs.uni-magdeburg.de

ABSTRACT

In this paper we present a language independent approach for conflation that does not depend on predefined rules or prior knowledge in the target language. Different from prior studies on Arabic text that use pure n-gram models without any attempt for further enhancement on the basis of refined n-gram similarity measures or stemmer techniques which are language-specific, we propose an unsupervised method based on an enhancement of the pure n-gram model that can group related words based on various string-similarity measures. The proposed approach is based on the enhancement of n-gram comparisons that restrict the search to be in specific locations of the target word by taking into account the order of n-grams. We show that the proposed method is effective to achieve high score similarities between all of the word form variations. Furthermore, it reduces the ambiguity, i.e. obtains a higher precision and recall, compared to the pure n-gram based approaches.

Categories and Subject Descriptors

H.3 **[Information Storage and Retrieval]**:H.3.3 **Information Storage and Retrieval**: Information Retrieval and Search—Conflation techniques;

General Terms

Algorithms, Measurement, Performance, Experimentation, Languages, Verification.

Keywords

Information retrieval, N-gram approaches, Stemming, Arabic language.

1. INTRODUCTION

Conflation is a general term for all processes of merging together nonidentical words which refer to the same principal concept i.e. to merge words which belong to same meaning class. The primary goal of conflation is to allow matching of different variants of the same word. In natural language processing, conflation is the process of merging or lumping together nonidentical words which refer to the same principal concept [1]. In the context of information retrieval (IR) conflation has a more restricted meaning and usually refers to grouping together morphological variants of the same or related words [2]. Conflation algorithms can be broadly divided into two main classes: stemming algorithms, which are language dependent and which are designed to handle morphological variants, and string-similarity algorithms, which are (usually) language independent and which are designed to handle all types of variant [3].

1.1 Arabic language

Arabic is a Semitic language, it consist of 28 letters, and its basic feature is that most of its words are built up from, and can be analyzed down to common roots. The exceptions to this rule are common nouns and particles. Arabic is a highly inflectional language with 85% of words derived from tri-lateral roots. Nouns and verbs are derived from a closed set of around 10,000 roots [4]. Arabic has three genders, feminine masculine and neuter; three numbers, singular, dual (represent 2 things), and plural. May be replace by "The specific characteristics of Arabic morphology make Arabic language particularly difficult for developing natural language processing methods for information retrieval. One of the main problems in retrieving Arabic language text is the variation in word forms, for example the Arabic word "kateb" (*author*) is built up from the root "ktb" (*write*). Prefixes and suffixes can be added to the words that have been built up from roots to add number or gender, for example adding the Arabic suffix ”ان“ (*an*) to the word "kateb“ (*author*) will lead to the word "kateban" (*authors*) which represent dual masculine. What makes Arabic complicated to process is that Arabic nouns and verbs are heavily prefixed. The definite article ”ال“ (*al*) is always attached to nouns, and many conjunctions and prepositions are also attached as prefixes to nouns and verbs, hindering the retrieval of morphological variants of words [5]. In Table 1 an example for the word *student* is presented in order to clarify this issue. Arabic is different from English and other Indo-European languages with respect to a number of important aspects: words are written from right to left; it is mainly a consonantal language in its written forms, i.e. it excludes vowels; its two main parts of speech are the verb and the noun in that word order, and these consist, for the main part, of trilateral roots (three consonants forming the basis of noun forms that are derived from them); it is a morphologically complex language in that it provides flexibility in word formation: as briefly motivated above, complex rules govern the creation of morphological variations, making it possible to form hundreds of words from one root [6]. Furthermore the letters shapes are changeable

SIGIR'07 iNEWS07 workshop, July 27, 2007, Amsterdam, The Netherlands.

Table 1. Word form variations that share the same principal concept whose English translation contain the word student or students

English Translation	Feminine	Masculine
Student	طالبة	طالب
The student	الطالبة	الطالب
(two) students(dual)	طالبتان	طالبان
by the student	بالطالبة	بالطالب
and by the student	و بالطالبة	و بالطالب
By student	بطالبة	بطالب
And By student	وبطالبة	وبطالب
and my student	وطالبتي	وطالبي
my student	طالبتي	طالبي
as, like student	كالطالبة	كالطالب
to the, for the student	للطالبة	للطالب
so , then , and student	فالطالبة	فالطالب
to her/his student	لطالبته	لطالبه
and to the student, and for the student	و للطالبة	و للطالب
his student	طالبته	طالبه
her student	طالبتها	طالبها
and his student	وطالبته	وطالبه
and her student	وطالبتها	وطالبها
their student	طالبتهم	طالبهم
and her students (Dual)	وطالبتيها	وطالبيها
his students	طالباته	طلبته
her students	طالباتها	طلبتها
and his students	وطالباته	وطلبته
and her students	وطالباتها	وطلبتها
his students (for 2 persons)	طالبتيهما	طالبيهما
her students (Dual)	طالبتيها	طالبيها
their students	طالباتهم	طلبتهم
and their students	وطالباتهم	وطلبتهم
our students	طالباتنا	طلبتنا
and Our students	وطالباتنا	وطلبتنا
his students (Dual)	طالبتيه	طالبيه
and his students (Dual)	وطالبتيه	وطالبيه
By students	بطالبات	بطلبة
And By students	وبطالبات	وبطلبة
More than two(plural) students	طالبات	طلبة
and her students (Dual)	وطالبتيها	وطالبيها

in form, depending on the location of the letter at beginning, middle or at the end of the word.

Based on these properties of Arabic language, i.e. that nouns and verbs are massively prefixed and suffixed, we derived the need for modifications of the commonly used n-gram based conflation techniques so that these specific properties are considered. Furthermore, the ambiguity with respect to the similarity score measure of the pure n-gram approach should be reduced.

The remainder of this paper is organized as follows. In Sec. 2 we discuss previous related work on conflation techniques. In Sect. 3 the proposed algorithm is described. The used data, the evaluation and results are discussed in Sect. 4. Some concluding remarks are finally given in Sect. 5.

2. Conflation techniques

In the following we briefly discuss the two major conflation techniques: stemmers and n-gram based techniques.

2.1 Stemmer approaches

In information retrieval systems stemming is used to reduce variant word forms to common roots and thereby improve the ability of the system to match query and document vocabulary [7]. Although stemming has been studied mainly for English, stemming techniques have also been developed for several other languages such as Malay [8], Latin [9], Indonesian [10], Swedish [11] Dutch[12], German [13], French [14], Slovene [15], Turkish [3] and Arabic [16,17]. There are three main approaches for stemming, Dictionary-based, Rule-based, and Statistical-based approaches [18].

Dictionary based approaches provide very good results at the cost of high development efforts for the dictionary. The dictionary contains all known words with their inflection forms. The main weakness for this approach is the missing words in the dictionary which would not be recognized by the system for stemming. Another weakness is the inability of this method to stem inert names and foreign words. Also the need to process a large dictionary during runtime can result in high requirements for storage space and processing time. The closest Arabic equivalent for this kind of stemmer is the *Root-Based stemmer* which is based on extracting the root of a given Arabic surface word by striping off all attached prefix and/or suffix then attempt to extract the root of a given Arabic surface word. Several morphological analyzers were developed based on this concept [19] [16]. The weaknesses for this stemmer are: it does nothing when it comes across some words which have no root, for example the Arabic words ”نحن“ (*we*), بعد (after), تحت (under). Furthermore, the construction of the corresponding dictionaries or rules is a tedious and labor consuming task due to the result of the morphology complexity of Arabic language. Another problem is that only some small linguistic resources are available for Arabic language. The second approach is the *Rule-Based approach*; it is based on set of predefined conditions rules. The most well known stemmer is Porter stemmer [20]. The main weakness for this stemmer is that building the rules for the arbitrary language is time consuming. Furthermore, there is a need for experts with linguistic knowledge in that particular language. The Arabic equivalent for this is the *Light stemmer*. Unlike English, both prefixes and suffixes need to be removed for effective stemming. it is based on striping of prefix and suffix from the word, it use predefined list of prefix and suffix, it is simply striping of prefix and/or suffix without any further processing in the rest of the stemmed word [21, 17, 22]. The weakness of this stemmer is that the striping of prefixes or suffix in Arabic is a not an easy task, removing them can lead to unexpected results, as many words start with one letter or more which can mistakenly assumed to be prefix or suffix. Due to the fact that all light stemmers use the normalization, which consist of several steps, one of them is to Replace آ , أ and إ with bare alef ا to avoid the ambiguity as most of the Arabic users use just the bare alef ا in their search, this is will lead to the result that all “ال “ (*al*) will be mistakenly identified as prefix even if they are in reality not. Example for that the Arabic words “آلات “ (*Machines*), “آلاف” (*Thousands*), “آلام” (*Afflictions*),“ألان” (now) ,“آلم” (pain),“آليات“ (*Mechanisms*). When stripping off all “ال “ (*al*) then the result of the stemmer will be whether other Arabic words, example for that the Arabic word “آلام” when stripping off the “ال “ then the result will be “ام“ which mean mother, or the result will be not an Arabic word.

2.2 N-gram conflation techniques

The main idea of n-gram based approaches, which groups together words that contain identical character sub-strings of length n called n-grams [23], is that the character structure of the word can be used to find semantically similar words and word variants. N-gram as conflation technique differs from stemmers in terms of not requiring language knowledge, predefined rules or a vocabulary database. Furthermore; n-gram approaches take into account the misspelled and the transliterated words.

2.2.1 N-Gram and Arabic text

Over the last years there were several studies which explore the use of n-grams for processing Arabic text. Mayfield et al. [24] have found that n-grams work well in many languages; furthermore they investigated the use of character n-grams for Arabic retrieval in TREC-2001 and found that n-grams of length 4 were most effective. Darwish and Oard examined multiple tokenization strategies for retrieval of scanned Arabic documents, they found out that n-grams of size n=3 or n=4 are well suited to Arabic document retrieval [25]. In [26] Suleiman H. Mustafa assessed the overall performance of two n-gram techniques that he called conventional and hybrid. The conventional approach combines as usual for comparison the first character with the second and second with third and so on till $w_{n-1} + w_n$. The so-called hybrid approach combines the first character with the second and first with third then second with third and second with fourth till $w_{n-2}+w_{n-1}$,$w_{n-2}+w_n$, $w_{n-1} + w_n$. Furthermore, three different levels of word stemming were applied: no stemming, light stemming, and higher-order stemming. In his results Mustafa pointed out that the hybrid approach outperforms the conventional approach. Classifying Arabic text using n-gram frequencies also have been fruitful [27]. However, all of the previous studies rely on the investigation of the use of n-gram on the Arabic text based on those factors: The effectiveness of n-gram size and assessing the performance of existing n-gram approaches. None of the prior studies attempt to modify the pure n-gram model such that it considers also language characteristic while computing the similarity score in order to improve its performance.

3. Computing similarity scores based on n-grams

The n-gram model can be used to compute the similarity between two strings by counting the number of similar n-grams they share. The more similar n-grams between two strings exist the more similar they are. Based on this idea the *similarity coefficient* can be derived. The similarity coefficient δ is defined by the following equation:

$$\delta_n(a,b) = \frac{|\alpha \cap \beta|}{|\alpha \cup \beta|} \qquad (1)$$

where α and β are the n-gram sets for two words *a* and *b* to be compared. $|\alpha \cap \beta|$ denotes the number of similar n-grams in α and β , and $|\alpha \cup \beta|$ denotes the number of unique n-grams in the union of α and β.

3.1 Revised n-gram approach

Arabic nouns and verbs are heavily prefixed and suffixed as described in the first section. As a result of that, it is possible to have words with different lengths that share same principal concept. Figure 1 shows an example of two Arabic words: استمرارية (*Continuousness*) and استمرار (*Continued*) that have different length but belong to same meaning class.

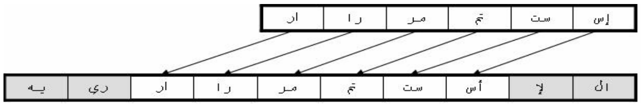


Figure 1 Bigram similarity measure between 2 words with different lengths

Furthermore, the pure n-gram based approach to compute the similarity coefficient as described above Eq (1), does not consider the order of the n-grams in the target word [28]. This increases the probability that the matching score between two strings will be higher even though they do not share the same concept. Therefore, we revised the computation of a similarity between words to take these two aspects into account.

Based on our previous work [29] where we applied a revised n-gram approach (Multispell) for spelling error corrections, we propose here a modified version for the conflation task. For simplicity, we describe our algorithm for n=2 (bigrams). However, the approach can be applied for trigrams and n-grams with n>3 as well. We define bigrams of words by their respective position in the word $w_{i,i+(n-1)}$ where i defines the position of the first letter and $i+(n-1)$ the position of the last letter of the considered n-gram. Thus, the last possible position of an n-gram in a word is defined by $j = |w| - n + 1$, where $|w|$ defines the length of the word. In order to deal with the first and second aspect mentioned above, we define a window of n-grams of the target candidate words that should be compared, i.e. while in Eq. (1) all n-grams are compared with each other, we only compare n-grams that are in close proximity to the position of the n-gram in the word to be compared when computing the similarity score. For example, for a window of size 3, which is the average of the Arabic prefix length, the search will shift to the left or right side. An example is given in Fig. 1, where *w'* defines the given word متسلسلة (*Serialized*) and w a target candidate تسلسل (*Sequence*), in case we don't find the n-gram $w'_{3,4}$ of *w*' in the proper location the algorithm will shift the search to the right side in specific locations, so the n-gram $w'_{3,4}$ will be compared first with the n-grams $w_{3,4}$, then $w_{2,3}$ or $w_{1,2}$ of the target candidate *w*, in case *w* greater than *w'* then the search will shift to left side. This will help also in case of misspelled words. Figure 3 show the similarity measure between the Arabic word التحالفات (the *Alliances*) and الفاتح (*the Conqueror*). Using the pure n-gram model, the similarly coefficient is quite high (85.72 %) although the two words do not belong to the same meaning class. This results from not taking into account the order of the n-gram on the target word. Figure 3 (right) shows the same example using the revised n-gram model. The similarity coefficient is quite low (28.57 %), since the order of n-gram was taken into account.

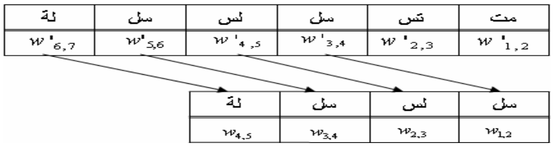


Figure 2. Words with different word lengths that belong to same meaning class

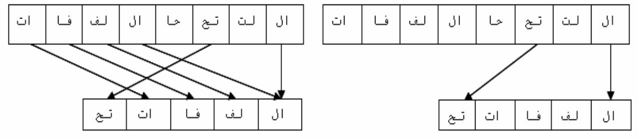


Figure 3. Pure bigram (left) and revised bigram (right)

Overall, the computation of the similarity score *S* for a given n-gram size *n* and a given odd-numbered window size *m* can be defined as follows assuming that *u* is the longer word (if *v* is longer than *u* then *u* and *v* can be simply exchanged):

$$S_{n,m}(u,v) = \frac{\sum_{i=2}^{|u|-n+1} \sum_{j=-\frac{m-1}{2}}^{\frac{m-1}{2}} g(u_{i,i+(n-1)}, v_{i+j,i+j+(n-1)})}{N}, \qquad (2)$$

where $g(a,b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases}$ and

$$u_{i,j} = \begin{cases} substring(u,i,j) & \text{if } i <= j \\ "" & \text{otherwise.} \end{cases}$$

Here, *u* and *v* are the words to be compared, the nested sum counts the number of n-grams in *v* that are similar to n-grams at a window of size *m* around the same position in word *v*. *N* is computed similarly as in Eq. (1).

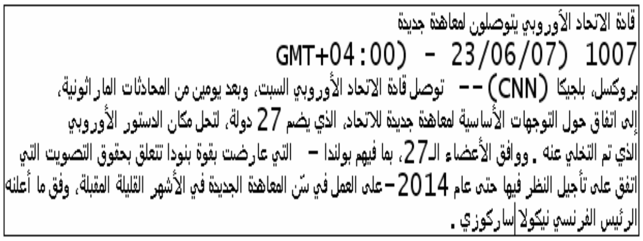
4. Evaluation

In our experiments we compared our approach with the pure n-gram approach for bigrams and trigrams. The reason for not taking a larger value for *n* is the problem of eliminating short words. Previous Arabic studies demonstrate that the character n-gram with n=3 or n=4 are well suited for Arabic document retrieval. Thus, words with length less than 3 or 4 will not be retrieved, since for these no n-grams can be constructed. For example, when trying to retrieve the query يقر (Acknowledges) using trigrams, the relevant result قر (Acknowledged) will be eliminated because no n-grams can be constructed for it as it is less than 3 characters long. The targets words must be at least one character longer than the size of n in order to have the chance to be retrieved. For this reason, we used n=2 in the proposed approach to enable retrieval of short words, as well as other words lengths Furthermore, we used the revised n-gram model to avoid ambiguity as described above in Sect. 3.1.

4.1 Data selection

To collect test data for our evaluations, we crawled the web for articles published on one popular Arabic news Web site ("CNN-Arabic"[1]) in the period from January 2002 until March 2007 (for an example see Fig. 4). We thus obtained 5,792 Arabic documents, all of which are abstracts of articles on news, sport, art, economy and Information Science (size ~60MB). More than 1,400,000 Arabic words were extracted with 101,210 unique words. These articles are supposed to be correctly written and have both a large and rich vocabulary and therefore offer more investigation points in terms of the number of word variations. The articles were carefully checked and cleaned.

The approaches were evaluated against 500 queries that were formulated randomly ensuring that the length of the query terms vary and short as well as long query terms are included. In order to construct the random queries, the algorithm requires the availability of a lexicon of terms that were extracted from the test data.



قادة الاتحاد الأوروبي يتوصلون لمعاهدة جديدة
GMT+04:00) - 23/06/07) 1007
بروكسل، بلجيكا (CNN) -- توصل قادة الاتحاد الأوروبي السبت، وبعد يومين من المحادثات الماراثونية،
إلى اتفاق حول التوجهات الأساسية لمعاهدة جديدة للاتحاد، الذي يضم 27 دولة، لتحل مكان الدستور الأوروبي
الذي تم التخلي عنه . ووافق الأعضاء الـ27، بما فيهم بولندا - التي عارضت بقوة بنودا تتعلق بحقوق التصويت التي
اتفق على تأجيل النظر فيها حتى عام 2014- على العمل في سّن المعاهدة الجديدة في الأشهر القليلة المقبلة، وفق ما أعلنه
الرئيس الفرنسي نيكولا ساركوزي .

Figure 4. Example of an Arabic Document

4.2 Comparison of revised and pure n-gram approaches

In a first experiment we calculated the average precision for each conflation approaches. Table 2 compares the result of the revised bigram and trigram approach with the result of the pure bigram and trigram models. As shown in the Table 2 the result are quite close. The reason for this is that only 6.5 % out of 500 queries words had a length of less than 3 characters, which is the length that affects the ambiguity. The revised bigram and trigram achieved a better improvement over the pure bigram and trigram due to the reduction of the ambiguity.

Table 2. Average precision for all approaches

Techniques	Precision
Revised bigram	92.28 %
Pure bigram	86.22 %
Revised trigram	98.74 %
Pure trigram	96.62 %

In a second experiment we calculated the average precision for the pure trigram and the revised bigram for the similarity thresholds of 60, 65, 70, 75, 80, 85, 90 and 95%. Table 3a and 3b show the comparison of retrieved, relevant, irrelevant and average precision between the revised bigram and pure trigram approaches. The revised bigram achieved clearly improvement over the pure trigram. The reason for that is that the revised bigram takes into account all words lengths which will increase the retrieved index terms size, on the other hand the it take into account the order of the n-gram which will decrease the pure n-gram ambiguity results. This will result in decreasing irrelevant terms retrieved. The trigram achieved better results in terms of the ratio of relevant index terms to the index terms retrieved. The revised bigram achieved better results in terms of how many relevant index terms were retrieved compared to the total number of index terms retrieved (relevant and irrelevant). For example, when selecting a threshold of 60 %, the revised bigram retrieved 5472 index terms relevant and 520 irrelevant, while the pure trigram retrieved 4253 index terms relevant and 189 irrelevant. The pure trigram retrieved less irrelevant index terms at the expense of the total number of relevant index terms retrieved while the revised bigram retrieved less irrelevant index terms compared to the total number of relevant index terms retrieved. It is important to notice, that when interpreting Figure 5c, one need to consider the big difference between the relevant index terms retrieved from each

[1] http://arabic.cnn.com/

method for different thresholds. As it is shown in Table 3a and 3b the performance of the revised n-gram approach is better than that of the pure n-gram in terms of the total number of relevant index terms retrieved. Table 4a and 4b provide a typical example where revised bigram model retrieved 33 relevant index terms while the pure trigram model retrieved 25 relevant index terms. In the second example, Table 4c and 4d show that the revised bigram model retrieved 18 index terms and all were relevant while the pure trigram retrieved only 8 relevant index terms. Figure 5a illustrates that although with a threshold of 85% both approaches have maximum precision, the revised bigram performs better than the pure trigram in terms of the number of relevant index terms retrieved.

Table 3a. Average precision of pure trigram model for different thresholds on 500 words queries

Threshold	Pure trigram			
	Ret.	Relev.	Irrelev.	Precision
60	4442	4253	189	0.957
65	3086	2969	117	0.962
70	2075	2045	30	0.985
75	1872	1843	29	0.984
80	1015	1007	8	0.992
85	549	549	0	1
90	549	549	0	1
95	549	549	0	1
Average Precision				**0.985**

Table 3b. Average precision of revised bigram model for different threshold on 500 words queries

Threshold	Revised bigram			
	Ret.	Relev.	Irrelev.	Precision
60	5992	5472	520	0.913
65	4367	4196	171	0.961
70	2960	2882	78	0.973
75	2464	2393	71	0.971
80	1817	1803	14	0.992
85	694	694	0	1
90	518	518	0	1
95	518	518	0	1
Average Precision				**0.976**

Table 4a. The result of the query "مساعد" (helper) using the revised bigram approach

Revised bigram approach			
S/N	Word	Rel/Irr	Translation
1	مساعد	Rel	Helper
2	بمساعد	Rel	By helper
3	بمساعدة	Rel	By help
4	تساعد	Rel	She helps
5	ساعد	Rel	He helped
6	ساعده	Rel	He helped him
7	ساعدت	Rel	She helped
8	يساعد	Rel	He helps
9	كمساعدة	Rel	As a help
10	ومساعد	Rel	And helper
11	ومساعده	Rel	And his helper
12	ومساعدة	Rel	And help
13	وساعد	Rel	And he helped
14	لمساعد	Rel	For helper
15	لمساعدة	Rel	For help
16	نساعد	Rel	We help
17	مساعدي	Rel	My helper
18	مساعدين	Rel	Helpers
19	مساعديه	Rel	His helpers
20	مساعدو	Rel	Helpers
21	مساعدون	Rel	Helpers
22	مساعدوه	Rel	His helpers
23	مساعده	Rel	His helper
24	مساعدها	Rel	Her helper
25	مساعدا	Rel	A helper
26	مساعداً	Rel	A helper
27	مساعدات	Rel	Helps
28	مساعدة	Rel	Help
29	مساعدتي	Rel	My help
30	مساعدته	Rel	His help
31	أساعد	Rel	I help
32	المساعد	Rel	The helper
33	مساعدون	Rel	Helpers
34	ومساع	Irr	-
35	بمساع	Irr	-
36	لمساع	Irr	-
37	مساعي	Irr	-

Table 4b. The result of the query "مساعد" (helper) using the pure trigram approach

Pure trigram approach			
S/N	Word	Rel/Irr	Translation
1	مساعد	Rel	Helper
2	بمساعد	Rel	By helper
3	بمساعدة	Rel	By help
4	ساعد	Rel	He helped
5	كمساعدة	Rel	As a help
6	ومساعد	Rel	And helper
7	ومساعده	Rel	And his helper
8	ومساعدة	Rel	And help
9	لمساعد	Rel	For helper
10	لمساعدة	Rel	For help
11	مساعدي	Rel	My helper
12	مساعدين	Rel	Helpers
13	مساعديه	Rel	His helpers
14	مساعدو	Rel	Helpers
15	مساعدون	Rel	Helpers
16	مساعدوه	Rel	His helpers
17	مساعده	Rel	His helper

18	مساعدها	Rel	Her helper
19	مساعدا	Rel	A helper
20	مساعداً	Rel	A helper
21	مساعدات	Rel	Helps
22	مساعدة	Rel	Help
23	مساعدتي	Rel	My help
24	مساعدته	Rel	His help
25	المساعد	Rel	The helper
26	مساع	Irr	-

Table 4c. The result of the query “السياسة” (The politics) using the revised bigram approach

Revised bigram approach			
S/N	Word	Rel/Irr	Translation
1	السياسة	Rel	The politics
2	السياسي	Rel	The Political (m)
3	السياسيين	Rel	The Politicians (m)
4	السياسيون	Rel	The Politicians (m)
5	السياسيّ	Rel	The Political (m)
6	السياسيات	Rel	The Politicians (f)
7	السياسية	Rel	The Political (f)
8	السياسات	Rel	The Policies
9	بالسياسية	Rel	By Political
10	بالسياسة	Rel	By politics
11	سياسة	Rel	politics
12	كالسياسة	Rel	As politics
13	وللسياسة	Rel	And for politics
14	والسياسي	Rel	And the Political (m)
15	والسياسية	Rel	And the Political (f)
16	والسياسة	Rel	And the politics
17	للسياسة	Rel	For politics
18	لسياسة	Rel	To politics

Table 4d. The result of the query “السياسة” (The politics) using the revised pure trigram approach

Pure trigram approach			
S/N	Word	Rel/Irr	Translation
1	السياسة	Rel	The politics
2	السياسي	Rel	The Political (m)
3	بالسياسة	Rel	By politics
4	سياسة	Rel	politics
5	كالسياسة	Rel	As politics
6	والسياسة	Rel	And the politics
7	للسياسة	Rel	For politics
8	لسياسة	Rel	To politics

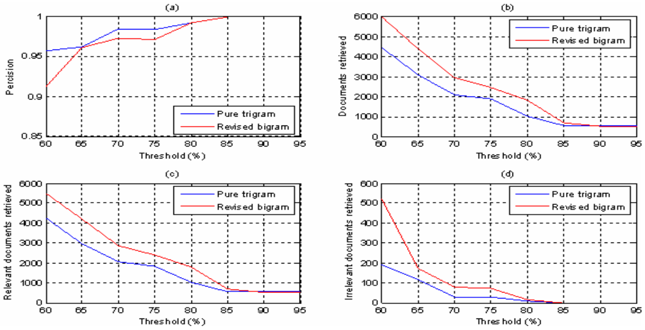


Figure 5. : a) - Average Precision. b) - Total index terms retrieved. c) - Relevant index terms retrieved. d) - Irrelevant index terms retrieved.

In a third experiment we estimated the average recall and F-measure for a sample of 30 queries out of 500. The query terms were selected in the same way as described in Sect. 4.1. For all queries the number of relevant documents were obtained manually, by selecting all possible word variations. As shown in Tables 5a and 5b both approaches have very similar precisions, but the pure trigram approach missed many relevant index terms and therefore has a lower average recall than the revised bigram approach. The revised bigram approach gained up to 75% average recall while the pure trigram approach achieved 49%. Figure 6 illustrates that revised bigram gained a higher average recall than the pure trigram approach, since it took into account different words length and similarity enhancement. As shown in Tables 5a and 5b revised bigram approach gained a higher F-measure up to 76% compared to the pure trigram approach that gained 59%. These results show that the revised n-gram has gained an overall higher degree of retrieval performance than the pure n-gram approach.

Table 5a. Average Recall, Precision and F-measure for the pure trigram approach

S/N	Pure trigram						
	Ret.	Rel.	Irr.	Miss. R.	Precision	Recall	F
1	7	6	1	7	0.85	0.47	0.61
2	6	6	0	11	1	0.36	0.53
3	17	17	0	13	1	0.57	0.73
4	1	1	0	2	1	0.34	0.51
5	29	28	1	0	0.96	1	0.98
6	10	9	1	11	0.90	0.45	0.60
7	22	22	0	3	1	0.88	0.94
8	13	13	0	23	1	0.37	0.54
9	7	7	0	22	1	0.25	0.40
10	6	6	0	14	1	0.30	0.46
11	1	1	0	19	1	0.05	0.10
12	6	5	1	11	0.83	0.32	0.23
13	3	3	0	23	1	0.12	0.46
14	11	11	0	8	1	0.58	0.73
15	14	14	0	24	1	0.37	0.54
16	1	1	0	6	1	0.15	0.26
17	14	13	1	6	0.92	0.69	0.79
18	18	17	1	19	0.94	0.48	0.64
19	16	16	0	14	1	0.54	0.70

20	28	28	0	2	1	0.94	0.97
21	10	10	0	6	1	0.63	0.77
22	10	10	0	30	1	0.25	0.40
23	11	11	0	17	1	0.40	0.57
24	20	20	0	13	1	0.60	0.75
25	12	12	0	8	1	0.49	0.66
26	12	12	0	30	1	0.29	0.45
27	2	2	0	2	1	0.51	0.68
28	38	38	0	19	1	0.57	0.73
29	16	16	0	10	1	0.62	0.77
30	5	5	0	1	1	0.84	0.91
	366	**360**	**6**	**374**	**0.98**	**0.49**	**0.59**

Table 5b. Average Recall, Precision and F-measure for the revised bigram approach

S/N	Pure trigram						
	Ret.	Rel.	Irr.	Miss. R.	Precision	Recall	F
1	9	7	2	6	0.77	0.54	0.63
2	7	7	0	10	1	0.42	0.60
3	28	26	2	2	0.92	0.93	0.92
4	3	3	0	0	1	1	1
5	29	28	1	0	0.96	1	0.98
6	13	12	1	6	0.92	0.67	0.78
7	25	24	1	0	0.96	1	0.98
8	36	35	1	1	0.97	0.98	0.97
9	15	14	1	15	0.93	0.49	0.64
10	10	10	0	10	1	0.50	0.67
11	7	5	2	13	0.71	0.28	0.40
12	18	16	2	0	0.88	1	0.94
13	12	12	0	14	1	0.47	0.64
14	29	19	10	0	0.65	1	0.79
15	38	38	0	0	1	1	1
16	4	4	0	3	1	0.58	0.73
17	20	13	7	6	0.65	0.69	0.67
18	18	17	1	19	0.94	0.48	0.64
19	21	17	3	13	0.80	0.57	0.67
20	29	27	2	1	0.93	0.97	0.95
21	16	16	0	0	1	1	1
22	27	26	1	14	0.96	0.65	0.78
23	17	17	0	11	1	0.61	0.76
24	28	28	0	5	1	0.85	0.92
25	27	23	4	0	1	1	1
26	22	22	0	20	1	0.53	0.70
27	3	3	0	1	1	0.75	0.86
28	49	49	0	8	1	0.86	0.92
29	30	29	1	7	0.96	0.81	0.88
30	6	6	0	0	1	1	1
	596	**553**	**42**	**185**	**0.93**	**0.75**	**0.76**

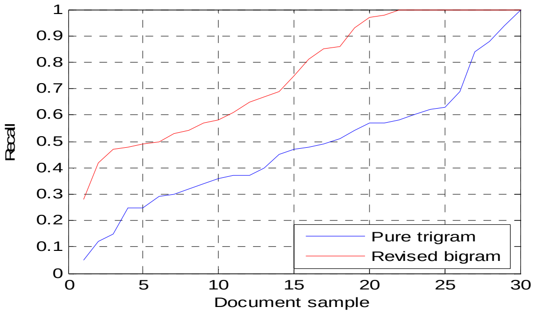


Figure 6 Average Recall for Revised bigram and Pure trigram approaches (sorted by recall value)

5. Conclusions

We presented a language independent conflation approach, i.e. the approach does not depend on any predefined rules or pre-linguistic information knowledge for the target language. We evaluated our approach on Arabic language which is one of most inflectional languages in the world. Since the previous Arabic studies demonstrated that n-grams of size 3 or 4 are the most suitable sizes for Arabic information retrieval, we focused on comparing our approach with trigram based models. The experimental results indicate, that the selection of the n-gram size affects the retrieval performance, i.e. the number of relevant and irrelevant documents retrieved. Using a big size of n lead to the fact that most of the documents retrieved are relevant but at the expense of missing many relevant documents, since the selection of a big n will eliminate short words to be considered. On the other hand, selecting a small value for n lead to the fact that many relevant documents are retrieved but at the same time many irrelevant documents are retrieved due to the ambiguity that is resulting of the small size of the n-grams. Therefore we proposed a revised approach to compare the similarity of words based on n-grams that take the order of n-grams into account. Based on the experimental results we could show that the revised bigram approach provided very good results compared to pure trigrams as well as n-grams with n>3. Furthermore, we demonstrated that the enhancement of the n-gram model provided very good results in term of conflation for heavy inflection languages such as Arabic. Our algorithm was evaluated against 500 randomly selected queries. Unfortunately we had no benchmark results to compare our results with, but based on the quantitative and qualitative experimental results we could show that our algorithm achieved better results than pure n-gram approaches. Furthermore, our algorithm helps to achieve a higher degree of accuracy in the conflation task.

6. REFERENCES

[1] Paice, C.D., 1990: "Another stemmer", *SIGIR Forum, 24(3)*, 56-61 (Fall 1990).

[2] Serhiy Kosinov. Evaluation of n-grams conflation approach in text-based information retrieval. *In 8th String Processing and Information Retrieval Symposium (SPIRE 2001)*, pages 136–142, 2001.

[3] Ekmekcioglu, F. C., Lynch, M. F., and Willett, P. Stemming and n-gram matching for term conflation in Turkish texts. *Information Research News, 7 (1)*, pp. 2-6, 1996.

[4] Al-Fedaghi Sabah S. and Fawaz Al-Anzi (1989) Anew algorithm to generate Arabic root-pattern forms. *Proceedings of the 11th National Computer Conference, King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia.*, pp04-07.

[5] Moukdad, H. (2004). Lost in Cyberspace: How do search engines handle Arabic queries? In Access to Information: Technologies, Skills, and Socio-Political Context. *Proceedings of the 32nd Annual Conference of the Canadian Association for Information Science, Winnipeg*, June 3-5, 2004.

[6] Moukdad, H. and A. Large. (2001). Information retrieval from full-text Arabic databases: Can search engines designed for English do the job? *Libri 51 (2)*, 63-74.

[7] Xu, Jinxi and Croft, W.B., "Corpus-Based Stemming using Co-occurrence of Word Variants" *in ACM TOIS, Jan. 1998, vol. 16, no. 1, pp. 61-81, Computer Science Technical Report TR96-67 (*1996),.

[8] Tai, S. Y., Ong, C. S., and Abdullah, N. A. On designing an automated Malaysian stemmer for the Malay language. (poster). *In Proceedings of the fifth international workshop on information retrieval with Asian languages, Hong Kong*, pp. 207-208, 2000.

[9] Greengrass, M., Robertson, A. M., Robyn, S., and Willett, P. Processing morphological variants in searches of Latin text. *Information research news, 6 (4)*, pp. 2-5, 1996.

[10] Berlian, V., Vega, S. N., and Bressan, S. Indexing the Indonesian web: Language identification and miscellaneous issues. *Presented at Tenth International World Wide Web Conference*, Hong Kong, 2001.

[11] Carlberger, J., Dalianis, H., Hassel, M., and Knutsson, O. Improving precision in information retrieval for Swedish using stemming. *In Proceedings of NODALIDA '01 - 13th Nordic conference on computational linguistics. Uppsala*, Sweden, 2001.

[12] Kraaij, W. and Pohlmann, R. Viewing stemming as recall enhancement. *In Proceedings of ACM SIGIR96.* pp. 40-48, 1996.

[13] Monz, Monz, C., de Rijke, M.: Shallow morphological analysis in monolingual information retrieval for Dutch, German and Italian. In Peters, C., Braschler, M., Gonzalo,J., Kluck, M., eds.: Evaluation of Cross-Language Information Retrieval Systems, *CLEF 2001. Volume 2406 of Lecture Notes in Computer Science.*, Springer (2002) 262–277

[14] Moulinier, I., McCulloh, A., and Lund, E. West group at CLEF 2000: Non-English monolingual retrieval. In Cross-language information retrieval and evaluation: *Proceedings of the CLEF 2000 workshop, C. Peters, Ed.*: Springer Verlag, pp. 176-187, 2001.

[15] Popovic, M. and Willett, P. The effectiveness of stemming for natural-language access to Slovene textual data. *JASIS, 43 (5)*, pp. 384-390, 1992.

[16] Khoja, S. and Garside, R. Stemming Arabic .Computing Department,Lancaster University, Lancaster,1999 *www.comp.lancs.ac.uk/computing/users/khoja/stemmer.ps*

[17] Larkey, L., Ballesteros, L. and Connell, M., "Light Stemming for Arabic IR," Arabic Computational Morphology: Knowledge-based and Empirical Methods, A.Soudi, A. van den Bosch, and Neumann, *G., Editors. Kluwer/Springer's series on Text, Speech, and Language Technology* (2005).

[18] Gelbukh, A., Alexandrov, M. and Han, S.Y.: Detecting Inflection Patterns in NL by Minimization of Morphological Model. *In CIARP 2004, LNCS 3287*, (2004) 432-438

[19] T. Buckwalter. 2002. Buckwalter Arabic Morphological Analyzer Version 1.0 *www.ldc.upenn.edu/Catalog/CatologEntry.jsp?catologId=LDC2002L49*.

[20] M.F. Porter. An algorithm for suffix stripping. Program, 14 (3): 130–137, 1980.

[21] De Roeck, A. N. and Al-Fares, W. A morphologically sensitive clustering algorithm for identifying Arabic roots. *In Proceedings ACL-2000. Hong Kong*, 2000.

[22] K. Darwish. An Arabic Morphological analyzer. http://www.glue.umd.edu/~Kareem/research/

[23] G. Adamson and J. Boreham. The use of an association measure based on character structure to identify semantically related pairs of words and document titles. *Information Storage and Retrieval, (10)*:253–260, 1974.

[24] James Mayfield, Paul McNamee, Cash Costello, Christine Piatko, and Amit Banerjee, JHU/APL at TREC 2001: Experiments in Filtering and in Arabic, Video, and Web Retrieval. In E. Voorhees and D. Harman (eds.), *Proceedings of the Tenth Text Retrieval Conference (TREC 2001)*, Gaithersburg, Maryland, July 2002.

[25] Darwish, K., & Oard, D. W. (2002). Term selection for searching printed Arabic. *In Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR—2002)*, Tampere, Finland (pp. 261–268).

[26] Suleiman H. Mustafa, 2004. Character contiguity in N-gram-based word matching: the case for Arabic text searching. *Information Processing and Management.41 (4)*, 819-827.

[27] Laila Khreisat: Arabic Text Classification Using N-Gram Frequency Statistics A Comparative Study. *The 2006 International Conference on Data Mining Part of the 2006 World Congress in Computer Sciences DMIN 2006*: 78-82

[28] Badam-Osor Khaltar; Atsushi Fujii; Tetsuya Ishikawa: Extracting loanwords from Mongolian corpora and producing a a Japanese-Mongolian bilingual dictionary , *Annual Meeting of the ACL Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL Sydney*, Australia Pages: 657 - 664 Year of Publication: 2006

[29] Farag Ahmed, Ernesto William De Luca und Andreas Nürnberger. MultiSpell: an N-Gram Based Language-Independent Spell Checker In: Poster-Proceedings of Eighth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2007). (to appear).

EusBila, a search service designed for the agglutinative nature of Basque

Igor Leturia
Elhuyar R&D
Zelai Haundi 3, Osinalde Industrialdea
20170 Usurbil (Basque Country)
34 - 943 363040
igor@elhuyar.com

Antton Gurrutxaga
Elhuyar R&D
Zelai Haundi 3, Osinalde Industrialdea
20170 Usurbil (Basque Country)
34 - 943 363040
agurrutxaga@elhuyar.com

Nerea Areta
Elhuyar R&D
Zelai Haundi 3, Osinalde Industrialdea
20170 Usurbil (Basque Country)
34 - 943 363040
nereaa@elhuyar.com

Iñaki Alegria
IXA Taldea, University of the Basque Country
649 postakutxa
20080 Donostia (Basque Country)
34 – 943 015076
i.alegria@ehu.es

Aitzol Ezeiza
IXA Taldea, University of the Basque Country
649 postakutxa
20080 Donostia (Basque Country)
34 – 943 018657
aitzol.ezeiza@ehu.es

ABSTRACT

The performance of major search engines for Basque is far from satisfactory, partly due to the agglutinative nature of the language –it is commonly known that search engines do not perform well with such languages– and partly because it is not a language to which search engines restrict their results.

In this paper we present EusBila, a search service for Basque that relies on the APIs of search engines, yet obtains a lemma-based and language-filtered search by means of morphological query expansion and language-filtering words. It is a cost-effective approach, which we think can be used for other agglutinative or minority languages. We also evaluate how well EusBila performs when carrying out a Basque query, and we compare this performance to that of a major search engine in terms of precision and recall, thus demonstrating that EusBila is a very valid solution.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *query formulation, selection process.*

I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *language generation, language models.*

General Terms

Performance, Design.

Keywords

Search engine, information retrieval, Basque, agglutinative language, minority language.

SIGIR'07 iNEWS07 workshop, July 27, 2007, Amsterdam, The Netherlands

1. MOTIVATION

The problems that non-English languages, and agglutinative languages in particular, have with search engines are well known [5] [6] [7]. While some search engines do seem to use some sort of additional techniques for languages like German [9], other languages, like Hungarian, have no choice but to implement their own engines in order to have a proper web searching tool available [8].

Basque is also an agglutinative language, so these problems are also applicable, but these are not the only difficulties. Being a minority language, Basque has an additional problem: no search engine offers the possibility of returning pages in Basque alone. Therefore, it is impossible to obtain results for numerous words in Basque, because their forms coincide with words existing in other languages.

So the need for a proper Basque search service is clear. A possible solution could be to set up our own search engine, one that would only include pages that are in Basque and which would not index the word forms that a page contains, but its lemmas, as proposed in [14] –Basque language detection and lemmatizing were implemented long ago [1]–, but it is beyond our possibilities and objectives to implement and maintain all the infrastructure that a search engine and its crawling, indexing and serving involves –bandwidth, disk, reliability, etc.–. This is why we embarked on a project to develop a proper Basque search service built upon the APIs of existing search engines, so that the solution obtained and the methodology could be applied to other agglutinative or minority languages as well.

2. METHODOLOGY

2.1 Description of the problem

There are two main reasons that make existing search engines unsuitable for the case of Basque. The first is that Basque is an agglutinative language, that is to say, a given lemma makes many different word forms, depending on the case (genitive, locative, etc.) or the number (singular, plural, indefinite) for nouns and adjectives, and the person (me, he, etc.) and the tense (present, past, etc.) for verbs. A brief morphological description of Basque can be found in [3]. For example, the lemma *lan* ("work") forms the inflections *lana* ("the work"), *lanak* ("works" or "the works"), *lanari* ("to the work"), *lanei* ("to the works"), *lanaren* ("of the work"), *lanen* ("of the works"), etc. This means that looking only for the exact word given or the word plus an "*s*" for the plural is not enough for Basque. And the use of wildcards, which some search engines allow, is not an adequate solution, as these can return occurrences of not only conjugations or inflections of the word, but also derivatives, unrelated words, etc. For example, looking for *lan** would also return all the forms of the words *lanabes* ("tool"), *lanbide* ("job"), *lanbro* ("fog"), and many more.

The second reason is that none of the existing search services can discriminate Basque pages in their searches. Searching in any of them for a technical word that also exists in other languages –*anorexia*, *sulfuroso*, *byte* or *allegro*, to cite just a few examples of the many that exist– or a proper noun or a short word, will not only *not* yield results exclusively in Basque, but often not yield any results in Basque at all.

2.2 Looking for conjugations and inflections

When asking a search engine for a word, we need it to return pages that contain its conjugations or inflections, too. Our approach to this matter is based on morphological query expansion. The importance and use of morphology for various IR tasks has been widely documented ([13] [15] [16] [4]), although it is normally applied by lemmatization at the indexation stage, which is an unattainable objective for us, as has been stated above. Instead, we apply morphological generation at the querying stage. In order to generate all the possible forms of a given lemma, we use a tool created by the IXA Group of the University of the Basque Country. This tool gives us all the possible inflections or conjugations of the lemma, and we ask the search engine to look for all of them by using an OR operator. For example, if the user asks for *etxe* ("house"), we ask the search engine for "(etxe OR etxea OR etxeak OR etxeari OR etxeek OR etxearen OR…)".

This is basically how we solve the first problem. It is a straightforward approach, easy to implement, but one which poses, of course, many minor problems and tweaks. The most relevant ones are as follows:

- The API of each search engine has its limitations with regard to search term count, length of search phrase, etc. We found no documentation on this, so we had to discover each limit by trial and error.
- These limitations render a proper lemmatized search for Basque impossible, as we cannot search for all the conjugations or inflections. So we used a corpus to see which the most frequent cases, numbers, tenses, etc. were, and we send their respective forms, in order to make the search results as satisfactory and representative as possible. In those cases in which the search engine is too limited, we make more than one query, each with some of the conjugations or inflections.
- Unfortunately, there is not much documentation about how search engines behave when they are given more than one search term in an OR. Do they start by looking for the first search term and return its results, and only go on to the next term if there are not enough results with the first one? If so, our results would only be better than those of a general search engine if the word in question was very rare. Anyway, we do not think this is what search engines do, as the *snippets* –short extracts of the pages containing the search term(s)– that they return often contain more than one of the search terms. In fact, we have the impression that they try to return pages that have as many different search terms as possible, which is best for our purposes as it improves representativeness. The increase in recall that emerged in the evaluation seems to confirm our previous assumptions.

All in all, we can conclude that this method enables us to obtain a satisfactory lemmatized search for Basque.

2.3 Language discrimination

We have mentioned earlier that there is no commercial search engine that can distinguish pages in Basque and return them alone. This poses a problem when searching for a proper noun or a word that exists in other languages; this often happens with technical words –*anorexia*, *sulfuroso*, *byte*, *allegro*…– and short words. Although there are language detection tools for Basque, a search for such words returns pages in English, Spanish, etc. but rarely any in Basque, so a subsequent filtering of these pages using a language detection tool would be useless.

The approach we have taken to solve this problem is to include, in the search phrase as a filter, the most frequently used words in Basque, in conjunction with an AND operator. Again, we used a corpus to see which these most used words were.

Unfortunately, the most frequent words in Basque are short and, as such, the chances of their existing in other languages or being used as abbreviations or acronyms is quite high –the four most used words are *eta* ("and"), *da* ("is"), *ez* ("no") and *ere* ("too"), and the first two at least have well-known meanings used in other languages–. Therefore, we had to include more than one filter word, but how many were needed? The higher the number of these words we included, the higher the precision obtained (fewer non-Basque pages were returned). However, there was also loss in recall (more Basque pages were left out because they did not contain one or more of the words), and vice versa. The logical choice was to opt for precision –showing the user results in other languages would give a poor image of a Basque search and, besides, the user would never know how many results he or she was missing–, so in the default behaviour we include four of these most frequent terms in the search phrase. However, if the number of results is too low, the user is given the option of trying again increasing the recall –that is, with less filtering words.

Nevertheless, this failed to resolve the language-filtering problem completely. Even with the filtering words method, non-Basque pages or bilingual pages in which the search term was in a non-Basque part were returned at times. To filter these results, we use LangId, a free language identifier based on word and trigram frequency developed by the IXA group of the University of the Basque Country. This is applied to the snippet returned by the search engine.

By combining these methods we are able to show results that are exclusively in Basque with a high degree of accuracy.

2.4 Variant searching

Expanding the query using variants of the search term to improve the results was suggested long ago [10]. When performing a

Basque search, having the option of looking not only for the word but also for different variants of a word –archaic spellings, common errors– or even typing errors is very interesting. It must be taken into account that the standardization of Basque only started in the late sixties, and that many rules, words and spellings have changed since. Besides, Basque was not taught in schools until the seventies, nor in universities until nearly into the eighties. All this has led to a scenario in which even written production abounds with misspellings, corrections, uncertainties, different versions of a word, etc. But, above all, the main problem is that there are many areas or words upon which no decision as to the standard word or spelling has yet been taken.

The possibility of looking for variants as well has been added as a user option in our tool. All the linguistic tools made for Basque rely upon EDBL, a lexical database developed by the IXA Group of the University of the Basque Country [2]. This database links each word with its known variants, common errors and archaic spellings. So when sending all the possible inflections or conjugations of a word in an OR to the search engine, it is possible to include these variants, too. If, for example, the user inputs the word *jarduera* ("activity"), the system can ask the search engine to seek , simultaneously, the forms of *iharduera*, a now deprecated spelling widely used until 1998.

3. EUSBILA

EusBila is the solution we have developed for a Basque search service, making use of the APIs of major search engines and applying the methods mentioned above –lemma-based searching, language-filtering words and variant searching option–. In this section we will explain in more detail how EusBila works, and what its features are.

3.1 System architecture

The general architecture of the system is as follows:

- The user enters a search term.
- If the user has selected the corresponding option, EusBila uses EDBL to obtain the variants of the search term.
- The morphological generator is called to obtain the inflections and conjugations of the search term.
- A search phrase is built by combining the conjugations and inflections of the search term within an OR operator, and the filtering words with an AND operator.
- The APIs of the search engines are queried with the search phrase.
- The snippets returned by the engines are subjected to a final language test using LangId.
- The results are returned to the user.

3.2 Features

Some of the features of EusBila are as follows:

- Lemma-based and language-filtered search: EusBila performs an internet search for Basque by making use of the APIs of search engines, but simultaneously using morphological generation to obtain a lemma-based search and filtering words to obtain a language-filtered search.
- Variant searching: The user can also choose to look for known variants –common errors, archaic forms…– of the word.
- More than one search term: The user can enter more than one search term, and the lemma-based search is performed for all of them.
- Exact phrase searching: Search engines usually offer the possibility of performing an exact phrase search by enclosing the search terms in double quotes. EusBila offers this possibility too, but it applies the morphological generation to the last word of the phrase, thus performing a proper lemma-based search for whole noun phrases or terms –in Basque only the last component of the noun phrase is inflected.

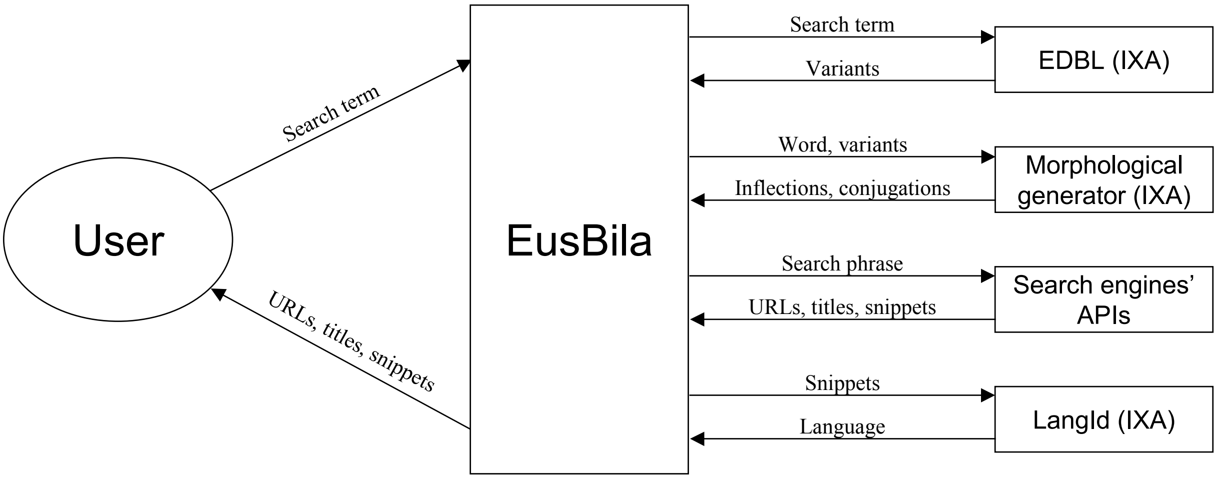


Figure 1. Diagram showing EusBila's architecture.

- Lemma and POS of the search term: The user can enter a search term that is not a plain lemma but a form of a lemma –conjugation or inflection–. The search term is analyzed to get its lemma and POS, and the morphological generation is made according to them. If the form is ambiguous, the most probable lemma and POS are taken for the morphological generation, but when the results are returned, the user is given the option of trying with the other analysis.

- Calls for showing proper snippets: Snippets are the short extracts of the pages that search engines return. As EusBila includes some language-filtering words in the search phrase, the snippets sometimes show these language-filtering words, rather than the word the user was looking for. In these cases EusBila shows no snippet, as the information it contains is irrelevant to the user. But snippets are very useful to help the user decide which link may contain the information he or she is looking for, so EusBila offers the possibility of trying to show as many snippets as possible. This is done by making another call to the APIs of the search engines' for each result without a proper snippet, but restricted to the site and without the filtering words. Naturally, activating this option makes the search slower.

- Various search engines: EusBila can choose among different search engines (Google, Yahoo, Microsoft, Alexa…). But each of these APIs have their own limit in terms of the number of queries per day. So when opening the service to the public, these limits have been taken into account, and we have chosen to offer EusBila's Basque search service through Microsoft's API. The other choices will either be insufficient for the use a Basque search service might have, or else a fee must be paid to use them. We are of the opinion that the number of queries per day offered by Microsoft's API will be enough for EusBila; if not, the commercial license is possible too. In any case, for other minority languages, the other choices might possibly be suitable. The following table shows the limits and licensing possibilities of the APIs we have implemented.

Table 1. Limits and licensing possibilities of the APIs

API	Free access		Commercial license
	Queries / day	Results / Query	
Google	1,000	10	No
Yahoo	5,000	100	No
Microsoft	25,000	50	Yes
Alexa	-	-	Yes

4. EVALUATION

The overall impression of any EusBila user is positive. It is clear that it outperforms the major search engines for a Basque search, as it solves the two problems mentioned above. But in order to translate these impressions into objective figures, we have designed and carried out a quantitative evaluation, comparing the results of EusBila with those of a major search engine.

4.1 Design of the evaluation

To carry out the evaluation, we decided to assess the two improvements of EusBila –morphological generation and language-filtering words– separately, and see the effect they had on precision and recall.

In order to do this, we ran searches for a sample of Basque words both through a commercial search engine and through EusBila (using the API of that same engine), in which only the improvement method being evaluated was activated, and then we compared the first 100 results. We thought it was best to use only one API throughout the whole evaluation, and we chose Microsoft, as it is the one that offers the highest number of queries per day –the intensive use of the API needed for the evaluation would easily surpass the daily limit of the others and would many days just to retrieve the results.

For evaluating the effects of the improvements in recall –either loss or gain–, we measured two variables: the difference in the estimated hit counts returned by the API and the number of different results in the improved query. We are aware that hit counts returned by search engines do not constitute an exact or reliable measure [12], but they are used by many researchers as an acceptable approximation [11]. For our case, we think that hit counts are a clearer indicator of recall than the other measure. Nevertheless, we show the results of the two variables. Both of them were measured and compared automatically, without human intervention.

For evaluating the gain in precision, we measured the difference in the percentage of Basque pages. This was done by language experts, who recorded the language each page returned was in.

With respect to the words, we thought it would be better to carry out the evaluation using real, ordinary Basque search terms, rather than choosing random words. For this purpose, we obtained the search logs spanning a whole year from a very popular science portal in Basque, Zientzia.net (http://www.zientzia.net), which meant that we had more than 500,000 searches that made up a total of more than 50,000 different words. We lemmatized these words and ordered them according to decreasing frequency, and took the topmost ones.

We mentioned above that EusBila's language-filtered search is most noticeable when the search term exists in other languages, or when it is short, or when it is a proper noun. If the word only exists in Basque, the language-filtering words might bring little benefit or even none at all. So when possible, the evaluation variables were measured separately for different categories of words:

- Short words: Words with 5 characters or less. The probability of their existing in other languages is high. The most searched for words in this category (and consequently the ones used for our evaluation) were: *ur* ("water"), *herri* ("people", "town"), *lur* ("earth", "ground"), *zuri* ("white", "to you"), *baso* ("wood"), *euri* ("rain"), *HIES* ("AIDS"), *berri* ("new"), *hartz* ("bear"), *nola* ("how").

- Proper nouns: Proper nouns are usually the same in other languages. The words for this category were *Egipto* ("Egypt"), *Galileo*, *Edison*, *Newton*, *Pluton* ("Pluto"), *Darwin*, *Galilei*, *Thomas*, *Franklin*, *Einstein*.

- International words: Words that we know definitely exist in another language (usually English, Spanish or French). These were the most searched for words in this category: *energia* (“energy”), *historia* (“history”), *mota* (“kind”), *sistema* (“system”), *ozono* (“ozone”), *planeta* (“planet”), *mineral* (“mineral”), *droga* (“drug”), *biografia* (“biography”), *natural* (“natural”).
- Words that are probably found in other languages: Technical words which, despite not being exactly the same in the three languages mentioned above, have quite similar spellings in all of them, so the probability of their existing in some other language is high. These were the words used: *animalia* (“animal”), *petrolio* (“petrol”), *zelula* (“cell”), *nuklear* (“nuclear”), *zentral* (“central”), *klima* (“climate”), *efektu* (“effect”), *zientzia* (“science”), *elektriko* (“electric”), *aparatu* (“system”, “device”).
- Basque words: Words that we are almost sure do not exist in any other language. The most searched for words in this category were *kutsadura* (“pollution”), *berriztagarri* (“renewable”), *elikadura* (“feeding”), *gaixotasun* (“illness”), *ugalketa* (“reproduction”), *berotegi* (“greenhouse”), *gizaki* (“human”), *basamortu* (“desert”), *elikagai* (“food”), *minbizi* (“cancer”).

For the overall measure, we made a weighted average of them, taking into account the frequency of use of each category. To calculate these frequencies, we classified approximately the first 400 words out of the more than 50,000 into one of the categories. This may not seem very much, but they do in fact account for more than 40% of the queries.

Table 2. Frequency and query percentage of each category of word

Category of word	Word		Query	
	Count	%	Count	%
Short words	72	18.65%	44,214	18.64%
Proper nouns	46	11.92%	17,491	7.37%
International words	63	16.32%	46,853	19.76%
Words probably in other languages	100	25.91%	63,266	26.68%
Basque words	105	27.20%	65,345	27.55%
Total categorized	**386**	**0.73%**	**237,169**	**40.27%**
Total	**52,701**		**588,996**	

4.2 Results

4.2.1 Gain in recall due to morphological query expansion

As we decided to evaluate each improvement of EusBila separately, in order to evaluate the effects of morphological generation without using the language-filtering words, it was necessary that it should be done only with the Basque words. We searched for them in Microsoft’s search API, and then we repeated the operation, but using morphological generation. These were the results obtained:

Table 3. Gain in recall due to morphological query expansion for Basque words alone

Word	Hit counts		Increase	New results among the first 100	
	without	with			
	morphological query expansion			Count	%
kutsadura	2,778	3,373	21.42%	37	37.00%
berriztagarri	65	2,729	4,098.46%	88	135.38%
elikadura	10,804	11,818	9.39%	41	41.00%
gaixotasun	4,113	7,617	85.19%	75	75.00%
ugalketa	1,474	1,467	-0.47%	34	34.00%
berotegi	226	247	9.29%	34	34.00%
gizaki	4,897	12,853	162.47%	85	85.00%
basamortu	210	845	302.38%	69	69.00%
elikagai	2,579	8,957	247.31%	84	84.00%
minbizi	147	1,795	1,121.09%	84	84.00%
Total	**27,293**	**51,701**	**89.43%**	**631**	**65.39%**

4.2.2 Gain in precision due to language-filtering words

We then evaluated the effect of language-filtering words without applying morphological query expansion. We first made a normal search and then an additional one with language-filtering words. We measured the increase in the percentage of Basque results for each category of word, and obtained the following results:

Table 4. Gain in precision obtained by language-filtering words for each category of word, and weighted average

Category of word	Weight	% of Basque pages		Increase
		without	with	
		filtering words		
Short words	18.64%	9.82%	97.38%	87.56
Proper nouns	7.37%	0.20%	76.41%	76.21
International words	19.76%	0.00%	97.18%	97.18
Words probably in other languages	26.68%	18.40%	100.00%	81.6
Basque words	27.55%	77.80%	99.57%	21.77
Weighted average		**27.19%**	**97.74%**	**70.55**

4.2.3 Loss in recall due to language-filtering words

In order to measure the loss in recall that language-filtering words could cause, we needed to have some Basque results before applying them, so it was essential that the chosen words should be exclusively Basque words. Thus we searched for such words in Microsoft’s search API, and then carried out the same search, but using language-filtering words. Again, we measured the difference in the hit counts returned by the API and the number of results that did not appear in the first 100 results of the non-language-filtered-search.

We have pointed out above that EusBila gives the option of choosing between precision and recall, and accordingly includes more or fewer language-filtering words. We have made searches with all the different options, from 1 filtering word to 4, so the result of this evaluation is a range of percentages, as shown in the following tables.

Table 5. Loss in recall due to language-filtering words for Basque words alone, measured in hit count decrease

Word	Decrease in hit counts with			
	1	2	3	4
	language-filtering words			
kutsadura	4.72%	19.26%	35.39%	42.84%
berriztagarri	-44.62%	-38.46%	-13.85%	-4.62%
elikadura	4.69%	45.82%	69.40%	73.85%
gaixotasun	1.56%	10.60%	24.48%	35.52%
ugalketa	60.65%	86.30%	83.45%	84.74%
berotegi	3.10%	13.72%	17.26%	21.68%
gizaki	2.37%	8.35%	14.03%	45.62%
basamortu	22.38%	7.62%	26.67%	28.10%
elikagai	0.58%	28.15%	44.44%	54.91%
minbizi	11.56%	13.61%	19.05%	76.19%
Total	**6.48%**	**30.67%**	**46.40%**	**57.69%**

Table 6. Loss in recall due to language-filtering words for Basque words alone, measured in pages no longer among the first 100

Word	% of pages no longer among the first 100 with			
	1	2	3	4
	language-filtering words			
kutsadura	31.43%	34.29%	37.14%	42.86%
berriztagarri	28.07%	35.09%	50.88%	47.37%
elikadura	41.79%	44.78%	67.16%	74.63%
gaixotasun	38.75%	40.00%	50.00%	58.75%
ugalketa	61.54%	58.97%	61.45%	65.38%
berotegi	34.09%	40.91%	46.59%	52.27%
Gizaki	46.91%	43.21%	49.38%	59.26%
basamortu	37.68%	34.78%	43.48%	56.52%
elikagai	30.77%	33.33%	46.15%	55.13%
minbizi	25.61%	24.39%	34.15%	75.61%
Total	**37.87%**	**39.07%**	**48.40%**	**59.07%**

Although the loss in recall is not negligible quantitatively speaking, it is not so important in terms of real user experience. The results that are left out because they do not have one or more of the filter words do not usually have very much content. Any text in Basque that is sufficiently long normally contains the filter words. Therefore, even if some results are left out, the ones that remain are usually longer and, therefore, more relevant. This is an impression we have; it has not been evaluated. And in any case, if there are not enough results or if the user does not find the desired result, the system gives the option of trying again with increased recall –that is, with fewer filter words.

4.2.4 Gain in recall due to morphological query expansion with language-filtering words applied

After measuring the two improvements separately, we thought it would be interesting to evaluate both of them together. The application of language-filtering words would let us measure the effect of morphological generation in words that do not exist exclusively in Basque.

This time we used the most searched for words for each category of word once again. Firstly, we tried a search with the language-filtering words and then with both language-filtering words and morphological generation. Again we measured the difference in the approximate hit counts returned by the API and the number of new results that did not appear in the first 100 results of the non-morphological-query-expansion search.

The results of each category of word and the weighted average can be seen in the following table:

Table 7. Gain in recall obtained by morphological generation for each category of word and weighted average

Category of word	Weight	Gain in hit counts	% of new results
Short words	18.64%	43.75%	71.30%
Proper nouns	7.37%	11.83%	37.85%
International words	19.76%	16.51%	53.47%
Words probably in other languages	26.68%	64.37%	61.05%
Basque words	27.55%	57.36%	59.50%
Weighted average		**40.19%**	**59.94%**

4.3 Summary

This is a summary of the results obtained in the evaluation:

- Gain in precision due to language-filtering-words: increase of 70.55 points –from 27.19% to 97.74%– in the percentage of Basque pages.
- Loss in recall due to language-filtering words: a decrease ranging between 6.48% and 57.69% in hit counts, depending on the number of words
- Gain in recall due to morphological generation:
    - With words that exist only in Basque and without language-filtering words: an 89.43% increase in hit counts
    - With any word and applying language-filtering words: a 40.19% increase in hit counts

The evaluation shows that the benefits obtained with our methodology for a Basque search are considerable, so we can conclude that EusBila is a valid service for searching in Basque. Although the loss in recall due to language-filtering words is significant in quantitative terms, we have the impression that those fewer results are qualitatively better, and in any case, the user can reduce the amount of filter words if necessary.

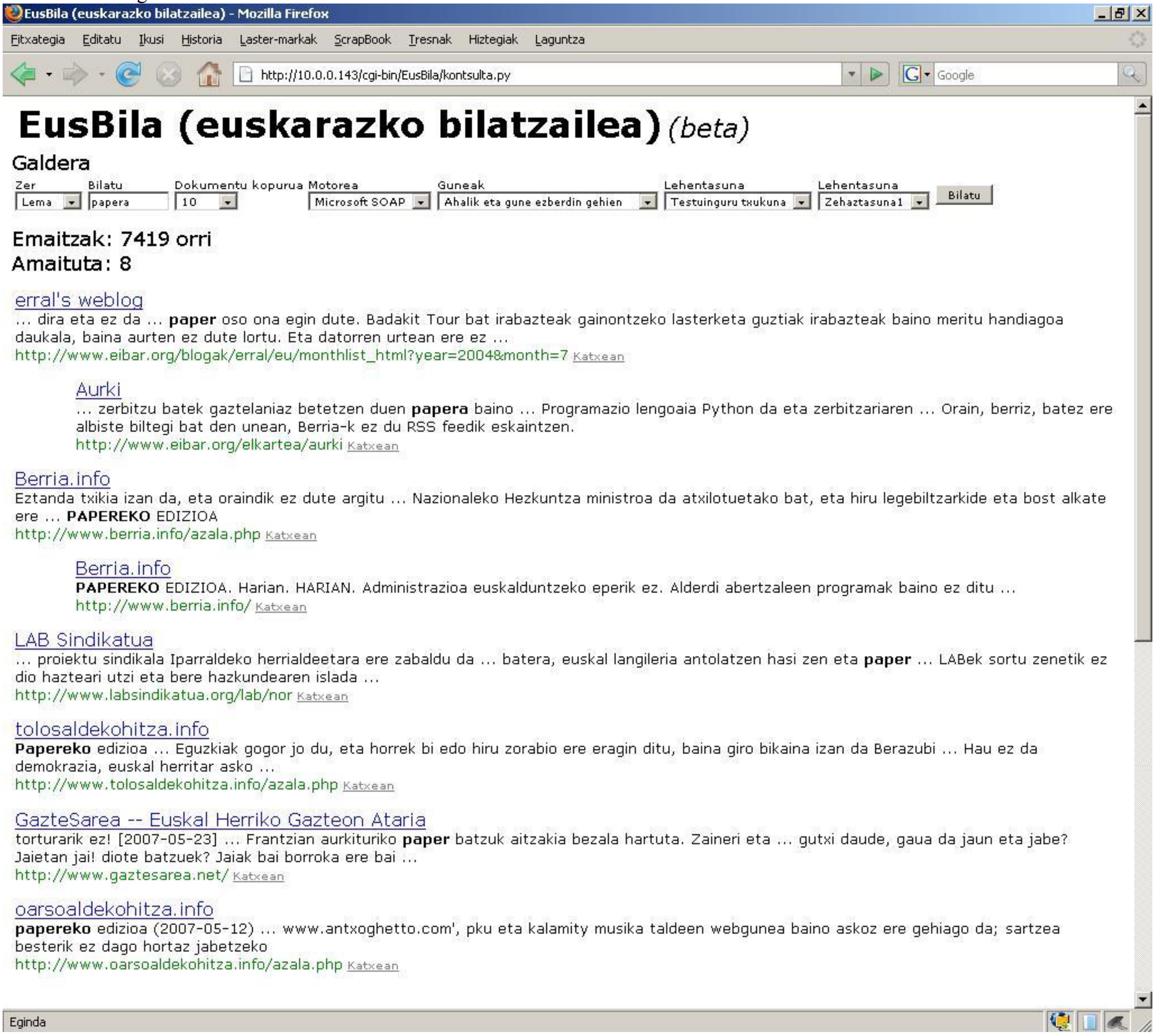
5. CONCLUSIONS

Using search engines for making a query in a minority and agglutinative language like Basque is often a frustrating experience, as they do not perform lemma-based searching or return results in Basque alone.

With EusBila we have built a Basque search service that doesn't need to crawl or index anything, as it makes use of the APIs of the main search engines. To obtain a lemma-based search it uses morphological query expansion, and to obtain pages in Basque alone it uses language-filtering words.

The evaluation has shown that the methodology used is valid, as the increase in performance –gain in precision due to language-filtering words and gain in recall due to morphological generation– is significant. Even if there is a loss in recall due to the language-filtering words, the reduced result set seems to be qualitatively better; moreover, it can be avoided as the inclusion of filter words –and the number of them– is optional.

Furthermore, it seems to us that the methodology used in EusBila could be used by other minority and agglutinative languages to build a search service suited to them, even more so if we take into account that the requirements of the system are very low, as it makes use of the APIs of the search engines.


EusBila (euskarazko bilatzailea) - Mozilla Firefox
Fitxategia Editatu Ikusi Historia Laster-markak ScrapBook Tresnak Hiztegiak Laguntza
http://10.0.0.143/cgi-bin/EusBila/kontsulta.py
Google
Bilatu
Egindа

EusBila (euskarazko bilatzailea) (beta)

Galdera

Zer: Lema | Bilatu: papera | Dokumentu kopurua: 10 | Motorea: Microsoft SOAP | Guneak: Ahalik eta gune ezberdin gehien | Lehentasuna: Testuinguru txukuna | Lehentasuna: Zehaztasuna1

Emaitzak: 7419 orri
Amaituta: 8

erral's weblog
… dira eta ez da … **paper** oso ona egin dute. Badakit Tour bat irabazteak gainontzeko lasterketa guztiak irabazteak baino meritu handiagoa daukala, baina aurten ez dute lortu. Eta datorren urtean ere ez …
http://www.eibar.org/blogak/erral/eu/monthlist_html?year=2004&month=7 Katxean

Aurki
… zerbitzu batek gaztelaniaz betetzen duen **papera** baino … Programazio lengoaia Python da eta zerbitzariaren … Orain, berriz, batez ere albiste biltegi bat den unean, Berria-k ez du RSS feedik eskaintzen.
http://www.eibar.org/elkartea/aurki Katxean

Berria.info
Eztanda txikia izan da, eta oraindik ez dute argitu … Nazionaleko Hezkuntza ministroa da atxilotuetako bat, eta hiru legebiltzarkide eta bost alkate ere … **PAPEREKO** EDIZIOA
http://www.berria.info/azala.php Katxean

Berria.info
PAPEREKO EDIZIOA. Harian. HARIAN. Administrazioa euskalduntzeko eperik ez. Alderdi abertzaleen programak baino ez ditu …
http://www.berria.info/ Katxean

LAB Sindikatua
… proiektu sindikala Iparraldeko herrialdeetara ere zabaldu da … batera, euskal langileria antolatzen hasi zen eta **paper** … LABek sortu zenetik ez dio hazteari utzi eta bere hazkundearen islada …
http://www.labsindikatua.org/lab/nor Katxean

tolosaldekohitza.info
Papereko edizioa … Eguzkiak gogor jo du, eta horrek bi edo hiru zorabio ere eragin ditu, baina giro bikaina izan da Berazubi … Hau ez da demokrazia, euskal herritar asko …
http://www.tolosaldekohitza.info/azala.php Katxean

GazteSarea -- Euskal Herriko Gazteon Ataria
torturarik ez! [2007-05-23] … Frantzian aurkituriko **paper** batzuk aitzakia bezala hartuta. Zaineri eta … gutxi daude, gaua da jaun eta jabe? Jaietan jai! diote batzuek? Jaiak bai borroka ere bai …
http://www.gaztesarea.net/ Katxean

oarsoaldekohitza.info
papereko edizioa (2007-05-12) … www.antxoghetto.com', pku eta kalamity musika taldeen webgunea baino askoz ere gehiago da; sartzea besterik ez dago hortaz jabetzeko
http://www.oarsoaldekohitza.info/azala.php Katxean

Figure 2. Screen capture of EusBila with results for *paper*. As can be seen, the results are lemma-based and in Basque alone

6. REFERENCES

[1] Aduriz, I., Aldezabal, I., Alegria, I., Artola, X., Ezeiza, N., and Urizar, R. *EUSLEM: A lemmatiser / Tagger for Basque*. In *Proceedings of Euralex Conference* (Göteborg, Sweden, 1996), vol. I 17-26.
Also [online] [date: 2007-05-20]: <http://ixa.si.ehu.es/Ixa/Argitalpenak/Artikuluak/1000911639/publikoak/96EUSLEM.ps>

[2] Aduriz, I., Aldezabal, I., Ansa, O., Artola, X., and Diaz de Ilarraza, A. *EDBL: a Multi-Purpose Lexical Support for the Treatment of Basque*. In *Proceedings of the First International Conference on Language Resources and Evaluation* (Granada, Spain, 1998), vol. II 821-826.
Also [online] [date: 2007-05-20]: <http://ixa.si.ehu.es/Ixa/Argitalpenak/Artikuluak/1000911709/publikoak/98LREC.ps>

[3] Alegria, I., Artola, X., and Sarasola, K. *Automatic morphological analysis of Basque*. In *Literary & Linguistic Computing* (Oxford University Press, Oxford, 1996), vol. II nº 4 193-203.
Also [online] [date: 2007-05-20]: <http://hal.ccsd.cnrs.fr/docs/00/08/13/51/PDF/96LITER_M.pdf>

[4] Ambroziak, J., and Woods, W.A. *Natural Language Technology in Precision Content Retrieval*. In *Proceedings of the International Conference of Natural Language Processing and Industrial Applications* (Moncton, Canada, 1998).
Also [online] [date: 2007-05-20]: <http://www.sun.com/research/techrep/1998/smli_tr-98-69.pdf>

[5] Bar-Ilan, J. *Expectations versus reality – Search engine features needed for Web research at mid 2005*. In *Cybermetrics, International Journal of Scientometrics, Informetrics and Bibliometrics* (vol. 9, 2005), nº 1 paper 2.
Also [online] [date: 2007-05-20]: <http://www.cindoc.csic.es/cybermetrics/articles/v9i1p2.html>

[6] Bar-Ilan, J., and Gutman, T. *How do search engines handle non-English queries? – A case study*. In *Proceedings of the 12th international World Wide Web Conference* (Budapest, Hungary, 2003), 415-424.
Also [online] [date: 2007-05-20]: <http://www2003.org/cdrom/papers/alternate/P415/415.pdf>

[7] Bar-Ilan, J., and Gutman, T. *How do search engines respond to some non-English queries?*. In *Journal of Information Science* (vol. 31, 2005), nº 1 13-28.

[8] Benczúr, A. A., Csalogány, K., Fogaras, D., Friedman, E., Sarlós, T., Uher, M., and Windhager, E. *Searching a small national domain - a preliminary report*. In *Poster of the 12th international World Wide Web Conference*, (Budapest, Hungary, 2003), 184-.
Also [online] [date: 2007-05-20]: <http://www2003.org/cdrom/papers/poster/p184/p184-benczur.html>

[9] Guggenheim, E., and Bar-Ilan, J. *Tauglichkeit von Suchmaschinen für deutschesprachige Abfragen*. In *Information, Wissenschaft und Praxis* (vol. 56, 2005), nº 1 35-40.

[10] Jones, K.S., and Tait, J.I. *Automatic search term variant generation*. In *Journal of Documentation* (vol. 40, 1984), nº 1 50-66.

[11] Keller, F., and Lapata, M. *Using the web to obtain frequencies for unseen bigrams*. In *Computational Linguistics* (vol. 29, 2003), nº 3 459-484.
Also [online] [date: 2007-05-20]: <http://acl.ldc.upenn.edu/J/J03/J03-3005.pdf>

[12] Kilgarriff, A. *Googleology is bad science*. In *Computational Linguistics* (vol. 33, 2007), nº 1 147-151.

[13] Krovetz, R. *Viewing morphology as an inference process*. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and Development in Information Retrieval* (Pittsburgh, Pennsylvania, 1993), 191-202.

[14] Langer, S. *Natural languages and the world wide web*. In *Bulletin de linguistique appliquée et générale* (vol. 26, 2001), 89-100.
Also [online] [date: 2007-05-20]: <http://www.cis.uni-muenchen.de/people/langer/veroeffentlichungen/bulag.pdf>

[15] Woods, W.A. *Aggressive morphology for robust lexical coverage*. In *Proceedings of the Sixth Conference on Applied Natural Language Processing* (Seattle, Washington, 2000), 218-223.
Also [online] [date: 2007-05-20]: <http://acl.ldc.upenn.edu/A/A00/A00-1030.pdf>

[16] Woods, W.A., Bookman, L.A., Houston, A., Kuhns, R.J., Martin, P., and Green, S. *Linguistic knowledge can improve information retrieval*. In *Proceedings of the Sixth Conference on Applied Natural Language Processing* (Seattle, Washington, 2000), 262-267.
Also [online] [date: 2007-05-20]: <http://acl.ldc.upenn.edu/A/A00/A00-1036.pdf>

Multilingual Query-Reformulation using an RDF-OWL EuroWordNet Representation [*]

Ernesto William De Luca
University of Magdeburg
Universitaetsplatz 2
39106 Magdeburg, Germany
deluca@iws.cs.uni-magdeburg.de

Martin Eul
University of Magdeburg
Universitaetsplatz 2
39106 Magdeburg, Germany
eul@cs.uni-magdeburg.de

Andreas Nürnberger
University of Magdeburg
Universitaetsplatz 2
39106 Magdeburg, Germany
nuernb@iws.cs.uni-magdeburg.de

ABSTRACT

In this paper, we describe our system architecture that supports users in query formulation and retrieval. Different functionalities for browsing multilingual lexical resources and related Web documents have been implemented. On the one hand, we support the learner/user who first wants to find all possible word senses, retrieve the appropriate translation from the lexical resources and categorize documents (if the user needs such an automatic help) to the most likely word sense and, finally, visualize the search results together with the information provided from the used lexical resources. On the other hand, we help the author who works with RDF/OWL structures for editing and structuring EuroWordNet word senses and translations that are used for query (re)formulation or translation. In this way we help users in formulating multilingual queries, giving also the possibility to explore the intended meanings in other languages.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**];
H.5 [**INFORMATION INTERFACES AND PRESENTATION**]: User Interfaces

General Terms

Multilingual Queries and Retrieval

Keywords

EuroWordNet, RDF/OWL, Multilingual Search Engines

1. INTRODUCTION

In general, an information retrieval system tries to find and retrieve relevant documents related to a user query, with documents and query being in the same language [1], [14], [15], [20]. 6,700 languages are spoken in 228 countries and English is the native language of only 6% of the world population [10]. There are Web pages in almost every popular language. While approximately 70% of the available Web content is in English, the number of native English speakers only constitutes 35.8% of the world's online population [18]. The first accessible Web sites were in English and the first search services (in about 1995) were implemented to meet the needs of this speaking community (e.g. Lycos, AltaVista, Yahoo!). The users of these services were mainly academic people and had enough knowledge of the English language to formulate meaningful queries and to understand the documents retrieved [16]. However, the number of web sites from non-English speaking countries is increasing progressively, and thus the multilingual processing of documents is becoming more and more important.

Nowadays, at least two different user types of a multilingual information retrieval system are identified [15], [16]: The first group of users have good skills in reading a text in a foreign language, but cannot express the information need as well as in the own language. For this case, the system should provide the possibility to find documents in the foreign language using their mother tongue. Such users will benefit enormously if they can enter the queries in their native language, because they can examine relevant documents even if they are not translated. The second users are persons who are monolingual but interested in finding information in documents that are written in foreign languages. Thus, they want to be able to evaluate the relevance of a document to their query before starting a search with a full translation of their information need. These users can use translation aids to be able to understand their search results in a second language.

Now, different features that were seen as too complicated to help the users in the search process, are used. Some examples are given from 'natural language queries', ranked retrieved document results, 'query-by example' or query formulation assistance [2]. Such features are now partially implemented in some search interfaces.

2. MULTILINGUAL LEXICAL RESOURCES

Lexical resources can be used in natural language processing in order to obtain a context description of different word

[*]Copyright is held by the author/owner(s). SIGIR'07 iNEWS07 workshop, July 27, 2007, Amsterdam, The Netherlands.

senses. Searching for a word, we can select concepts based on the linguistic relations of the lexical resource that defines the different word senses. Such disambiguating relations are intuitively used by humans. However, if we want to automate this process, we have to use resources - such as probabilistic language models or ontologies - that define appropriate relations. One of the most important resources available to researchers for this purpose is WordNet [13] and its variations like MultiWordNet [17] and EuroWordNet [22].

2.1 EuroWordNet

In the beginning, WordNet was only developed for the English language. Then, different versions were developed for other languages as for example EuroWordNet [22] for several European languages (Dutch, Italian, Spanish, German, French, Czech and Estonian). Its structure is the same as the Princeton WordNet [13] in terms of SynSets with different semantic relations between them. Each individual wordnet represents a unique language-internal system of lexicalizations. The Inter-Lingual-Index (ILI) was introduced in order to connect the WordNets of the different languages. Thus, it is possible to access the concepts (SynSets) of a word sense in different languages.

Since our goal is to support the user in searching relevant documents in multilingual web collections, we decided to use EuroWordNet [22] for retrieving the meanings of the query and the related translations that can be used for query (re)formulation and translation (see also Sect. 3). But analyzing EuroWordNet, we encountered different problems that had to be solved in order to use it as supporting resource in the search process.

2.2 Fine and Coarse Grained Representation of Word Senses

Since many lexical resources or ontologies, especially WordNet, frequently provide too fine grained word sense distinctions, we implemented the tool LexiRes [7] that offers the possibility to navigate lexical information and helps authors of already available lexical resources to delete or restructure concepts by using semi-automatic merging methods. The restructured information can be navigated and explored. Authors can decide if word senses are unambiguous and important enough to keep them at the same place in the hierarchy or if they express similar concepts and can be merged under the same (now, more general) meaning. One way to obtain a higher granularity is to merge SynSets if they describe a very similar meaning of the same word (see also [9]).

For web search, such methods could be used for creating a reduced structure of the ontology hierarchy, having fewer word senses that are carriers of a more distinctive meaning, in order to categorize the documents retrieved [5]. Therefore, we implemented four online methods to merge SynSets based on the relations of hypernymy and hyponymy, meaning context and domain. An overview and a detailed description of the merging methods is given in [9]. With these methods we can adapt the word sense granularity of a term to the users' needs. Every user has different associations within a concept, so we can adapt the description granularity of a word, adapting it to these associations.

2.3 Combining and Translating Word Senses

The search of a word sense can be expanded using different words. These words not only describe the word context, but also a combination of meanings. In German, for example, there are a lot of words that are compounds. An example of transparent compound words "Schrankwand" (wall unit) in German. Compound words are often not contained in linguistic ontologies such as EuroWordNet. However, the meaning of such a word can, in many cases, be obtained from the combination of the meanings of the word parts. If people, for example, do not know what this compound word means, they start to decompose it in order to extract the individual word senses. In order to understand the sense of the complete compound word, the word parts are then translated in their own language. This process can be applied to many languages.

2.4 RDF/OWL EuroWordNet Representation

Because of the different problems related to WordNet and its variations (see Sect. 2.3, Sect. 2.2 and [9], [8], [4]), we decided to convert it into an RDF/OWL representation, in order to enable the development of more flexible revision methods. In EuroWordNet, one SynSet contains all related word senses, synonyms and relations to other SynSets and to the Inter-Lingual-Index. This information had to be prepared for inclusion in the appropriate RDF Schema and reorganized for a new data representation.

The decision of converting EuroWordNet was also based on the need of extending it (because not all meanings are covered) with other resources. Since most domain-specific ontologies are in OWL and a WordNet monolingual RDF/OWL representation has already been implemented, we decided to extend it for multilinguality purposes. Based on the work done in [21], we converted EuroWordNet into an RDF/OWL representation [11].

Since EuroWordNet has several relations and a structure that is different from the Princeton WordNet, several steps were required to adapt the data to the RDF/OWL Schema of WordNet and to extend this RDF Schema with the new relations. We first analysed the requirements for EuroWordNet and adapted the WordNet RDF Schema to a multilingual representation of EuroWordNet. Then, we converted the EuroWordNet relations into OWL properties and extended the ontology with two domain ontologies [11]. In previous work [11], we discussed this conversion and extension of EuroWordNet in OWL. We described the steps of this conversion and the problems that arose. Afterwards, we showed the inclusion of the OWL "pizza" and "travel" ontologies under the EuroWordNet structure with examples. The first step before including the domain ontologies in the new EuroWordNet OWL hierarchy was to convert these into the OWL format taken from [21]. We applied some merging methods to add these domain ontologies to the EuroWordNet OWL representation implemented. The domain ontology is then added to the generic one, directly under its new hyperonym. The new resulting OWL structure is then shown in LexiRes [8], a visualization tool we developed and adapted, in this case, for handling OWL ontology structures. This work was a first attempt to evaluate how well EuroWordNet could be used as OWL ontology. The use of this OWL implementation and its performance has to be evaluated further in order to see the benefits of it. Another important remark is that at the moment we can only extend EuroWordNet in a “monolingual way”. But finding multilingual parallel resources, we could also easily extend it in a “multilingual way”.

3. SUPPORTING WEB SEARCH WITH MULTILINGUAL LEXICAL RESOURCES

In this section, we discuss approaches for using multilingual lexical resources for combining language exploration and web searches. The main idea is to support users to navigate information using semantic connections between word senses provided by multilingual lexical resources. This can help the user to better understand the different meanings of a word in his/her native language, and even more important, to explore its meanings in a foreign language. Combined web searches can help to understand meanings, since the search results provide examples for word and phrase usage. Furthermore, hit statistics of word co-occurrences in web pages provide hints about correct translations or word usage.

Tools designed to combine the information from both resources in order to support multilingual web search or help to disambiguate word meanings by providing information about the distribution of words in the web to a user, are presented in the following sections.

3.1 Multilingual Web Exploration

Due to the increasing globalization, people are nowadays forced to obtain and to process information not only in their native language, but also in foreign languages. Especially if people want to access and search in multilingual document collections, they need to posses good language skills to discover the correct meaning of the concepts in the target language. Unfortunately, people frequently have a good passive understanding of a foreign language, but are very often not able to find the correct word sense translation. Thus, tools that are able to translate words and implicitly support language acquisition, would be beneficial. In order to support this need, we consider the Web as a learning repository where learners can find examples of word usage. The web documents are a representative example of the combination of words for finding the correct translation and the word-related relevant documents. This combination can be used in tools for language acquisition in computer-assisted language learning (CALL) environments [19] or for cross-language retrieval, while solving some of the still existing problems of multilingual retrieval systems and at the same time implicitly supporting the user in language acquisition. Approaches like CALL applications are used for language teaching and learning in order to support language learners with computer technology. Usually, these tools help the learner to evaluate, reinforce and present the learned topics essentially with interactive elements.

3.2 Multilingual Lexical Resource Exploration

In order to support users dealing with multilingual document collections, we use multilingual lexical resources because they provide information about the linguistic relation of words in-between languages. However, they usually cannot be applied directly, e.g., for tasks like translation or multilingual search, due to the ambiguity of words. On the other hand, huge document collections like the World Wide Web provide statistical information about the distribution and co-occurence of words in almost all languages. This tool was designed to combine the information provided by multilingual lexical resources with the information provided by web searches. Thus, it allows us to study how both resources can be efficiently combined.

A first visualization interface for multilingual search, MultiLexExplorer [4], was developed with a focus on multilingual explorative search. MultiLexExplorer combines word sense disambiguation with a text retrieval approach in an interactive framework. It uses lexical resources to support the user in disambiguating documents (retrieved from the web or a local document collection) given the different meanings (retrieved from lexical resources, in our case EuroWordNet [22]) of a search term having unambiguous descriptions in different languages. By visualizing search results grouped by keyword combinations and word senses, the user can discover languages using lexical resources for disambiguating meanings, combining words and their translation. The translations of all possible source language senses are provided in the target language based on the ILI entries of EuroWordNet (see [22]). Thus, the multilingual exploration is carried out in two directions: finding the correct translations using lexical resources and finding documents according to the search terms and their translations.

3.3 Combining Lexical Resources and Web Searches

Figure 1 shows how our tools are related to the system architecture which implements different functionalities for browsing multilingual lexical resources and related Web documents. Two types of users are recognized. On the one hand, we have the learner/user who first wants to find all possible word senses, retrieve the appropriate translation from the lexical resources and categorize documents (if the user needs such an automatic help) to the proper word sense and, finally, visualize the search results together with the information provided from the used lexical resources. This user can use the MultiLexExplorer [3] for navigating both multilingual lexical resources and documents. On the other hand, we find the author who uses the RDF/OWL LexiRes tool that works with RDF/OWL structures (see Sect. 4), where he/she can load OWL ontologies. In this way the EuroWordNet word senses and translations are restructured and provided for query (re)formulation or translation.

4. RDF/OWL LEXIRES

The main idea of the RDF/OWL LexiRes Tool is to give authors the possibility to navigate the ontology hierarchy in order to re-structure it, by manual merging, adding or deleting word senses. The tool is implemented in Java and uses the Jena Semantic Web Framework [12] for querying and retrieving lexical data. It provides an RDF/OWL model in order to access and query the lexical resource. Using EuroWordNet for cross-language retrieval, we support the author in:

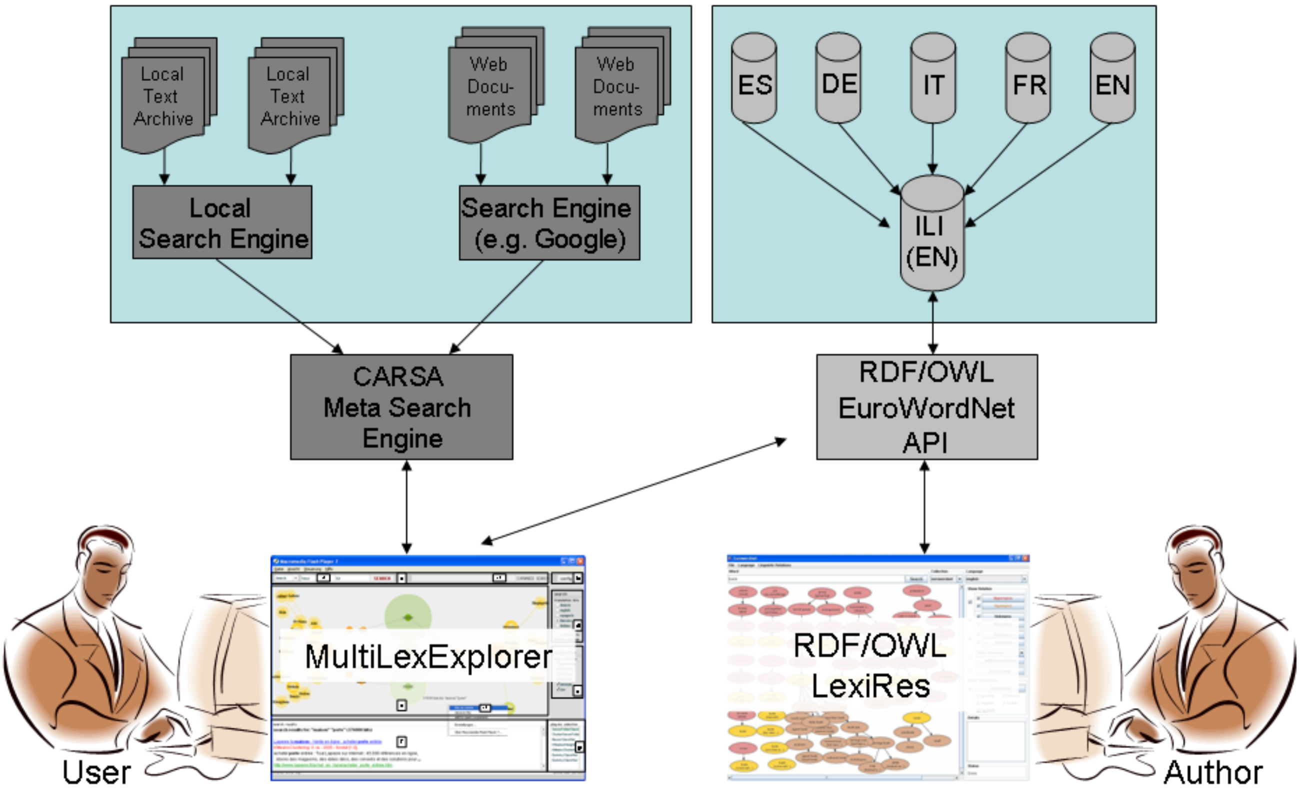


Figure 1: System Architecture.

- Exploring the lexical resource ontology hierarchy
- Disambiguating the word senses of the search word
- Giving the translations of the search word in different languages
- Creating individual lexical collections
- Adding and deleting meanings
- Merging meanings
- Importing OWL ontologies

Figure 2 shows a screenshot of the LexiRes editor. On the top left side, we can choose the source language and enter the query term. On the right side (under the "Show Relations" area), we can choose which collection we want to use and which linguistic relations are to be considered for visualization. Query translations can be enabled in the "Show Translations" area.

Looking for the word "bank", in the English language, the ontology engine retrieves 15 meanings. These meanings describe the different word senses. Every word sense is represented as a SynSet. The author can choose to "Show Properties" or "Hide Properties" with a left mouse click on a SynSet. Here all SynSet-related information is shown. The original RDF resource part of the SynSet can also be displayed by clicking on the right mouse button and choosing the "Show RDF Resource" option. The properties and the RDF code are then shown on the right-hand side under the "Details" box. After logging in, a user-specific lexical resource collection can be created. In our case, the collection contains a reference to the EuroWordNet lexical resource (as default). The author can add or remove meanings in order to enrich or restructure the hierarchy. It is also possible to query the adapted EuroWordNet lexical resource.

To create new meanings, the author has to integrate them into the hierarchy. This is achieved by specifying the most appropriate superordinate node. New words (and their related terms) can be entered in the "Create New Word Sense" dialog. The system searches for known meanings of these terms and suggests (to the author) a list of candidates with their synonyms, descriptions and generic terms. If any meaning matches the meaning of the query term in the hierarchical context, it can be selected and grouped under the superordinate node. Alternatively, the author can generate a new meaning which is then added to the hierarchy (see Figure 3). External domain-specific ontologies can be merged into the collection using the "Import Ontology" option. Then, the ontology can be uploaded and, if suitable, be added in the relation hierarchy. Further details are given in [11].

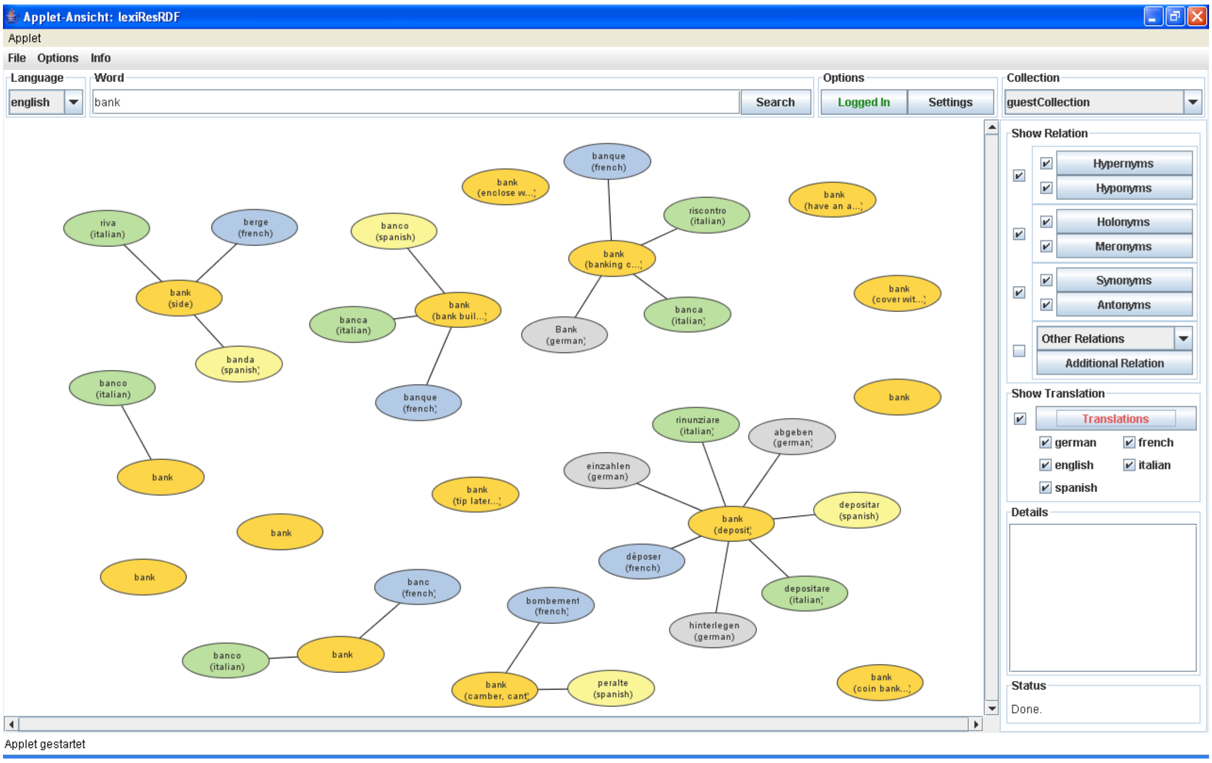
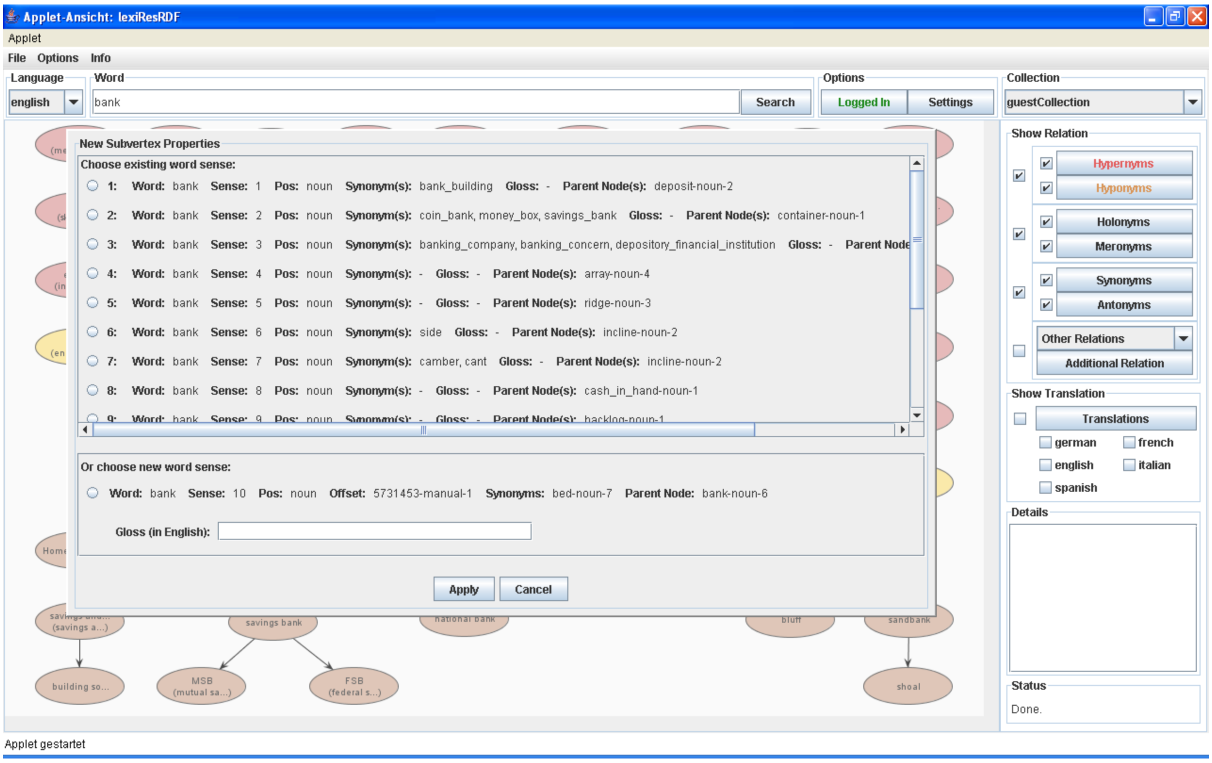


Figure 2: Example of the word “bank” - SynSet translations - in the LexiRes Editor.



New Subvertex Properties

Choose existing word sense:

○ 1: Word: bank Sense: 1 Pos: noun Synonym(s): bank_building Gloss: - Parent Node(s): deposit-noun-2

○ 2: Word: bank Sense: 2 Pos: noun Synonym(s): coin_bank, money_box, savings_bank Gloss: - Parent Node(s): container-noun-1

○ 3: Word: bank Sense: 3 Pos: noun Synonym(s): banking_company, banking_concern, depository_financial_institution Gloss: - Parent Node

○ 4: Word: bank Sense: 4 Pos: noun Synonym(s): - Gloss: - Parent Node(s): array-noun-4

○ 5: Word: bank Sense: 5 Pos: noun Synonym(s): - Gloss: - Parent Node(s): ridge-noun-3

○ 6: Word: bank Sense: 6 Pos: noun Synonym(s): side Gloss: - Parent Node(s): incline-noun-2

○ 7: Word: bank Sense: 7 Pos: noun Synonym(s): camber, cant Gloss: - Parent Node(s): incline-noun-2

○ 8: Word: bank Sense: 8 Pos: noun Synonym(s): - Gloss: - Parent Node(s): cash_in_hand-noun-1

○ 9: Word: bank Sense: 9 Pos: noun Synonym(s): - Gloss: - Parent Node(s): backlog-noun-1

Or choose new word sense:

○ Word: bank Sense: 10 Pos: noun Offset: 5731453-manual-1 Synonyms: bed-noun-7 Parent Node: bank-noun-6

Gloss (in English):

Apply Cancel

Figure 3: Example of the word “bank” - create new word sense - in the LexiRes Editor.

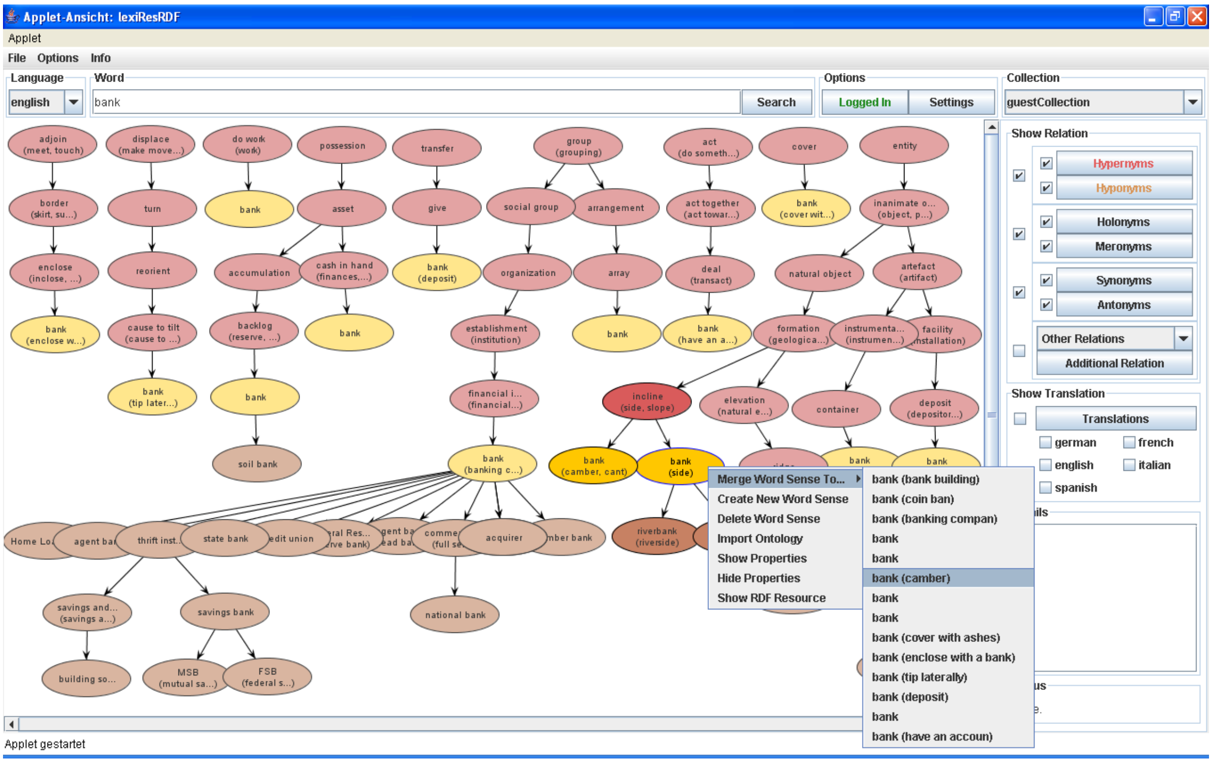


Figure 4: Example of the word "bank" - manual merging functions - in the LexiRes Editor.

When a word sense is removed, the system updates the hierarchy by also removing the respective connections from the linguistic relations. In a graphical representation, this corresponds to deleting all adjacent edges along with the node. If a meaning is deleted, the resulting lack of connection between super- and subordinate words becomes a remarkable situation. Because semantic relations do not have to be transitive, the super- and subordinate nodes cannot always be directly connected. Such situations have to be resolved by the author.

The tool also allows the manual merging of SynSets when the author decides that two SynSets belong to the same meaning and/or describe the same concept. For example, the two "bank" SynSets under the superordinate "incline" SynSet in Figure 4 could be merged. Therefore, the author can pick a "source" SynSet in the hierarchy that should be merged to a "target" SynSet. The "Merge Word Sense To" menu shows all possible target meanings. The "source" meaning with all its relations is transferred to the "target" meaning.

5. CONCLUSIONS

Summarizing, by using EuroWordNet for cross-language text retrieval, we support users in different tasks. They can explore the linguistic context of a word in the general hierarchy. They can search in different languages, e.g., by translating word senses using EuroWordNet. The word senses of different word combinations can be disambiguated. Users can interact with the system changing the search context of the original query and, thus, also the search words and the number of retrieved results, or expanding the original query to restrict the number of retrieved documents. The retrieved web documents can be automatically categorized by using different categorization methods (e.g., as described in [5, 6]).

In our future work, we plan to use information from a learner profile that could be used to automatically modify the granularity of the senses that are distinguished by the system (see the discussion in Section 2.1). An advanced learner might be interested in very fine grained sense distinctions, while a beginner is usually more interested in learning quickly rough language concepts. Currently, it is only possible to manually adapt the granularity of word sense distinction. The use of different ontologies is already possible through our system architecture. A Protégé plug-in could be made available for building knowledge-based tools and applications, or we could make our "RDF/OWL LexiRes" tool publicly available. In the future, we are also planning a user study in order to evaluate the performance of our tools.

6. REFERENCES

[1] A. Abdelali, J. Cowie, D. Farwell, B. Ogden, and S. Helmreich. Cross-language information retrieval using ontology. In *Proceedings of the Traitement Automatique des Langues Conference (TALN 2003)*, Batz-sur-Mer, France, 2003.

[2] W. B. Croft. What do people want from information retrieval? (the top 10 research issues for companies that use and sell ir systems). *D-Lib Magazine*, November, 1995.

[3] E. W. De Luca, S. Hauke, A. Nürnberger, and S. Schlechtweg. Multilexexplorer: Combining

multilingual web search with multilingual lexical resources. In *Proceedings of the combined Workshop on Language-Enabled Educational Technology and Development and Evaluation of Robust Spoken Dialogue Systems. In Conjunction with ECAI'06*, pages 17–21, Riva del Garda, Italy, 2006.

[4] E. W. De Luca, S. Hauke, A. Nürnberger, and S. Schlechtweg. Using multilingual ontologies for adaptive web-based language exploration. In *Proceedings of the International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL06). In Conjunction with the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2006)*, Dublin, Ireland, 2006.

[5] E. W. De Luca and A. Nürnberger. Improving ontology-based sense folder classification of document collections with clustering methods. In P. Joly, M. Detyniecki, and A. Nürnberger, editors, *Proceedings of the 2nd International Workshop on Adaptive Multimedia Retrieval (AMR 2004), part of ECAI 2004*, 2004.

[6] E. W. De Luca and A. Nürnberger. Supporting mobile web search by ontology-based categorization. In B. Fisseni, H.-C. Schmitz, B. Schröder, and P. Wagner, editors, *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen, Proceedings of GLDV 2005*, pages 28–41, 2005.

[7] E. W. De Luca and A. Nürnberger. Lexires: A tool for exploring and restructuring eurowordnet for information retrieval. In *Proceedings of the Workshop on Text-based Information Retrieval (TIR-06). In conjunction with the 17th European Conference on Artificial Intelligence (ECAI'06)*, Riva del Garda, Italy, 2006.

[8] E. W. De Luca and A. Nürnberger. Rebuilding lexical resources for information retrieval using sense folder detection and merging methods. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, Genova, Italy, 2006.

[9] E. W. De Luca and A. Nürnberger. The use of lexical resources for sense folder disambiguation. In *Workshop Lexical Semantic Resources (DGfS-06)*, Bielefeld, Germany, 2006.

[10] H. Haddouti. Survey: Multilingual text retrieval and access. Technical Report review issue, FORWISS (Bavarian Research Center for Knowledge-Based Systems), 1999.

[11] E. W. D. Luca, M. Eul, and A. Nürnberger. Converting eurowordnet in owl and extending it with domain ontologies. In *Proceedings of the Workshop on Lexical-Semantic and Ontological Resources. In Conjunction with the GLDV Conference (GLDV 2007)*, 2007.

[12] B. McBride, D. Boothby, and C. Dollin. An introduction to rdf and the jena rdf api. Technical report, HP, 2006.

[13] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Five papers on wordnet. *International Journal of Lexicology*, 3(4), 1990.

[14] D. W. Oard and B. J. Dorr. A survey of multilingual text retrieval. Technical Report CS-TR-3615, University of Maryland, 1996.

[15] W. C. Ogden and M. W. Davis. Improving cross-language text retrieval with human interactions. In *HICSS*, 2000.

[16] C. Peters and P. Sheridan. Multilingual information access. In *Lectures on Information Retrieval, Third European Summer-School, ESSIR 2000, Varenna, Italy*, 2000.

[17] E. Pianta, L. Bentivogli, and C. Girardi. Multiwordnet: developing an aligned multilingual database. In *First International Conference on Global WordNet*, Mysore, India, 2002.

[18] G. Reach. Global internet statistics, 2004.

[19] J.-B. Son. *Computer-assisted language learning: concepts, contexts and practices.* Lincoln, NE: iUniverse, 2004.

[20] M. Stevenson and P. Clough. Eurowordnet as a resource for cross-language information retrieval. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, Portugal, 2004.

[21] M. van Assem, A. Gangemi, and G. Schreiber. Wordnet in rdfs and owl. Technical report, W3C, 2004.

[22] P. Vossen. Eurowordnet general document, version 3, final. In *www.illc.uva.nl/EuroWordNet/docs/GeneralDocPS.zip*, 1999.

Farsi e-Orthography: An Example of e-Orthography Concept

Behrang Qasemizadeh

Text and Speech Ltd
Tehran, Iran
qasemizadeh@comp.iust.ac.ir

ABSTRACT

Farsi, also known as Persian, is the official language of Iran and Tajikistan and one of the two main languages spoken in Afghanistan. Farsi enjoys a unified Arabic script as its writing system. The fact of using Arabic scripts, a Semitic Language, for representation of Farsi, an Indo-European Language, leads to problems when analyzing, and retrieving Farsi e-text. In this paper we briefly introduce Farsi writing system, and highlight problems when analyzing Farsi electronic texts especially during retrieving Farsi e-texts. Then we introduce the concept of e-orthography. We discuss how e-orthography could be used to improve search results while using keyword based search engines.

Keywords

E-Orthography, Farsi.

1. INTRODUCTION

People in different countries use different characters to represent the words of their native languages. With library automation and the development of networked information structures, the problem of finding a unique way to show information has become much more complex [1][2]. Unicode [4] was devised so that one unique code is used to represent each character, even if that character is used in multiple languages [3]. In this paper, we describe Farsi language transcription in Unicode framework and we discuss challenges that someone would face when processing and retrieving Farsi e-texts.

Farsi is a member of the Indo-Iranian family of the Indo-European languages. Farsi has the properties of agglutinative languages. [5][6] The majority of affixes in Farsi are suffix with limited prefixes as well. After the Arab's conquest in 651 A.D., the Persians adopted an extension of unified Arabic script for writing. Salient characteristics of Arabic script are: existence of various connecting letters, varying graphic forms for many letters depending on their position in a word, varying letter width, absence of full size characters for vowels (vowels are represented with particular signs above and below characters), existence of a number of digraphs and composite letters, writing direction from right to left and absence of upper case and lower case letters. General rules of Arabic writing system are followed by the writing system of Farsi.

Since Arabic is a cursive script, the number of possible shapes that letters actually can adopt exceeds the number of these letters [8]. Letters attach to each other to represent a word. Since Arabic is a Semitic language, it is obvious that how letters must be attached to each other to represent a word. In Farsi, however, due to the fact that it is an agglutinative language, there could be ambiguity in what letters should be written attached together or detached. For instance, the plural form of the word 'كتاب ' /ketâb/ (book) may be written as 'كتابها' /ketâbhâ/ or 'كتاب ها' /ketâb hâ/ (books). This results in some difficulties in Farsi text analysis as cited in [7][8][9], i.e. tokenization of Farsi e-text since word boundaries are not clear. Also, the fact that short vowels usually are not written and capitalization is not used will result in ambiguities that impede computational analysis of the texts. Since these various representations of Farsi are encoded in different manner, then in many cases a search engine can not retrieve Farsi texts.

In the following, after a brief introduction to Farsi encoding, we will introduce the concept of e-orthography and we discuss how it may be used to tackle the problems when analyzing and retrieving Farsi e-texts. The rest of paper is organized as follows: section 2 introduces Farsi transcription and encoding. Section 3, describes the e-orthography concept and its application to Farsi. Finally, we conclude in section 4.

2. Farsi Transcription and Encoding in Digital Environments

"Iranian Academy of Persian Language and Literature", which is a governmental body presiding over the use of the Farsi language, has created an official orthography of the Farsi language, entitled "Dastur-e Xatt-e Fârsi" (Farsi Script Orthography) [10], for the proper representation of texts in the *paper based* system of writing. This orthography is the common orthography widely used by the Persian speakers and indicates how characters must attached to each other to present a Farsi Word. For example, it specifies how affixes should be attached to words.

Unicode standard version 4.0 reserves the range 0600 to 06FF for Arabic characters. The important design principles observed in the Unicode standard and relevant to the representation of Arabic script are characters not glyphs. As mentioned in the previous section, Arabic letters can have up to four different positional forms depending on their position relative to other letters or spaces. According to the design principle "characters, not glyphs", there is no individual code for each visual form (glyph) that an Arabic character can take in varying contexts but there exists only

SIGIR'07 iNEWS07 workshop, July 27, 2007, Amsterdam, The Netherlands.

one code for each actual letter. The correct glyphs to be displayed for a particular sequence of Arabic characters can be determined by an algorithm. In order to display the characters properly, two special characters namely Zero Width Joiner (0x200D) and Zero Width Non Joiner (0x200C) are added to the character codes, either before or after them. The use of these special characters after a code means that a ZWJ or a ZWNJ should be added after the character if the character is not followed by a "right-join causing" character, or a "non-joining character" respectively.

The ISIRI 6219:2002 (Information Technology – Farsi Information Interchange and Display Mechanism, using Unicode) [11] has been proposed as the Farsi standard for using Unicode in digital environment. This standard indicates a subset of Arabic character set in Unicode to be used by Farsi users. Despite this standard, Farsi keyboard layouts are using different codes and therefore, many of Farsi users do not follow this standard. Moreover, the ISIRI 6219:2002 standard does not enlighten how Farsi Orthography can be obeyed in this standard.

The mentioned fact imposes difficulties when retrieving Persian texts, since characters, and therefore words are represented with different codes and search engines do not cover this problem. For example, a word like 'اتمی' /atomi/ which means "Atomic" can be represented in two different coding string since the last character has two encoding option. So, if you search for documents which contain this word, you may miss number of actual results since you have searched just for one of the forms of the word depending on the keyboard layout of your system. The problem is getting more complex when an affix is used to change morphosyntactic features of words. Usually affixes can be written in three different forms regarding the word, attached to the word, detached and with a space between word and affix, detached but with a ZWNJ character between them.

We should consider that the policy of text encoding, tokenization, orthography, and text processing are in interaction with each other. As a real example, consider we would like to define a tag set for Farsi Corpus tagging. As mentioned, in Farsi it is possible that a bound morpheme appears detached from its stem with an intervening space; if we assume space as a delimiter in the tokenization process according to the used orthography, either we have to consider a tag for these bound morphemes during corpus tagging or, we have to consider a more complicated tokenization process as it is cited in [7] [9].

3. Farsi e-Orthography

Unfortunately there exists no standard format for Farsi orthography in the digital environment. As mentioned above, the encoding standard is not sufficient to represent a consistent representation for Farsi. For this reason, we have suggested an approach to represent Farsi electronic texts, or e-orthography. In other words, the e-orthography indicates how the orthography of a language can be followed within an encoding system. Therefore, e-orthography should notice what character codes must be used, how they attach to each other to form a word, and finally which tokenization policy must be taken.

As to Persian, according to the proposed paper-based orthography by the Academy, Farsi affixes must be written attached to their stem. In some cases when the stem ends in a letter which is a "right-join causing character", the affix must attach to the stem with a short space character before it. In order to reach this objective in electronic texts, ZWNJ character has been used as the short space. Also a character set based on the proposed standard in [11] has been used. This way, space characters represent unambiguous word boundaries and the orthography of Farsi e-texts remains consistent with the one proposed in [10]. Also, this transcription results in Farsi e-texts which are more consistent with the e-texts of other languages.

In a keyword based search engine, the e-orthography with the proposed definition influences the effect of search engines in two ways. First of all, the index terms may be changed since the tokenization policy may be varied. Moreover, the user query can be described in other forms which are consistent with proposed e-orthographies. To have an idea, as to Farsi, if we search for a word like 'کتاب‌ها' /ketâbhâ/, a search engine may just retrieve 10% of documents containing this term, considering that first of all character may be represented by different codes, the suffix is written in other forms, characters represented with different lengths, and short vowels may be written or not. An application of proposed e-orthography may be viewed in the development of '1984 corpus' for Farsi [12].

4. Conclusion

This paper introduces the concept of e-orthography and its important role in the efficiency of keyword based search engines. e-orthography tells us how the orthography of a language can be followed in an encoding system, what character codes should be used, how they attach to each other to form a word, and which tokenization policy must be taken in document processing.

E-Orthography can be a guideline for both systems that generate e-text, as well systems which are used to retrieve and manage e-texts. As to the keyword based search engines, the e-orthography can describe how the input query of the users should be refined to retrieve documents. Also e-orthography can change the indices and keywords which are used to retrieve documents.

Although the paper concerns Farsi, the concept of e-orthography can be expanded to other languages as well. Including the e-orthography concept as part of search engines' design can enhance recall and precision parameters. Moreover, the e-orthography concept can be used in other domains like natural language processing and corpus tagging. The mentioned fact indicates that the present standards for text encoding are not sufficient for proper representation, as well as retrieving e-texts.

The concept of e-orthography is getting more important while analyzing languages such as Farsi and Kurdish; languages that have problems in their representation because of the language nature and their writing system.

5. REFERENCES

[1] Erickson J.C. Options For Presentation of Multi-Lingual Text: Use Of the Unicode Standard, *Library Hi Tech, Vol. 15, No. 3-4, 1997.*

[2] Lutz W. Unicode and Arabic Script, *Workshop "Unicode Und Mehrschriftlichkeit In Katalogen", Sbb Pk, Berlin, 2003.*

[3] Wells J.C. Orthographic Diacritics and Multilingual Computing, Language Problems and Language Planning. *Vol. 24, No. 3, 2000.*

[4] The Unicode Standard At Http://www.Unicode.org/.

[5] Samare I. Typological Features Of Farsi. *Journal Of Linguistics, Iran University Press, No. 7, pp 61-80, 1990*.

[6] 7. Keshani, K. Suffix Derivation in Contemporary Farsi. *First Edition, Iran University Press, 199*2.

[7] Karine M., and Zajac R. Processing Farsi Text: Tokenization In The Shiraz Project..*Nmsu, Crl, Memoranda In Computer And Cognitive Scienc, 2000*.

[8] Qasemizadeh, B. and Rahimi, S. Farsi Morphology. *11th Computer Society of Iran Computer Conference, IPM, Tehran, Iran, 2006*.

[9] Rezaie S. Tokenizing an Arabic Script Language. *Arabic Language Processing: Status and Prospects, Acl/Eacl, 2001.*

[10] Iran's Academy Of Farsi Language and Literature. Official Farsi Orthography. *ISBN: 964-7531-13-3, 3rd Edition, 2005*.

[11] Isiri 6219:2002. Information Technology - Farsi Information Interchange and Display Mechanism Using Unicode. *2002*.

[12] QasemiZadeh B., and Rahimi S. Persian in MULTEXT-East Framework. *FinTAL 2006: 541-551, Springer Publisher, Lecture Notes in Computer Science, Vol. 4139, 2006*.

Thesaurus topic assignment using hierarchical text categorization

Francisco J. Ribadas
Dept. de Informatica
Universidade de Vigo
Campus de As Lagoas,
s/n, 32004
Ourense, Spain
ribadas@uvigo.es

Erica Lloves
Telemaco, I. D. S., S.L.
Parque Tecnologico de Galicia
San Cibrao das Vinas
Ourense, Spain
ericalloves@gmail.com

Victor M. Darriba
Dept. de Informatica
Universidade de Vigo
Campus de As Lagoas,
s/n, 32004
Ourense, Spain
darriba@uvigo.es

ABSTRACT

In this paper we present a method for assigning topics from a hierarchical thesaurus to documents written in natural languages. The approach we have followed models thesaurus topic assignment as a multiple label classification problem, where the whole set of possible classes is hierarchically organized. In our case the classification problem is reduced to a sequence of partial classifications, guided by the structure of the topic tree, using a specific set of features at each node in the hierarchy.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*linguistic processing, thesauruses*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*information filtering*

General Terms

Documentation, Design

Keywords

Natural language processing, text categorization, thesaurus

1. INTRODUCTION

In many document processing tasks a correct identification of relevant topics offers a helpful starting point to develop advanced applications to deal with browsing and searching in large collections of documents. In this context, one of the most valuable tools are specialized thesauri. These kinds of structure organize a set of concepts relevant to a given domain in a hierarchical structure, making it possible to employ a sort of controlled vocabulary to simplify document processing.

The classical approach [2] relies on human processing to perform thesaurus term selection after reading each document in the collection. This approach requires the availability of trained experts and suffers from a lack of scalability, since this kind of work is very time consuming and difficult to apply on large collections of documents. We propose to partially replace this kind of human made task with an automatic tool able to identify, for each input document, a list of potential descriptors taken from the domain thesaurus.

In this paper we describe our preliminary work on the automatic assignment of relevant topics, taken from a structured thesaurus, to documents written in natural languages. In our case we are interested in the domain of legislative texts in Spanish. We have an available thesaurus, manually built, with more than 1800 concepts, arranged in a tree structure. We also have a collection of legislative documents, whose main topics have been identified by humans according to the entries in that thesaurus.

The approach we have followed models thesaurus topic assignment as a multiple label classification problem, where the whole set of possible classes is hierarchically organized. Many previous proposals [11] have dealt with text categorization, but the case of hierarchical classes is usually omitted [8] or the generalization to multiple label classification is not directly supported [3, 4].

Our aim is to build a system able to assign descriptive topics to input documents. The set of possible topics is taken from the thesaurus entries and for each document many descriptors may be selected with no special restrictions about the relationships among them. So, in the set of assigned descriptors we could find pairs of sibling entries or any combination of ancestors and descendants.

We also want our system to model in some way the processing made by humans when they perform this kind of task. In our domain, legal texts in Spanish, a very restricted kind of document structures is commonly employed. Document contents can be segmented into consistent text regions and expert users pay special attention to those specific portions which usually carry the most relevant content. Examples of these are the document introduction, the description of the document aims, the destination section or text portions

SIGIR'07 iNEWS07 workshop,
July 27, 2007, Amsterdam, The Netherlands.

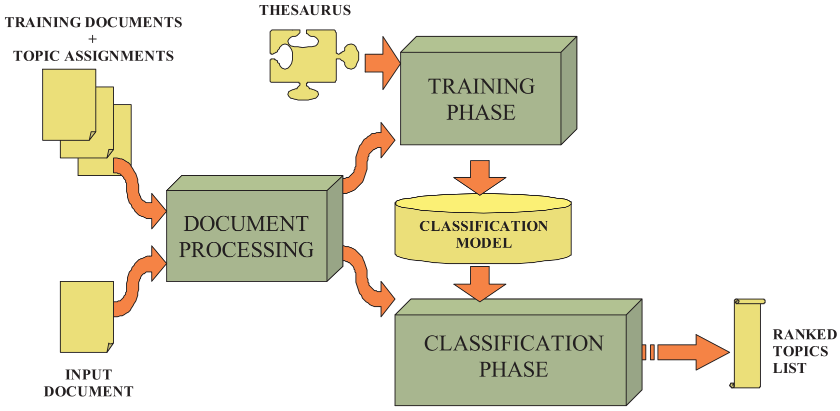


Figure 1: Classification framework.

dealing with legal motivations and background. Also, human experts tend to use the thesaurus structure as a guide to select descriptive topics from it. So, we maintain this two intuitions in our approach that filters entries from the topics hierarchy in a top-down fashion.

The article is outlined as follows. Section 2 introduces our classification framework. Next, Section 3 describes the document representation and processing. In Section 4 the most relevant details about the training and classification strategies are described. Section 5 shows the results obtained in our preliminary experiments. Finally, Section 6 presents our conclusions and future work.

2. TOPIC ASSIGNMENT AS A CLASSIFICATION TASK

Some questions need to be taken into account before starting to describe our proposal. First of all, we are working on a big domain from a text processing point of view. On the one hand, we have a very large collection of legislative documents. These documents tend to be quite long, with sizes ranging from hundreds to thousands of words. On the other hand, we also have a big set of potential classes arranged in a tree. In this context there are two main aspects to have in mind. First of all, we must ensure a practical computational cost, both in the training phase and specially in the topic assignment phase. We also must offer a robust classification framework, able to return a consistent list of topics for a great variety of input documents, without contradictions. With regard to the available resources, we have a thesaurus built by hand for the domain of legislative text and a set of historic documents with their corresponding descriptors assigned by human experts, which will be employed in the training phase.

With these premises in mind, a first approach could be to take the available documents and train a big classifier using all of the topics in the thesaurus as output classes. This approach is almost impractical from a computational cost point of view, but also it has many important problems with output quality and a lack of robustness and consistency. Training such a classifier involves estimating a large number of parameters with too many irrelevant features that will disturb the classification decision.

The strategy we have chosen is inspired by Koller and Sahami's work [6] and takes advantage of the class hierarchy to simplify the classification task in two aspects. Firstly, the global classification problem is reduced to a sequence of partial classifications, guided by the structure of our topic tree. Secondly, the computational cost for each classification step is reduced and the resulting quality is improved by means of the use of a specific set of features, exclusive to each node in the hierarchy. In this manner, the classification decision is distributed over a set of partial classifiers across the topics tree. In this model each internal node will be responsible for a local classification decision, where only a small set of features from the document will be taken into account to select the most promising descendants, where this processing will be repeated.

The main difference from Koller and Sahami's proposal is the final output of our classifier. Our aim is to get a set of relevant topics taken from the thesaurus, ordered according to their relevance, instead of a single class. We replace the greedy approach used in the original work, where only the best successor for each node was considered at each step. In our proposal, we proceed level by level, and all of the paths starting at successors with higher evaluation values are taken into account, and they are followed until no further expansion is admissible. The final set of topics is composed of those class nodes, ranking them according to the strength of the classification steps that lead to them.

In Fig. 1 we show the main phases in our approach, where three main components are outlined. Document processing, which is applied both in training and classification, has the responsibility of cleaning the documents to reduce the number of features employed to represent them. In the training phase, the training set of documents are taken and the topic hierarchy is traversed top-down, performing at each node local feature selection and training the corresponding partial classifier. Finally, in the classification phase, for each input document the topic tree is traversed top-down using the trained classifiers to decide whether the corresponding topic is suitable to be taken as a final descriptor and to make routing decisions to select one or more branches to continue searching. At the end, the list of potential descriptors for that document will be ranked and returned to the user.

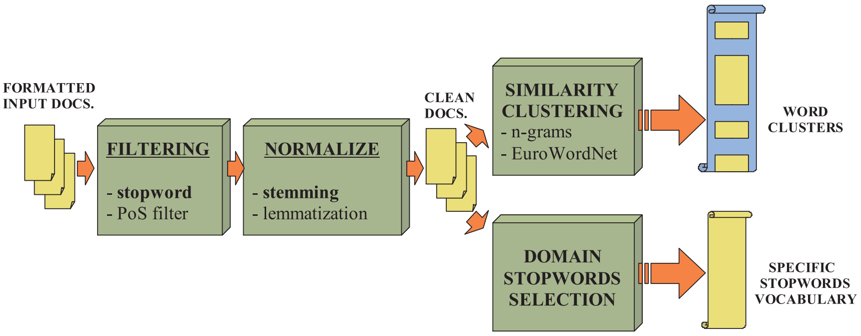


Figure 2: Collection preprocessing.

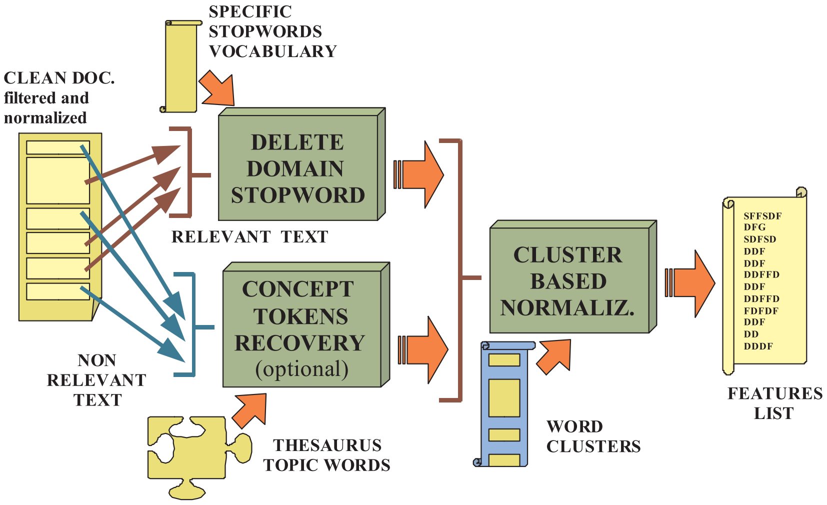


Figure 3: Document processing and representation.

3. DOCUMENT PROCESSING

Since this is a first approach to this kind of problem, we have tried to avoid using complex linguistic resources, like taggers, lemmatizers or shallow parsing [12]. Original documents were in HTML and PDF format and the first step was to extract plain text from them. Those text files were previously preprocessed to segment their text into regions and to identify which of those regions are relevant and could be suitable to extract potential descriptors from them.

A first processing, shown in Fig. 2, is performed on the whole collection. To omit non-relevant words we use a generic stopword list for Spanish. Remaining words are normalized by means of stemming rules to overcome lexical variation problems. Once all of the the documents in the collection have been cleaned, two structures are built. A specific stopword list, containing a vocabulary of commonly used words in the considered domain, makes it possible to get rid of words frequently employed in legislative texts. A dictionary of similar words, that allows us to identify groups of related words, is also built. We have employed a method to detect similar tokens at orthographic level by means of a hierarchical clustering algorithm which uses a n-gram based distance [7] between word characters.

Both in training and classification, the list of features to be employed is extracted from cleaned documents in the way shown in Fig. 3. From the relevant regions of the input document, domain specific stopwords are deleted. Optionally, some words that appear as labels in the thesaurus topics can be recovered to be taken into account as features. These features have been demonstrated to be useful when short documents are processed. The surviving features will undergo a sort of semantical normalization using the similar word clusters built from the whole training collection. After that we obtain the list of features that will describe the input document in the training and classification phases.

4. TRAINING AND CLASSIFICATION

In this section we show the main components that comprise our approach. Once the set of training documents have been processed they are employed to train our hierarchical categorization model. This model contains for each thesaurus node the set of features with higher discrimination power at that level and a trained partial classifier to make the routing decisions. The idea behind this strategy is that using this set of classifiers' decisions will be more precise and the overall classification quality will be improved.

4.1 Training the hierarchy

In the training phase we take the whole training collection with the set of topics associated to each document and we

Thesaurus nodes and documents

ganadería /*livestock* **(1)** **[doc1, doc2, doc10]**
- certamen ganadero/*cattle contest* **[doc2, doc3 , doc4]**
- mejora del ganado/*cattle improvement* **[doc4]**
  - mejora de la raza/*race improvement* **[doc5, doc6]**
  - protección razas autóctonas/*native races protect.* **[doc6**
- sanidad animal/*animal heath* **(2)** **[doc 7, doc9]**
  - higiene de los animales/*animal hygiene* **[doc10, doc11]**
  - mejora genética/*genetic improvement* **[doc12]**
  - sacrificio de animales/*animal sacrifice* **[doc1]**
- sistemas de ganadería/*cattle ranch systems* **[doc5, doc8]**

Training at node "ganadería" (1)

5 classes + 16 training instances

current_topic:	doc1, doc2, doc10
certamen_ganadero:	doc2, doc3, doc4
mejora_del_ganado:	doc4, doc5, doc6
sanidad_animal:	doc7, doc9, doc10, doc11, doc1, doc12
sistemas_de_ganaderia:	doc5, doc8

Training at node "sanidad animal" (2)

4 classes + 6 training instances

current_topic:	doc7, doc9
higiene_de_los_animales:	doc10, doc11
mejora_genetica:	doc12
sacrificio_de_animales:	doc1

Figure 4: An example of training at two nodes.

traverse the topic tree, performing two tasks at every thesaurus entry. For each node, the subset of documents with at least one descriptor being a descendant of the current concept is selected. With these documents we apply very simple feature selection techniques to find a set of features with the highest discrimination power among the different branches starting at this node.

The actual feature selection method employed in our system is controlled by two thresholds, **Th1** and **Th2**, and works as follows:

1. The set of classes for the current node, $\mathcal{C}$, is built.
   - One class will correspond to the topic at the current node, which will be associated with documents having that topic as a descriptor.
   - For every direct descendant of the current topic another class is defined, which will be associated with documents having at least one of the descriptors belonging to the branch starting at that descendant, as shown in Fig. 4.
2. For each class $C_i \in \mathcal{C}$:
   - Every word w_{ij} in a document j associated with C_i is inspected.
   - Word w_{ij} will survive feature selection if:
     (a) w_{ij} is present in AT LEAST **Th1** % of documents being associated with class C_i
     (b) w_{ij} is present in NO MORE than **Th2** % of documents not being associated with class C_i

Once the external feature selection is performed, a specialized classifier is trained to select the most promising branches at the current level. For each document a feature vector is built. Only the relevant stems selected for the current concept are employed, using their *tf-idf* [10] as feature values. The class for this training vector will be the current topic, if it is actually associated with the document, or the label of one of its sons, if some topic associated with the document falls into that branch. Fig. 4 illustrates this idea. We have employed the WEKA machine learning engine [13] to train the specialized classifiers for each topic in our hierarchical thesaurus. We have tested several classification algorithms to be employed inside this hierarchical categorization scheme, as it can be seen in the experimental results section.

4.2 Hierarchical classification

Once all of the partial classifiers have been trained, the assignment of topics to new documents means traversing the thesaurus tree, as shown in Fig. 5. Starting at the thesaurus root, the feature vector for the document is built using the selected features for each node, and the most promising branches according to the partial classifier results are followed.

The original proposal by Koller and Sahami defines a single class output. They perform a greedy search selecting at each node only one class and stopping when a leaf is reached. Since we are interested in multilabel classification, we have added two new components in our classification strategy. In this way, node classifiers have two missions. The first one is to detect if a topic is suitable to be considered as a final descriptor, and the second one is to make a routing decision to determine the next steps in the search.

The routing decisions taken at each node are controlled by simple thresholds that take into account both the number of alternatives at each node and the strength of the potential classes returned by the classifier. If the class for the current topic has an evaluation value higher than this threshold it is considered to be suitable as a topic for describing this document. When a leaf is reached or no successor classes have sufficient strength, the deeping is stopped. The final list of potential topics is ranked according to the set of values obtained in the sequence of partial classifications that lead to them. Different formulae, average, maximum or product, can be used to combine the strength values obtained in the path of classifications from the root to that descriptor.

5. EXPERIMENTAL RESULTS

To illustrate our proposal we will review some preliminary experiments we have performed to test our method. In these

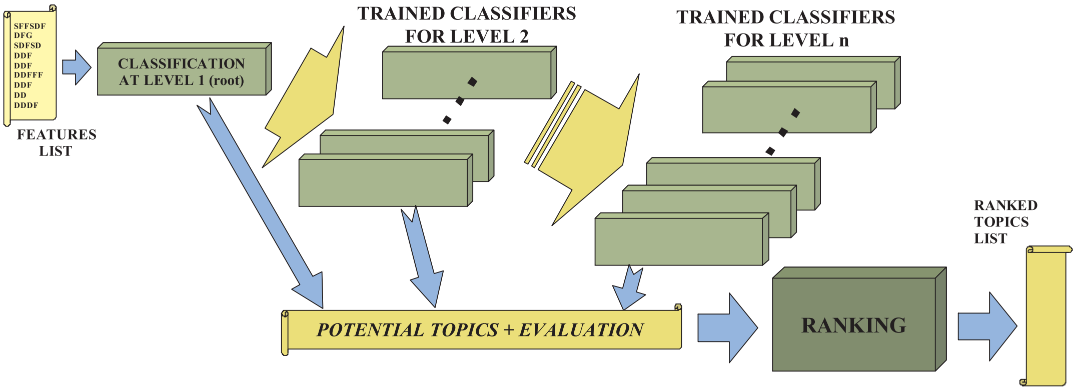

SFFSDF
DFG
SDFSD
DDF
DDF
DDFFF
DDF
DD
DDDF
FEATURES LIST
CLASSIFICATION AT LEVEL 1 (root)
TRAINED CLASSIFIERS FOR LEVEL 2
TRAINED CLASSIFIERS FOR LEVEL n
POTENTIAL TOPICS + EVALUATION
RANKING
RANKED TOPICS LIST

Figure 5: Hierarchical classification.

experiments we have employed a portion of the legal corpus donated by Telemaco, S.L., with 2333 legislative documents with their corresponding set of descriptors assigned by human experts. These descriptors where taken from a set of 1873 thesaurus topics about the fields of agriculture, livestock and fishing. This corpus was randomly split to build a training data set with 2124 documents and a test dataset with 209 documents. To evaluate the experimental results we have employed two well known measures in the Information Retrieval field, precision and recall, using a modified version of the standard `trec_eval` [1] tool to compute them.

In the experiments reported in this paper we have evaluated two aspects in our proposal. Firstly we have tested the influence on the final results of different approaches to generating the input text. Secondly we have evaluated the suitability of several text classification algorithms. Fig. 6 shows the results obtained using different text sources from the original documents to extract features from them. We have taken words only from the document title (experiment [T]), words from the title and the relevant regions (experiment [T+RR]) and we included selected words taken from non-relevant regions, giving them different weights (experiments [T+RR+SW] and [T+RR+2SW], where selected words count twice). As can be seen in Fig. 6 the best results were obtained using words from the title, words from relevant regions and selected words from non-relevant ones.

In Fig. 7 we show the average precision and recall values obtained in a set of experiments to test the use of different machine learning algorithms to perform the partial classifications across the thesaurus tree. We have tested a Naive-Bayes implementation [5], a k-Nearest Neighbors(k-NN) [1] learning method, with different values for k, and a Support Vector Machine model using Sequential Minimal Optimization(SMO) [9], all of them are included in the WEKA machine learning engine [13]. As can be seen, the best results, both in precision and recall, where obtained with the k-NN method, with a better balance between absolute recall and precision when seven neighbors were employed. In a deeper review of the descriptors obtained in that run, our approach gave better results when dealing with the most general topics, but it was unable to get a human level performace with the most specific descriptors.

6. CONCLUSIONS AND FUTURE WORK

In this article we have proposed the use of a hierarchical multilabel classification approach which allows us to face the thesaurus topic assignment problem. We have followed a very flexible method, easy to be adapted to deal with different practical domains and allowing the use of several classification and text processing algorithms. The developed system offers quite good performance on average documents, even being able to avoid some human inconsistencies. When complex or very specific documents are processed, our tools are unable to work at human expert level, opening a field for further improvements.

With respect to future work, several aspects should be studied in our classification approach. Firstly, we intend to extend our experiments to other domains and languages, in order to test its generality. Secondly, we aim to improve the system by integrating more powerful natural language processing tools.

Acknowledgements

The research reported in this paper has been partially supported by Telémaco, Información, Documentación y Sistemas, S.L. (`http://www.telemaco.com`), Xunta de Galicia (PGIDIT05SIN044E, PGIDIT05PXIC30501PN), Ministry of Education and Science (TIN2004-07246-C03-01) and University of Vigo

7. REFERENCES

[1] Aha, D., and D. Kibler. Instance-based learning algorithms. *Machine Learning*, vol.6, pp. 37-66, 1991.

[2] J.H. Choi, J.J. Park, J.D. Yang, and D.K. Lee. An object-based approach to managing domain specific thesauri: semiautomatic thesaurus construction and query-based browsing. *Technical Report TR 98/11*, Dept. of Computer Science, Chonbuk National University, 1998.

[3] W. Chuang, A. Tiyyagura, J. Yang, and G. Giuffrida. A fast algorithm for hierarchical text classification. In *Proc. of the 2nd Int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK'00)*, pp 409-418, London, UK, 2000.

[1] `http://trec.nist.gov/trec_eval`

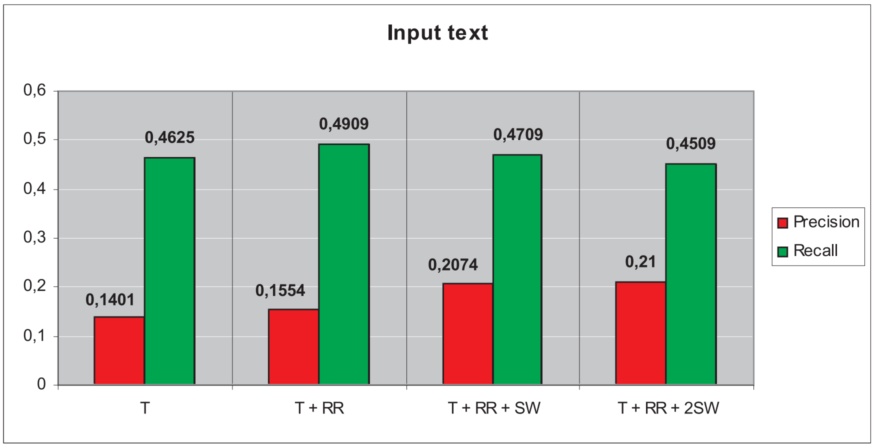


Figure 6: Evaluation of input text extraction.

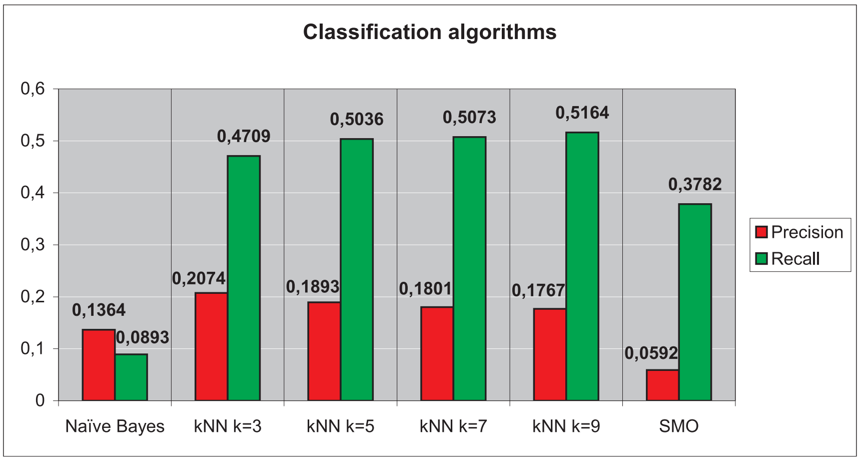


Figure 7: Evaluation of classification algorithms.

[4] S. Dumais and H. Chen. Hierarchical classification of Web content. In *Proc. of ACM-SIGIR-00, 23rd ACM Int. Conf. on Research and Development in Information Retrieval*, pp 256–263, Athens, GR, 2000

[5] G.H. John and P. Langley. Estimating Continuous Distributions in Bayesian Classifiers. In *Proc. of the Eleventh Conference on Uncertainty in Artificial Intelligence.* pp. 338-345. Morgan Kaufmann, 1995.

[6] D. Koller and M. Sahami. Hierarchically classifying documents using very few words. In *Proc. of 14th Int. Conf. on Machine Learning*, pp. 170–178, Nashville, US, 1997

[7] C.D. Manning, H. Schtze. Foundations of Statistical Natural Language Processing. *MIT Press.* 1999. ISBN 0-262-13360-1

[8] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Learning to classify text from labeled and unlabeled documents. In *Proc. of the 15th National Conference on Artifical Intelligence, AAAI-98*, 1998.

[9] J. Platt. Fast Training of Support Vector Machines using Sequential Minimal Optimization. *Advances in Kernel Methods - Support Vector Learning*, B. Schoelkopf, C. Burges, and A. Smola, eds., MIT Press, 1998.

[10] Gerald Salton. *Automatic text processing*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1988

[11] F. Sebastiani. Machine learning in automated text categorization. In *ACM Computing Surveys, 24-1*, pp. 1–47, 2002

[12] E. Tzoukermann, J. Klavans, C. Jacquemin. 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. In *Proc. of ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 148–155. 1997, Philadelphia, PA, USA

[13] I. Witten and E. Frank Data Mining: Practical machine learning tools and techniques, 2nd Ed. *Morgan Kaufmann*, San Francisco, 2005.

More Accurate Fuzzy Text Search for Languages Using Abugida Scripts

Anil Kumar Singh, Harshit Surana and Karthik Gali
Language Technologies Research Centre
International Institute of Information Technology
Hyderabad, India
anil@research.iiit.net, surana.h@gmail.com, karthikg@students.iiit.net

ABSTRACT

Text search is a key step in any kind of information access. For doing it effectively, we can use knowledge about the concerned writing systems. Methods based on such knowledge can give significantly better results for searching text, at least for some languages. This can improve information retrieval in particular and information access in general. In this paper, we present a method for fuzzy text search for languages which use Abugida scripts, e.g. Hindi, Bengali, Telugu, Amharic, Thai etc. We use characteristics of a writing system for fuzzy search and are able to take care of spelling variation, which is very common in these languages. Our method shows an improvement in F-measure of up to 30% over scaled edit distance.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Fuzzy text search*; J.m [**Computer Applications**]: Miscellaneous

General Terms

Natural language processing

Keywords

Fuzzy text search, Spelling variation, Orthographic and phonetic similarity, Writing systems

1. INTRODUCTION

Text search is the first step in information access or retrieval, without which effective information retrieval (IR) is not possible. However, for many languages, it becomes meaningful only when it is fuzzy, not literal. In this paper we present a more accurate method for this purpose. This method uses deeper information about the writing system used by a language. In this paper we do not consider other aspects of IR such as estimating the relevance of a document because the biggest problems for the languages considered in this paper are at the level of text search and they have not been adequately addressed so far.

SIGIR'07 iNEWS07 workshop, July 27, 2007, Amsterdam, The Netherlands.

Fuzzy text search is required mainly because of the widespread variation and rich morphology in many very highly used languages of the world, and sometime also because of the nature of problem requires approximate matching of strings. This variation can be spelling variation, dialectal variation or regional variation. The variants need not be 'errors': some or all of them may be acceptable (Section-4 and Figure-1). It is useful even for languages like English, but mostly for applications like spell checking or Google Suggest[1] etc. For Indian and many other languages (Section-3) on the other hand, it is unavoidable for almost any kind of information access. Masuyama and Nakagawa [19, 18] and Ohtake et al. [21] have previously discussed the importance of accounting for variants for the purpose of information access or retrieval.

Our focus in this paper is on the languages using scripts or writing systems belonging to the Abugida category (Section-3). We present a method for fuzzy text search which works much better than Scaled Edit Distance or SED [8] for these languages. Pingali et al. [23], who attempted to build a crawler called WebKhoj for the Indian languages, had also faced problems in searching text due to variation.

We propose that the idea of fuzzy text search is based on the notion of *surface similarity*, which (at least for Abugida scripts) can be roughly defined as combined orthographic and phonetic similarity. A method based on a measure of surface similarity can give better results. The Abugida scripts have characteristics (like highly phonetic nature) which can be used for designing a very effective measure of surface similarity. Our method is based on this measure.

The paper is organized as follows. In Section-2, we present a brief literary survey of some related work. Section-3 is an introduction to the Abugida and Brahmi scripts from the point of view of our work. In the same section, we also mention the Indian languages, as these are the languages on which we have evaluated our method. Section-4 is about variation which is very common in Indian languages and because of which fuzzy text search is important. Section-5 introduces the notion of *surface similarity* which is different from string similarity and is the basis of fuzzy text search. In this section, we also describe the Computational Phonetic Model of Scripts (CPMS) proposed by Singh [25], on which our measure of surface similarity and our method of fuzzy text search is based (Section-6). In Section-7, we describe the experimental setup and the evaluation of our approach. Finally, we conclude in Section-8.

[1] http://www.google.com/webhp?complete=1&hl=en

इन्फॉर्मेशन	173	నారాయణ	5190	तमिलनाडु	67,000
इन्फर्मेशन	153	నారాయణ్	128	तमिलनाडू	8,800
इन्फ़ॉर्मेशन	91	నారాయణా	125	तामिलनाडु	89
इन्फोर्मेशन	91	నారయణ	17	तमिलनाड	65
इंफॉर्मेशन	73	నారాయణ్	6	तामिलनाडू	32
इंफार्मेशन	72	నారయణ్	4		
इन्फार्मेशन	67	నారాయన	3		
इनफॉर्मेशन	45	నారాయన్	3		
इंफोर्मेशन	42	నారాయాణ	1		
इनफार्मेशन	23				
इंफ़ॉर्मेशन	6				
इनफोर्मेशन	5				
इन्फ़ार्मेशन	2				
इन्फ़ोर्मेशन	1				

Figure 1: *First Column*: Variants of a commonly used borrowed word 'information' found by searching on Google. *Second Column*: Variants of a very familiar proper noun 'Tamilnadu' (the name of one of India's states) found by searching on Google. *Third Column*: Variants of a very familiar proper noun 'Narayana' (a person name as well as the name of a god) found by searching on Google. The numbers are the results returned by the search engine for a particular variant.

2. RELATED WORK

Emeneau [9], in his classic paper 'India as a Linguistic Area' showed that there are a lot of similarities among Indian languages, even though they belong to different families. One of these similarities is that many of these languages use scripts derived from Brahmi.

There has been a lot of linguistic work on writing systems [4, 7, 33] from the linguistic point of view. An example of work relevant to computation is a computational theory of writing systems by Sproat [31]. Sproat also studied the Brahmi scripts [29] and presented a formal computational analysis of Brahmi scripts [30].

The development of a standard for Brahmi origin scripts [1, 3], called Indian Standard Code for Information Interchange (ISCII) can also be mentioned here. This super-encoding [16] takes into account some of the similarities among the alphabets of Brahmi origin scripts. This is why ISCII has been used as the basis for the 'model of alphabet', which is a part of the Computational Phonetic Model of Scripts [25]. *Om* transliteration scheme [11] also provides a script representation which is common for all Indian languages. The display and input is in human readable Roman script. Transliteration is partly phonetic.

There has also been work on phonetic modeling of graphemes. For example, Rey et al. [24] argued that graphemes are perceptual reading units and can thus be considered the minimal 'functional bridges' in the mapping between orthography and phonology. Black et al. [2] discuss some issues in building general letter to sound rules within the context of speech processing. Galescu and Allen [10] present a statistical model for language independent bidirectional conversion between spelling and pronunciation, based on joint grapheme/phoneme units extracted from automatically aligned data. Daelemans and Bosch [5] describe another method for the same. Killer [14] has tried building a grapheme based speech recognition as a way to build large vocabulary speech recognition systems. Kopytonenko et al. [15] also focussed on computational models that perform grapheme-to-phoneme conversion.

Two of the best known methods for approximate string matching are the SOUNDEX algorithm [6] and the double metaphone algorithm [22]. The latter uses some information about the phonetic values of letters.

Loan words and spelling variations in a corpus or on the Web create a problem for information retrieval. A previous work on solving this problem was by Li et al. [17]. It involved spelling correction of the query based on distributional similarity. A work on extraction of spelling variants for loan words in Japanese [19] used a large corpus and contextual similarities. Since Indian languages are lacking in large resources these methods may not be very applicable.

Singh [25] had proposed a computational phonetic model of Brahmi scripts based on orthographic and phonetic features. These features were defined based on the characteristics of the scripts. The similarity between two letters was calculated using an SDF and the algorithm used for 'aligning' two strings was dynamic time warping (DTW). This model tries to relate letters with phonetic and orthographic features in a way that allows some fuzziness by using linguistic knowledge about the writing systems. It has been used for shallow morphological analysis [26], study of cognates among Indian languages [27] and comparative study of languages using text corpora [28].

3. SCRIPTS AND LANGUAGES

Abugida is a term for a type of scripts such as those used by most of the major languages of the Indian subcontinent. In fact, about half of the writing systems used in the world belong to this category[2]. Such scripts are also sometimes called *alphasyllabary* or *syllabics* because one the basic unit in these scripts more or less corresponds to a syllable, even though these scripts also have alphabets. A consonant in these scripts is implicitly associated with a vowel, which means that absence (rather than presence) of a vowel after a consonant has to be indicated explicitly. Another major characteristic of these scripts is that the letters have a very close and almost unambiguous correspondence with phonetic features. Some other important (graphemic) char-

[2] http://en.wikipedia.org/wiki/Abugida

acteristics are about the way letters are written together, but since these characteristics have more to do with shapes, we will not discuss them. We will only consider characteristics relevant for electronic text, i.e. encoded text where letters have integer codes. The shapes assigned to them are relevant only for rendering, not for text processing.

The most important family of Abugida scripts is the Indic or Brahmi family [13]. The most well known Brahmi script is perhaps Devanagari, which is used for Hindi, Sanskrit, Marathi, Nepali and many other languages. These have originated from the ancient Brahmi script which was used for Sanskrit, Pali, Prakrit etc. The important point is that they have retained many characteristics of Brahmi which are crucial for the method we are presenting in this paper. Some of these characteristics can be summarized as:

- Close correspondence among letters and phonetic features (Figure-2)
- The main unit of the script corresponds closely to a syllable
- The letters are organized very systematically in the alphabet, in such a way that letter positions indicate phonetic and orthographic features
- The arrangement of letters in the alphabet is common among all the Brahmi origin scripts, even if letter shapes seem to be completely different
- It is possible to use a common super-encoding like ISCII [1] for all these scripts

The Indian or South Asian subcontinent is home to hundreds of languages belonging to different linguistic families. However, most of the major Indian languages fall within two families: Indo-Aryan and Dravidian [12]. And most of these languages use Brahmi origin scripts. In fact, many languages of other areas also used these scripts, e.g. Thai, Laotian, Cambodian (Khmer) etc. The Indian or South Asian languages alone account for more than one billion people. In terms of number of speakers, at least three or four Indian languages are usually placed among the top ten most heavily used languages of the world [32]. Some of these languages are: Hindi/Urdu, Bengali, Telugu, Punjabi, Tamil, Malayalam, Kannada, Marathi, Gujarati, Oriya, Assamese. Our method works for all these languages.

Two characteristics of Indian languages are very relevant for the present work. The first is their rich morphology, which makes processing of verbs (and sometimes even nouns) much more difficult, even for relatively easy problems like stemming. The second is lack of standardization, due to which variation is very common in text written in these languages.

4. VARIATION AND FUZZY SEARCH

The problem of spelling variants in Indian Languages is somewhat similar to that in East Asian Languages. For example, in Japanese, the Katakana variants cause a lot of problems in information retrieval, text summarization, machine translation and question-answering.

To give an indication of the extent of the problem, we conducted a small experiment. We took one highly used English word (‘information’) borrowed into Hindi and one very familiar (to Indians) proper noun (‘Tamilnadu’) and searched them among the Hindi (UTF-8) documents on Google. Then we tried to search all the possible variations of these words and noted down the number of results returned by the search engine. These are shown in Figure-1. Note the large number of variations in spite of the fact that the amount of Hindi text in UTF-8 on the Web is nowhere near the text in English, which means that the ‘Web as corpus’ in Hindi (in UTF-8 encoding) is very small in size.

Fuzzy text search (as opposed to literal text search) is needed to take care of the variation mentioned in the previous section. The computational method used for this purpose should be able to take into account the usual phenomenon in string variation like deletions, additions, substitutions, etc. But more importantly, the method should be able to give scores for these phenomenon such that all the available information is used. For example, if we know that /t/ is more similar to /d/ than to /f/, then the similarity score for matching two strings should reflect this fact. Abugida scripts allow this (and many other such things) to be done easily because of their characteristics described earlier. And our method does this more thoroughly than other methods.

The possible variants of a word are usually not arbitrary. They follow some phonetic or orthographic principles (e.g., /t/ is more likely to become /d/ then /f/) and these principles are closely tied to the nature of the scripts, at least in the case of Abugida scripts. This is why we can use a much better way of finding out how similar two strings are.

5. SURFACE SIMILARITY AND CPMS

Surface similarity is a kind of string similarity which is deeper (despite the name) than literal string similarity. More specifically, it includes some linguistic knowledge about the units of a script. It is different from similarities based on edit distance. We are calling it *surface* similarity even though it is a deeper similarity because it doesn’t include semantic similarity. We are still talking about similarity of the surface forms, not their meanings.

The notion of *surface similarity* can be applicable wherever string similarity is applicable, but it is an especially more suitable idea for natural language processing applications. If we can find a good method to calculate such similarity, we can have much better fuzzy text search. The method used by us is based on the Computational Phonetic Model of Scripts [25].

5.1 Computational Phonetic Model of Scripts

Given the similarities among the alphabets of Brahmi origin scripts and the fact that these scripts have phonetic characteristics, it is possible to build a phonetic model for these scripts. We have used a modified version of the Computational Phonetic Model of Scripts (CPMS) proposed by Singh [25]. The phonetic model tries to represent the sounds of Indian languages and their relations to the letters. It includes phonetic or articulatory features, some orthographic features, numerical values of these features, and a distance function to calculate how phonetically similar two letters are. The scripts covered by this model are: Devanagari (Hindi, Marathi, Nepali), Bengali (Bengali and Assamese), Gurmukhi (Punjabi), Gujarati, Oriya, Tamil, Telugu, Kannada, and Malayalam.

The CPMS itself consists of the model of alphabet, the model of phonology and the SDF. The core of the model of

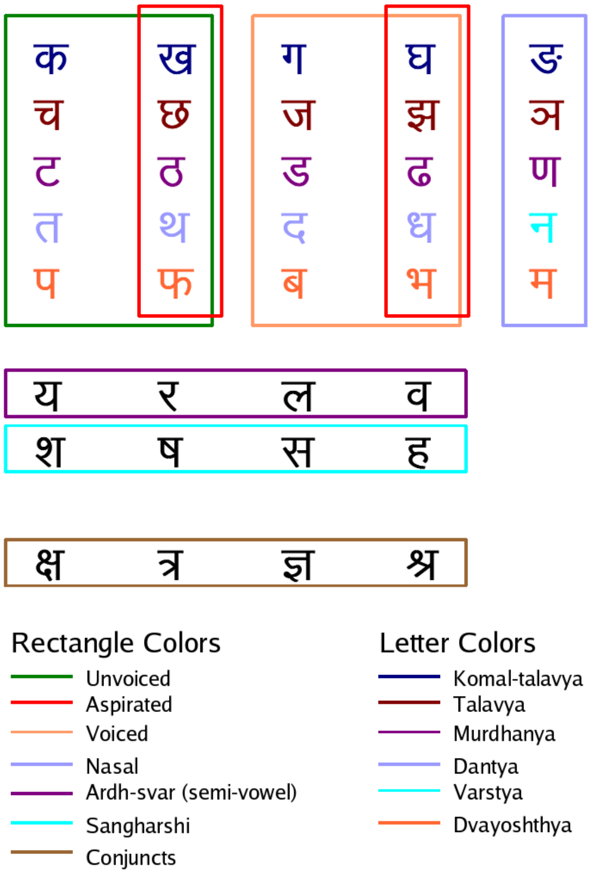


Figure 2: Phonetically arranged basic consonants in the unified Brahmi alphabet. The vowels also have a systematic arrangement.

phonology is the definition of phonetic features (table-1) and the numerical values assigned to them. The CPMS assigns a mostly phonetic representation for each ISCII letter code in terms of the phonetic and orthographic features. For example, vowel *o* and consonant *n* will be represented as:

176 → [type=**v**, voiced=**t**, length=**s**, vowel2=**m**,
vowel1=**m**, height=**b**]
198 → [type=**c**, voiced=**t**, place=**v**, manner=**n**

5.2 Model of Alphabet

The model of alphabet is meant to cover all the alphabets of the related scripts, but it may be more than a superset of these alphabets. By 'model of alphabet' we essentially mean a meta alphabet, i.e., number of letters and their arrangement, including the basis of this arrangement. It is a conceptual view of the alphabet and also includes a representation based on this view. Of course, this model will be applicable for only those scripts which have an alphabet.

Since Brahmi origin scripts have a very well organized alphabet with arrangement of letters based on phonetic features, and also because these alphabets are very similar, it is possible and very useful to have a unified model of alphabet for these scripts. Such a model can simplify computational processing in a multilingual environment, e.g. in our case it allows us to use the same setup for all the languages which Brahmi origin scripts.

The phonetic nature of Brahmi based alphabets can be seen in the following properties (see Figure-2 too):

- Letters neatly arranged on phonetic basis
- Vowels and consonants separated
- Consonants themselves separated on the basis of phonetic features

This is evident from the fact that if the alphabet is written in the usual conventional way on paper (Figure-2), we can draw rectangles around consonants such that each rectangle represents a particular articulatory feature. The CPMS makes explicit, in computational terms, the phonetic (as well as orthographic) characteristics of the letters in this unified alphabet by mapping the letters to a set of feature and their (numerical) values.

5.3 Stepped Distance Function (SDF)

To calculate the orthographic and phonetic similarity between two letters, we use a stepped distance function (SDF). Since phonetic features differentiate between two sounds (or the letters representing them) in a cascaded or hierarchical way, the SDF calculates similarity at several levels. For example, the first level compares the type (vowel, consonant, punctuation etc.). There is a branching at the second level and, depending on whether the letters being checked

Feature	**Possible Values**
Type	**C**onsonant, **V**owel, Vowel **m**odifier, Nu**k**ta, **N**umber, **P**unctuation, **H**alant, **U**nused
Height	**F**ront, **M**id, **B**ack
Length	**L**ong, **S**hort, **M**edium
Svar1	**L**ow, Lower Mi**d**dle, Upper, **M**iddle, Lower Hi**g**h, **H**igh
Svar2	**S**amvrit, Ardh-Sa**m**vrit, Ardh-V**i**vrit, **V**ivrit
Place	Dva**y**oshthya (Bilabial), Dant**o**shthya (Labio-dental), **D**antya (Dental), **V**arstya (Alveolar) **T**alavya (Palatal), **M**urdhanya (Retroflex), **K**omal-Talavya (Velar), **J**ivhaa-Muliya (Uvular), **S**varyantramukhi (Pharynxial)
Manner	**S**parsha (Stop), **N**asikya (Nasal), **P**arshvika (Lateral), P**r**akampi (Voiced), San**g**harshi (Fricative), Ardh-S**v**ar (Semi-vowel)

Table 1: Non-Boolean Phonetic Features

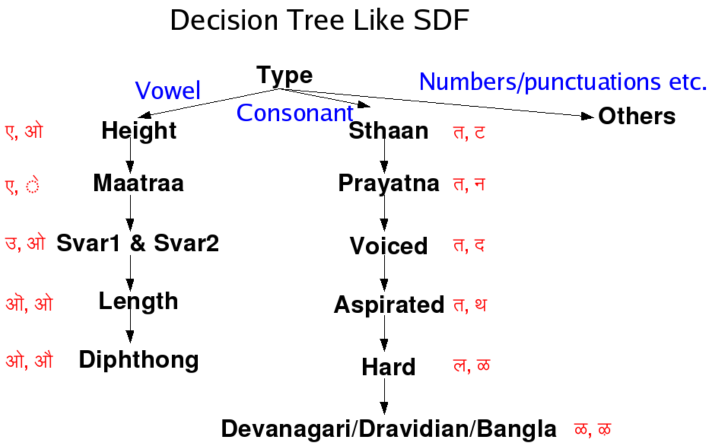


Figure 3: Stepped distance function: various steps differentiate between different kinds of letters. At the end, a quantitative estimate of the orthographic and phonetic distance is obtained.

are both vowels or consonants, further comparison is done based on the significant feature at that level: height in the case of vowels and *sthaan* (place) in the case of consonants. At the third level, values of *maatraa* and *prayatna* (manner), respectively, are compared. Thus, each step is based on the previous step. The weights given to feature values are in the non-decreasing order. The highest level (type) has the highest weight, whereas the lowest level (diphthong, for vowels) has the lowest weight. This process (somewhat simplified) is shown in figure-3.

6. MEASURING SURFACE SIMILARITY

In this section we will first formally define surface similarity measure and then describe a method to use this measure with reference to the background given in the previous sections.

6.1 Surface Similarity Measure

Surface similarity measure is a fuzzy measure of similarity between two strings or words. As mentioned earlier, it includes knowledge about the scripts. Formally, we can define this measure for Abugida scripts as follows:

$$S_s \;=\; f(w_1, w_2, A, W, W_n, P, P_n, D) \qquad (1)$$

where f is a function representing an alignment algorithm, w_1 and w_2 are the two words or strings to be compared, A is the alphabet, W is the set of orthographic features, P is the set of phonetic features, W_n and P_n are the sets of numerical values assigned to the orthographic and phonetic features, and D is a distance function for calculating the similarity between two letters.

To relate the parameters to the preceding and the following sections, f represents the modified DTW algorithm used by us, A represents the model of alphabet, W and P represent the orthographic and phonetic features (the model of phonology) and D represents the SDF. Note that D can itself be defined as:

$$D \;=\; f(l_1, l_2, A, W, W_n, P, P_n) \qquad (2)$$

where l_1 and l_2 are the two letters being compared as part of the alignment algorithm.

Another important point here is that this formulation allows a lot of flexibility with respect to the model of alphabet, the way orthographic and phonetic features are designed, the numerical values given to them, the distance function used to calculate the similarity of two letter, and the alignment algorithm used to align the strings or words. Therefore, the method used by us is, strictly speaking, just one instance of this type of methods. In other words, there is a scope of a lot of exploration here.

	Hindi		Telugu	
	SED	**CPMS**	**SED**	**CPMS**
Precision	53.22%	94.16%	42.58%	83.67%
Recall	76.76%	94.90%	59.87%	71.52%
F-Measure	62.86%	94.53%	49.77%	77.12%
Threshold	0.4	0.9	0.2	1.0

Table 2: Comparison of results for fuzzy text search. SED stands for a method based on a measure of string similarity called Scaled Edit Distance. CPMS stands for our method using a measure of surface similarity based on the Computational Phonetic Model of Scripts. These results are for those thresholds which gave the best performance for a particular Language-Method pair. Note that the thresholds of SED and CPMS are not directly comparable.

6.2 Method of Fuzzy Text Search

Once a surface similarity measure is defined, fuzzy text search is just a matter of setting up a threshold and finding the matches with similarity scores S_s lower than (or higher than, depending upon the way scores are calculated) the threshold t. Most of the detail has already been presented in the preceding sections. The only thing that remains to be described is the modified DTW algorithm used by us.

6.3 Modified DTW Algorithm

The DTW algorithm [20] is heavily used in speech recognition and for problems like gene sequencing. Our version of this algorithm can be roughly described as follows:

```
Let the query string be Sq
Let the retrieval string be Sr

m = stringLength(Sq)
n = stringLength(Sr)

initMatrix DTW[m,n]

for i = 1 to n
 for j = 1 to m
   cost = SDF[Sq[i], Sr[j]] * K(i,j)

   DTW[i,j] = min(DTW[i-1, j] + cost,// insertion
    DTW[i, j-1] + cost,             // deletion
    DTW[i-1, j-1] + cost)           // substitution
```

Here, $K(i,j)$ is a heuristic function which can take into account language specific issues like the inflectional nature of a language, e.g. giving the last two characters (which are most likely to represent an inflection) a lesser weight for Hindi. $SDF[Sq[i], Sr[j]]$ is the cost between two letters $Sq[i]$ and $Sr[j]$ of the two strings which are being compared at a particular node in the trellis. This is the basic formulation of our modified DTW algorithm. However, several optimization techniques were used to increase the speed of the algorithm, including trie based search.

7. EVALUATION

Since there was no standard data set over which we could perform our experiments, we randomly selected words from a corpus consisting of documents obtained by crawling the Web. We randomly selected 400 words each from Hindi and Telugu. For each word we collected all the possible spelling variants. Some words did not have spelling variants, so we dropped them from our data set. In case of uncertainties about spelling variations, we verified them by checking their document level contexts. We were left with 318 Hindi words with 1020 variant pairs. For Telugu, 202 words were left with 674 variants. We tested our algorithm on this data set.

Since we could not find any algorithm based on phonetic matching for Indian languages, we used a scaled version of the Levenshtein distance[3] [8]. Such a version has been used in various applications including cognate alignment and dialectology. Edit distances in general have been used in many other applications including spell checking and identifying spelling variants.

Scaled Edit Distance (SED) is an edit distance which is scaled with the sum of the lengths of words under consideration. The advantage of scaling is that it alleviates the disparity between long words in comparison to short words, which is a problem in simple edit distances.

If ED is an edit distance between two words w_1 and w_2 (with lengths $|w_1|$ and $|w_1|$, respectively), then SED can be defined as:

$$SED(w_1, w_2) = \frac{2 * ED(w_1, w_2)}{|w_1| + |w_2|} \tag{3}$$

To evaluate our algorithm we performed fuzzy search of the words in our test set word list (318 Hindi, 202 Telugu) over the words from entire web corpus that we had. We compared the list returned by our fuzzy text search to our reference variant pair list (1020 Hindi, 674 Telugu). This allowed us to calculate precision, recall and F-measure. We tried various thresholds to select the one which gives the maximum F-measure. We did a similar experiment on SED.

The results of the evaluation are given in table-2. The performance of both the methods for Hindi and Telugu has been plotted against the threshold in Figures-4 to 7. As can be seen from the results, our method outperforms the method based on SED by up to or even more than 30%. The results are somewhat lower for Telugu. This is explained by the fact that Telugu is a more agglutinative language than Hindi and has a richer morphology.

The plots against thresholds indicate that for both the methods there is a lower value of threshold up to which performance (F-measure) increases. Beyond this value, there is not much increase in performance, as the F-measure more or less stabilizes.

Another objective way to compare the performance of the two methods would be to look at the F-measure at the point (on the plots shown in Figures 4-7) at which precision and recall lines cross (say, $P = R$ point). As is clear from the

[3] http://www.merriampark.com/ld.htm

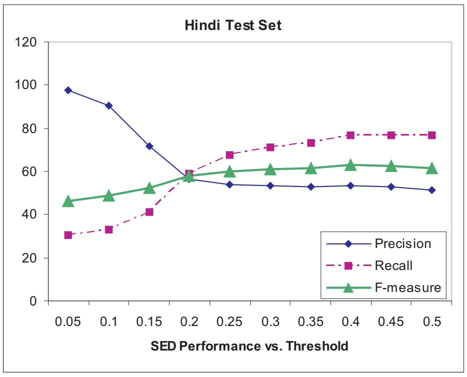


Figure 4: Performance of SED for Hindi plotted against threshold.

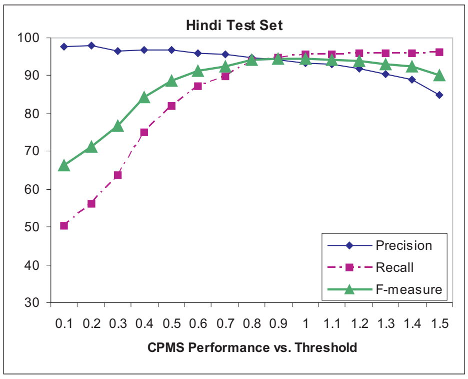


Figure 5: Performance of CPMS for Hindi plotted against threshold.

graphs, our method performs significantly better than SED for both Hindi and Telugu.

An interesting observation is that precision is more stable after the $P = R$ point in the case of SED, but in the case of CPMS it is more stable *before* the $P = R$ point. However, recall has similar behavior for both the approaches in these terms. This might have important implications for practical applications where a trade-off is to be achieved between precision and recall and we might not know where exactly the $P = R$ point lies.

8. CONCLUSION

We argued in this paper that fuzzy text search is an important, unavoidable problem for languages which use Abugida scripts. We presented a more accurate method of fuzzy text search for Indian languages. We also introduced the notion of *surface similarity*. In our opinion, fuzzy text search is based on a measure of surface similarity. For Abugida scripts (which include Brahmi origin scripts), surface similarity can be defined roughly as combined orthographic and phonetic similarity. Our method for calculating surface similarity uses a Computational Phonetic Model of Scripts (CPMS) and thereby takes into account the characteristics of Brahmi origin scripts. Moreover, the same setup can be used for all the languages which use Brahmi origin scripts. We were able to improve results (in terms of F-measure) for some Indian languages by up to 30% over scaled edit distance. Based on the experiments for various thresholds, some observations were reported with regard to the trade-off between precision and recall.

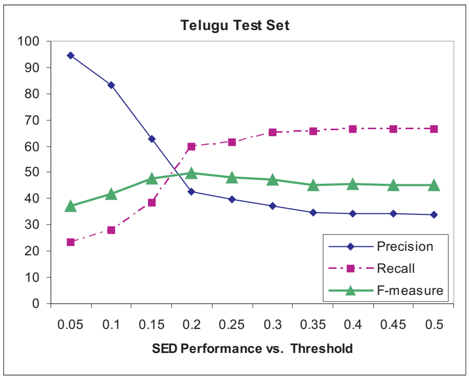


Figure 6: Performance of SED for Telugu plotted against threshold.

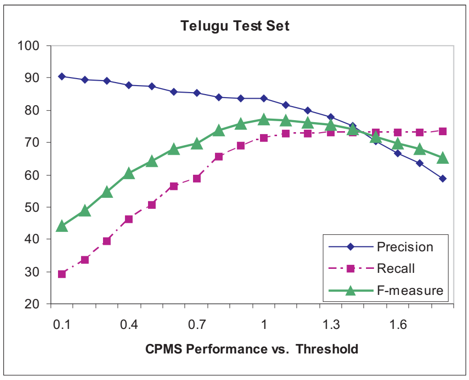


Figure 7: Performance of CPMS for Telugu plotted against threshold.

An interesting question is whether the method described in this paper can be applied to or adapted for other kinds of scripts. This should be possible for scripts like Hangul because Hangul too is a 'phonemic alphabet organized into syllabic blocks'[4]. For Latin like scripts, it might be a bit harder, and even more hard for logographic or ideographic scripts. This can be a good area for further work.

9. REFERENCES

[1] BIS. Indian standard code for information interchange (iscii), 1991.

[2] A. Black, K. Lenzo, and V. Pagel. Issues in building general letter to sound rules. In *ESCA Synthesis Workshop, Australia.*, pages 164–171, 1998.

[3] C-DAC. Standards for indian languages in it, 2006a. http://www.cdac.in/html/ gist/standard.asp.

[4] F. Coulmas. *Writing Systems: An Introduction to their Linguistic Analysis*. Cambridge University Press, 2003.

[5] W. Daelemans and A. van den Bosch. A language-independent, data-oriented architecture for

[4] http://en.wikipedia.org/wiki/Hangul

grapheme-to-phoneme conversion. In *Proceedings of ESCA-IEEE'94*, 1994.

[6] F. Damerau and E. Mays. A technique for computer detection and correction of spelling errors. In *Communications of the ACM, 7(3):171176*, 1964.

[7] P. Daniels and W. Bright. *The World's Writing Systems.* Oxford University Press, New York, 1996.

[8] T. M. Ellison and S. Kirby. Measuring language divergence by intra-lexical comparison. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia, 2006. Association for Computational Linguistics.

[9] M. B. Emeneau. India as a linguistic area. In *Linguistics 32:3-16*, 1956.

[10] L. Galescu and J. Allen. Bi-directional conversion between graphemes and phonemes using a joint n-gram model. In *Proceedings of the 4th ISCA Tutorial and Research Workshop on Speech Synthesis*, 2001.

[11] M. Ganapathiraju, M. Balakrishnan, N. Balakrishnan, and R. Reddy. OM: One Tool for Many (Indian) Languages. *ICUDL: International Conference on Universal Digital Library, Hangzhou*, 2005.

[12] R. G. Gordon. Ethnologue: Languages of the world, fifteenth edition (ed.), 2005a. Online version: http://www. ethnologue.com/web.asp.

[13] R. Ishida. An introduction to indic scripts. In *Proceedings of the 22nd Int. Unicode Conference*, 2002.

[14] M. Killer, S. Stker, and T. Schultz. Grapheme based speech recognition, 2003.

[15] M. Kopytonenko, K. Lyytinen, and T. Krkkinen. Comparison of phonological representations for the grapheme-to-phoneme mapping. In *Constraints on Spelling Changes: Fifth International Workshop on Writing Systems*, Nijmegen, The Netherlands, 2006.

[16] LDC. Found resources: Hindi, 2003. http://lodl.ldc.upenn.edu/found.cgi? lan=HINDI.

[17] M. Li, M. Zhu, Y. Zhang, and M. Zhou. Exploring Distributional Similarity Based Models for Query Spelling Correction. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia, 2006. Association for Computational Linguistics.

[18] T. Masuyama and H. Nakagawa. Web-based acquisition of japanese katakana variants. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 338–344, New York, NY, USA, 2005. ACM Press.

[19] T. Masuyama, S. Sekine, and H. Nakagawa. Automatic Construction of Japanese KATAKANA Variant List from Large Corpus. *Proceedings of the 20th International Conference on Computational Linguistics (COLING04)*, 2:1214–1219, 2004.

[20] C. S. Myers and L. R. Rabiner. A comparative study of several dynamic time-warping algorithms for connected word recognition. In *The Bell System Technical Journal, 60(7)*, pages 1389–1409, 1981.

[21] K. Ohtake, Y. Sekiguchi, and K. Yamamoto. Detecting transliterated orthographic variants via two similarity metrics. *Proceedings of the 20th international conference on Computational Linguistics*, 2004.

[22] L. Philips. The double metaphone search algorithm. In *C/C++ Users Journal, vol. 18, no. 5*, 2000.

[23] P. Pingali, J. Jagarlamudi, and V. Varma. WebKhoj: Indian language IR from multiple character encodings. *Proceedings of the 15th international conference on World Wide Web*, pages 801–809, 2006.

[24] A. Rey, J. C. Zieglerc, and A. M. Jacobse. Graphemes are perceptual reading units. In *Cognition 74*, 2000.

[25] A. K. Singh. A computational phonetic model for indian language scripts. In *Constraints on Spelling Changes: Fifth International Workshop on Writing Systems*, Nijmegen, The Netherlands, 2006b.

[26] A. K. Singh and H. Surana. Using a model of scripts for shallow morphological analysis given an unannotated corpus. *Workshop on Morpho-Syntactic Analysis, Pathum Thani, Thailand*, 2007a.

[27] A. K. Singh and H. Surana. Study of cognates among south asian languages for the purpose of building lexical resources. In *Proceedings of National Seminar on Creation of Lexical Resources for Indian Language Computing and Processing*, Mumbai, India, 2007c.

[28] A. K. Singh and H. Surana. Can corpus based measures be used for comparative study of languages? In *Proceedings of the ACL Workshop Computing and Historical Phonology*, Prague, Czech Republic, 2007d.

[29] R. Sproat. Brahmi scripts. In *Constraints on Spelling Changes: Fifth International Workshop on Writing Systems*, Nijmegen, The Netherlands, 2002.

[30] R. Sproat. A formal computational analysis of indic scripts. In *International Symposium on Indic Scripts: Past and Future*, Tokyo, Dec. 2003.

[31] R. Sproat. *A Computational Theory of Writing Systems.* Nijmegen, The Netherlands, 2004.

[32] Wikipedia. List of languages by number of native speakers, 2006b. http://en.wikipedia.org/wiki/ List_of_languages_by_number_of_native_speakers.

[33] Wikipedia. Writing system, 2006c. http://en.wikipedia.org/wiki/Writing_system.