CLIA 2008

2nd International Workshop on

Cross Lingual Information Access (CLIA)

Addressing the Information Need of Multilingual Societies

The Third International Joint Conference On Natural
Language Processing IJCNLP 2008

Proceedings of the Workshop

11 January 2008, Hyderabad, India

Copyright Transfer Agreement

Preface

Welcome to the second international workshop on Cross Lingual Information Access (CLIA 2008), with a focus on "Addressing the Information Need of Multilingual Societies".

In this workshop, like in the previous year, our aim was to bring together various trends in cross and multi-lingual information retrieval and access. This year we have accepted eight papers after a careful review process and these accepted papers are included in the proceedings.

The workshop will have four sessions, each focusing on a specific theme: Cross Language Information Retrieval, Translations and Transliterations in CLIR, Information Extraction/Summarization in CLIR contexts, and, finally a session on the overview of the experiences of Indian research groups in the CLEF-2007 competition.

There are three papers in the first session on Cross Language Information Retrieval:
The first paper explores the effects of language relatedness on multilingual Information retrieval. This paper presents a case study with Indo-European and Semitic Languages and addresses some of the challenges posed by Semitic languages IR. The paper on Identifying Similar and Co-referring Documents Across Languages, authors make use of Vector Space Model (VSM) and Named Entities in identifying the co-reference and similarity. In the paper on finding parallel texts on the web using cross-language information retrieval, CLIR techniques are used in combination with structural features to retrieve candidate document pairs from the web. These three papers are part of the session on Cross Language Information Retrieval.

In the second session on Translations and Transliterations in CLIR, we will again have three papers will be presented: The first paper presents results of some experiments in Mining Named Entity Transliteration Pairs from Comparable Corpora, employing English-Tamil named entity parallel comparable corpus texts. The second paper on Domain-Specific Query Translation for Multilingual Information Access using Machine Translation Augmented with Dictionaries Mined from Wikipedia authors demonstrates that effective query translation for CLIA can be achieved in the domain of cultural heritage using a standard MT system, and that domain specific phrase dictionaries that are may be automatically mined from the online Wikipedia. The paper Statistical Transliteration for Cross Language Information Retrieval using HMM alignment model and CRF, presents a technique that combines HMM and CRF for transliteration task in CLIR.

In the third session we have two papers. The first paper is Script Independent Word Spotting in Multilingual Documents, which describes a system that accepts a query in the form of text from the user and returns a ranked list of word images from document image corpus based on similarity with the query word. The second paper is about building a document graph based multi-document summarizer that makes use of a graph model at offline processing time as well as the query time.

Finally, in addition to all the refereed papers, we have six invited presentations by various teams focusing on Indian language CLIR. These presentations are based on the work done by these teams for Ad-hoc task in Cross Language Evaluation Forum (CLEF) in 2007. Teams from IIT Bombay (focusing Marathi, Hindi), IIT Kharagpur (Bengali and Hindi), IIIT Hyderabad (Telugu and Hindi), Microsoft Research India (Tamil, Telugu and Hindi) and Jadhavpur University (Bengali, Telugu and Hindi) will present their work to achieve CLIR for queries in Indian languages and documents in English. In this special session, a team from ISI, Kolkata will make a presentation on FIRE (Forum for Information Retrieval Evaluation), a proposed cross language evaluation forum, specifically for Indian languages. Abstracts of these presentations are also included in these proceedings.

We would like to thank all authors for the hard word that they have put in, in submission, rework and presentation. The workshop would not be possible without them. We would also like to thank the program committee and all the reviewers for their valuable feedback. We hope you would enjoy the workshop.

"We would like to thank Minhaj Babji for all his help in preparing these proceedings as well as supporting the organizing committee during all phases of the workshop."

Vasudeva Varma,
Pushpak Bhattacharya,
Sivaji Bandyopadhyay,
A. Kumaran,
Sudeshna Sarkar.

(Editors CLIA 2008 Workshop)

Committees

Organizing Committee

Vasudeva Varma, IIIT Hyderabad, India
Pushpak Bhattacharya, IIT Bombay, India

Sudeshna Sarkar, IIT Kharagpur, India

A Kumaran Microsoft Research, India

Sivaji Bandyopadhyay, Jadavpur University, Kolkata, India

Program Committee

Asanee Kawtrakul, Kasetsart University, Bangkok, Thailand
Carol Peters, Istituto di Scienza e Tecnologie dell'Informazione and CLEF campaign, Italy

Gilles Serasset, GETALP-LIG, Grenoble, France

Kumaran A, Microsoft Research, Bangalore, India

Lucy Vanderwende, Microsoft Research, USA

Mandar Mitra, ISI Kolkata, India

Paolo Rosso, Universidad Politecnica de Valencia (UPV), Spain

Patrick Saint Dizier, IRIT, Universite Paul Sabatier, Toulouse, France

Paul McNamee, Johns Hopkins University, USA

Petri Myllymaki, University of Helsinki, Finland

Pushpak Bhattacharya, IIT Bombay, India

Ralf Steinberger, European Commission - Joint Research Centre, Italy

Sivaji Bandyopadhyay, Jadavpur University, Kolkata, India

Sobha L, AU-KBC, Chennai, India

Sudeshna Sarkar, IIT Kharagpur, India

Vasudeva Varma, IIIT Hyderabad, India

Workshop Program
11 January 2008, Hyderabad, India

08:45-09:00 Workshop Introduction and Opening Remarks

09:00-10:30 Session-1
Cross Language Information Retrieval

The Effects of Language Relatedness on Multilingual Information Retrieval: A Case Study With Indo-European and Semitic Languages
Peter Chew and Ahmed Abdelali.

Identifying Similar and Co-referring Documents Across Languages
Pattabhi R K Rao T and Sobha L.

Finding parallel texts on the web using cross-language information retrieval
Achim Ruopp and Fei Xia.

10:30 - 11:00 Tea Break

11:00 - 12:30 Session II
Translation and Transliteration in CLIR

Some Experiments in Mining Named Entity Transliteration Pairs from Comparable Corpora
K Saravanan and A Kumaran.

Domain-Specific Query Translation for Multilingual Information Access using Machine Translation Augmented With Dictionaries Mined from Wikipedia
Gareth Jones, Fabio Fantino, Eamonn Newman and Ying Zhang.

Statistical Transliteration for Cross Language Information Retrieval using HMM alignment model and CRF
Prasad Pingali, Suryaganesh, Sreeharsha Yella and Vasudeva Varma.

12:30 - 14:00 Lunch Break

14:00 - 15:00 Session III
Cross Language Information Access and Evaluation

Script Independent Word Spotting in Multilingual Documents
Anurag Bhardwaj, Damien Jose and Venu Govindaraju.

A Document Graph Based Query Focused Multi-Document Summarizer
Sibabrata Paladhi and Sivaji Bandyopadhyay.

15:00 - 15:30 Tea Break

15:30 - 17:30 Session IV

CLIR in Indian Languages - Invited Talks

Hindi and Marathi to English Cross Language Information Retrieval
Manoj Kumar Chinnakotla, Sagar Ranadive, Om P. Damani and Pushpak Bhattacharyya

Bengali and Hindi to English CLIR Evaluation
Debasis Mandal, Sandipan Dandapat, Mayank Gupta, Pratyush Banerjee, Sudeshna Sarkar

Bengali, Hindi and Telugu to English Ad-hoc Bilingual task
Sivaji Bandyopadhyay, Tapabrata Mondal, Sudip Kumar Naskar, Asif Ekbal, Rejwanul Haque, Srinivasa Rao Godavarthy

Cross-Lingual Information Retrieval System for Indian Languages
Jagadeesh Jagarlamudi and A Kumaran

Hindi and Telugu to English CLIR using Query Expansion
Prasad Pingali, Vasudeva Varma

FIRE: Forum for Information Retrieval Evaluation
Mandar Mitra and Prosenjit Majumdar.

17:30 - 17:45 Conclusions and Closing Remarks

Table of Contents

CLIR in Indian Languages - Invited Talks

The Effects of Language Relatedness on Multilingual Information Retrieval: A Case Study With Indo-European and Semitic Languages

Peter A. Chew
Sandia National Laboratories
P. O. Box 5800, MS 1012
Albuquerque, NM 87185-1012, USA
pchew@sandia.gov

Ahmed Abdelali
New Mexico State University
P.O. Box 30002, Mail Stop 3CRL
Las Cruces, NM 88003-8001, USA
ahmed@crl.nmsu.edu

Abstract

We explore the effects of language relatedness within a multilingual information retrieval (IR) framework which can be deployed to virtually any language, focusing specifically on Indo-European versus Semitic languages. The Semitic languages present unique challenges to IR for a number of reasons, so we set out to answer the question of whether cross-language IR for Semitic languages can be boosted by manipulation of the training data (which, in our framework, includes multilingual parallel text, some of which is morphologically analyzed). We attempted three measures to achieve this: first, the inclusion of genetically related (i.e., other Semitic) languages in the training data; second, the inclusion of non-related languages sharing the same script, and third, the inclusion of morphological analysis for Semitic languages. We find that language relatedness is a definite factor in boosting IR precision; script similarity can probably be ruled out as a factor; and morphological analysis can be helpful, but – perhaps paradoxically – not necessarily to the languages which are subjected to morphological analysis.

1 Introduction

In this paper, we consider how related languages fit into a general framework developed for multilingual cross-language information retrieval (CLIR). Although this framework can deal with virtually any language, there are some special considerations which make related languages more interesting for exploration. Taking one example, Semitic languages are distinguished by their complex morphology, a characteristic which presents challenges to an information retrieval model in which terms (usually, separated by white space or punctuation) are implicitly treated as individual units of meaning. We consider three possible methods for investigating the phenomena. In all cases, we keep the overall framework the same but simply make changes to the training data.

One method we consider is to augment the training data with text from related languages; we compare results obtained from using Semitic languages with those obtained when non-Semitic languages are used. The other two relate to morphological analysis: the second is to replace inflected forms (in just one language, Arabic) with just the root in the training data; and the third is to remove vowels (again in just one language, Hebrew).

The paper is organized as follows. Section 2 describes our general framework, which is a standard one used for CLIR. At a high level, section 3 outlines some of the challenges Semitic languages present within the context of our approach. In section 4, we compare results from using a number of different combinations of training data with the same test data. Finally, we conclude on our findings in section 5.

2 The Framework

2.1 General description

The framework that we use for IR is multilingual Latent Semantic Analysis (LSA) as described by Berry et al. (1994:21, and used by Landauer and Littman (1990) and Young (1994). A number of different approaches to CLIR have been proposed; generally, they rely either on the use of a parallel corpus for training, or translation of the IR query. Either or both of these methods can be based on the use of dictionaries, although that is not the approach that we use.

In the standard multilingual LSA framework, a term-by-document matrix is formed from a parallel aligned corpus. Each ‘document’ consists of the concatenation of all the languages, so terms from all languages will appear in any given document. Thus, if there are K languages, N documents (each of which is translated into each of the K languages), and T distinct linguistic terms across all languages, then the term-by-document matrix is of dimensions T by N. Each cell in the matrix represents a weighted frequency of a particular term *t* (in any language) in a particular document *n*. The weighting scheme we use is a standard log-entropy scheme in which the weighted frequency $x_{t,n}$ of a particular term *t* in a particular document *n* is given by:

$$W = \log_2 (F + 1) \times (1 + H_t / \log_2 (N))$$

where F is the raw frequency of *t* in *n*, and H_t is a measure of the entropy of the term across all documents. The last term in the expression above, $\log_2$ (N), is the maximum entropy that any term can have in the corpus, and therefore $(1 + H_t / \log_2 (N))$ is 1 for the most distinctive terms in the corpus, 0 for those which are least distinctive. The log-entropy weighting scheme has been shown to outperform other schemes such as tf-idf in LSA-based retrieval (see for example Dumais 1991).

The sparse term-by-document matrix is subjected to singular value decomposition (SVD), and a reduced non-sparse matrix is output. Generally, we used the output corresponding to the top 300 singular values in our experiments.

To evaluate the similarity of unseen queries or documents (those not in the training set) to one another, these documents are tokenized, the weighted frequencies are calculated in the same way as they were for the training set, and the results are multiplied by the matrices output by the SVD to project the unseen queries/documents into a ‘semantic space’, assigning (in our case) 300-dimensional vectors to each document. Again, our approach to measuring the similarity of one document to another is a standard one: we calculate the cosine between the respective vectors.

For CLIR, the main advantages of an approach like LSA are that it is by now quite well-understood; the underlying algorithms remain constant regardless of which languages are being compared; and there is wide scope to use different sets of training data, providing they exist in parallel corpora. LSA is thus a highly generic approach to CLIR: since it relies only on the ability to tokenize text at the boundaries between words, or more generally semantic units, it can be generalized to virtually all languages.

2.2 Training and test data

For our experiments, the training and test data were taken from the Bible and Quran respectively. As training data, the Bible lends itself extremely well to multilingual LSA. It is highly available in multiple languages[1] (over 80 parallel translations in 50 languages, mostly public-domain, are available from a single website, www.unboundbible.org); and a very fine-grained alignment is possible (by verse) (Resnik et al 1999, Chew and Abdelali 2007). Many purpose-built parallel corpora are biased towards particular language groups (for example, the European Union funds work in CLIR, but it tends to be biased towards European languages – for example, see Peters 2001). This is not as true of the Bible, and the fact that it covers a wider range of languages is a reflection of the reasons it was translated in the first place.

The question which is most commonly raised about use of the Bible in this way is whether its coverage of vocabulary from other domains is sufficient to allow it to be used as training data for most applications. Based on a variety of experiments we have carried out (see for example Chew et al. forthcoming), we believe this need not always be a drawback – it depends largely on the intended application. However, it is beyond our scope to address this in detail here; it is sufficient to note that for the experiments we describe in this paper, we were able to achieve perfectly respectable CLIR results using the Bible as the training data.

[1] It has proved hard to come by reliable statistics to allow direct comparison, but the Bible is generally believed to be the world’s most widely translated book. At the end of 2006, it is estimated that there were full translations into 429 languages and partial translations into 2,426 languages (Bible Society 2007).

As test data, we used the 114 suras (chapters) of the Quran, which has also been translated into a wide variety of languages. Clearly, both training and test data have to be available in multiple languages to allow the effectiveness of CLIR to be measured in a meaningful way. For the experiments reported in this paper, we limited the testing languages to Arabic, English, French, Russian and Spanish (the respective abbreviations AR, EN, FR, RU and ES are used hereafter). The test data thus amounted to 570 (114 × 5) documents: a relatively small set, but large enough to achieve statistically significant results for our purposes, as will be shown. In all tests described in this paper, we use the same test set: thus, although the test documents all come from a single domain, it is reasonable to suppose that the comparative results can be generalized to other domains.

The complete list of languages used for both testing and training is given in Table 1.

Language	Bible -training-	Quran -test-	Language Family	Sub-Family
Afrikaans	Yes	No	Indo-European	Germanic-West
Amharic	Yes	No	Afro-Asiatic	Semitic-South
Arabic	Yes	Yes	Afro-Asiatic	Semitic-Central
Aramaic	Yes	No	Afro-Asiatic	Semitic-North
Czech	Yes	No	Indo-European	Slavic-West
Danish	Yes	No	Indo-European	Germanic-North
Dutch	Yes	No	Indo-European	Germanic-West
English	Yes	Yes	Indo-European	Germanic-West
French	Yes	Yes	Indo-European	Italic
Hebrew	Yes	No	Afro-Asiatic	Semitic-Central
Hungarian	Yes	No	Uralic	Finno-Ugric
Japanese	Yes	No	Altaic	
Latin	Yes	No	Indo-European	Italic
Persian	Yes	No	Indo-European	Indo-Iranian
Russian	Yes	Yes	Indo-European	Slavic-East
Spanish	Yes	Yes	Indo-European	Italic

Table 1. Languages used for training and testing

2.3 Test method

We tokenized each of the 570 test documents, applying the weighting scheme described above to obtain a vector of weighted frequencies of each term in the document, then multiplying that vector by U × S^{-1}, also as described above. The result was a set of projected document vectors in the 300-dimensional LSA space.

For some of our experiments, we used a light stemmer for Arabic (Darwish 2002) to replace inflected forms in the training data with citation forms. It is commonly accepted that morphology improves IR (Abdou et al. 2005, Lavie et al. 2004, Larkey et al. 2002, Oard and Gey 2002), and it will be seen that our results generally confirm this.

For Hebrew, we used the Westminster Leningrad Codex in the training data. Since this is available for download either with vowels or without vowels, no morphological pre-processing was required in this case; we simply substituted one version for the other in the training data when necessary.

Various measurements are used for evaluating IR systems performance (Van Rijsbergen 1979). However, since the aim of our experiments is to assess whether we could identify the correct translation for a given document among a set of possibilities in another language (i.e., given the language of the query and the language of the results), we selected ‘precision at 1 document’ as our preferred metric. This metric represents the proportion of cases, on average, where the translation was retrieved first.

3 Challenges of Semitic languages

The features which make Semitic languages challenging for information retrieval are generally fairly well understood: it is probably fair to say that chief among them is their complex morphology (for example, ambiguity resulting from diacritization, root-and-pattern alternations, and the use of infix morphemes as described in Habash 2004).

These challenges can be illustrated by means of a statistical comparison of a portion of our training data (the Gospel of Matthew) as shown in Table 2.

	Types	**Tokens**
Afrikaans	2,112	24,729
French	2,840	24,438
English	2,074	23,503
Dutch	2,613	23,099
Danish	2,649	21,816
Spanish	3,075	21,279
Persian	3,587	21,190
Hungarian	4,730	18,787
Czech	4,236	18,000
Russian	4,196	16,826
Latin	3,936	16,543
Hebrew (Modern)	4,337	14,153
Arabic	4,607	13,930
Japanese	5,741	13,130
Amharic	5,161	12,940
TOTAL	55,894	284,363

Table 2. Statistics of parallel texts by language

From Table 2, it should be clear that there is generally an inverse relationship between the number of types and tokens. Modern Indo-European (IE) (and particularly Germanic or Italic languages) are at one end of the spectrum, while the Semitic languages (along with Japanese) are at the other. The statistics separate 'analytic' languages from 'synthetic' ones, and essentially illustrate the fact that, thanks to the richness of their morphology, the Semitic languages pack more information (in the information-theoretic sense) into each term than the other languages. Because this results in higher average entropy per word (in the information theoretic sense), a challenge is presented to information retrieval techniques such as LSA which rely on tokenization at word boundaries: it is harder to isolate each 'unit' of meaning in a synthetic language. The actual effect this has on information retrieval precision will be shown in the next section.

4 Results with LSA

The series of experiments described in this section have the aims of:

- clarifying what effect morphological analysis of the training data has on CLIR precision;
- highlighting the effect on CLIR precision of adding more languages in training;
- illustrating what the impact is of adding a partial translation (text in one language which is only partially parallel with the texts in the other languages)

We choose Arabic as the language of focus in our experiment; specifically for these experiments, we intended to reveal the effect of adding languages from the same group (Semitic) compared with that of adding languages of different groups.

First, we present results in Table 3 which confirm that morphological analysis of the training data improves CLIR performance.

	ES	RU	FR	EN	AR
without morphological analysis of Arabic					
ES	1.0000	0.5614	0.8333	0.7368	0.2895
RU	0.4211	1.0000	0.5263	0.7632	0.2632
FR	0.7807	0.7018	1.0000	0.8158	0.4035
EN	0.7193	0.8158	0.8596	1.0000	0.4825
AR	0.5000	0.2807	0.6228	0.5526	1.0000
Average precision: Overall 0.677, within IE 0.783, IE-Semitic 0.488					
with morphological analysis of Arabic					
ES	1.0000	0.6579	0.8772	0.7807	0.4123
RU	0.4912	1.0000	0.7193	0.8158	0.3947
FR	0.8421	0.7719	1.0000	0.8421	0.3772
EN	0.8070	0.8684	0.8947	1.0000	0.3684
AR	0.3947	0.3509	0.5614	0.4561	1.0000
Average precision: Overall 0.707, within IE 0.836, IE-Semitic 0.480					

Table 3. Effect of morphological analysis[2]

An important point to note first is that CLIR precision is generally much lower for pairs including Arabic than it is elsewhere, lending support to our assertion above that Arabic and other Semitic languages present special challenges in information retrieval.

It also emerges from Table 3 that when morphological analysis of Arabic was added, the overall average precisions increased from 0.677 to 0.707, a highly significant increase ($p \approx 6.7 \times 10^{-8}$). (Here and below, a chi-squared test is used to measure statistical significance.)

Given that the ability of morphological analysis to improve IR precision has been documented, this result in itself is not surprising. However, it is interesting that the net benefit of adding morphological analysis – and just to Arabic within the training data – was more or less confined to pairs of non-Semitic languages. We believe that the explanation is that by adding morphology more relations (liai-

[2] In this and the following tables, the metric used is precision at 1 document (discussed in section 2.3).

sons) are defined in LSA between the words from different languages. For language pairs including Arabic, the average precision actually decreased from 0.488 to 0.480 when morphology was added (although this decrease is insignificant).

With the same five training languages as used in Table 3, we added Persian. The results are shown in Table 4.

	ES	RU	FR	EN	AR
ES	1.0000	0.6140	0.8246	0.7632	0.3246
RU	0.5088	1.0000	0.6667	0.7982	0.2281
FR	0.8772	0.7368	1.0000	0.8158	0.3947
EN	0.8246	0.8333	0.8947	1.0000	0.4035
AR	0.4474	0.4386	0.6140	0.5526	1.0000
Average precision: Overall 0.702, within IE 0.822, IE-Semitic 0.489					

Table 4. Effect on CLIR of adding Persian

First to note is that the addition of Persian (an IE language) led to a general increase in precision for pairs of IE languages (Spanish, Russian, French and English) from 0.783 to 0.822 but no significant change for pairs including Arabic (0.488 to 0.489). Although Persian and Arabic share the same script, these results confirm that genetic relatedness is a much more important factor in affecting precision.

Chew and Abdelali (2007) show that the results of multilingual LSA generally improve as the number of parallel translations used in training increases. Our next step here, therefore, is to analyze whether it makes any difference whether the additional languages are from the same or different language groups. In Table 5 we compare the results of adding an IE language (Latin), an Altaic language (Japanese), and another Semitic language (Hebrew) to the training data. In all three cases, no morphological analysis of the training data was performed.

Based on these results, cross-language precision yielded only very slightly improved results overall by adding Latin or Japanese. With Japanese, the net improvement (0.677 to 0.680) was not statistically significant overall, neither was the change significant for pairs either including or excluding Arabic (0.488 to 0.485 and 0.783 to 0.789 respectively). Note that this is even though Japanese shares some statistical (although of course not linguistic) properties with the Semitic languages, as shown in Table 2. With Latin, the net overall improvement (0.677 to 0.699) was barely significant ($p \approx 0.01$) and was insignificant for pairs including Arabic (0.488 to 0.496). With Hebrew, however, the net improvement was highly significant in all cases (0.677 to 0.718, $p \approx 3.36 \times 10^{-6}$ overall, 0.783 to 0.819, $p \approx 2.20 \times 10^{-4}$ for non-Semitic pairs, and 0.488 to 0.538, $p \approx 1.45 \times 10^{-3}$ for pairs including Arabic). We believe that these results indicate that there is more value overall in ensuring that languages are paired with at least one other related language in the training data; our least impressive results (with Japanese) were when two languages in training (one Semitic and one Altaic language) were 'isolated'.

	ES	RU	FR	EN	AR
Latin included in training data					
ES	1.0000	0.6140	0.8333	0.7456	0.2544
RU	0.4737	1.0000	0.6316	0.8246	0.3333
FR	0.8596	0.7368	1.0000	0.8333	0.4474
EN	0.7719	0.7982	0.8860	1.0000	0.4474
AR	0.5088	0.3509	0.6140	0.5088	1.0000
Average precision: Overall 0.699, within IE 0.813, IE-Semitic 0.496					
Japanese included in training data					
ES	1.0000	0.5789	0.8333	0.7456	0.2895
RU	0.4298	1.0000	0.5526	0.7807	0.2719
FR	0.7719	0.7368	1.0000	0.8070	0.4035
EN	0.7193	0.807	0.8596	1.0000	0.4123
AR	0.5088	0.2982	0.614	0.5702	1.0000
Average precision: Overall 0.680, within IE 0.789, IE-Semitic 0.485					
Modern Hebrew (no vowels) in training data					
ES	1.0000	0.6140	0.8596	0.7807	0.3509
RU	0.4561	1.0000	0.6667	0.7719	0.3684
FR	0.8509	0.7193	1.0000	0.8684	0.4298
EN	0.7632	0.8509	0.9035	1.0000	0.4298
AR	0.5263	0.4474	0.6491	0.6404	1.0000
Average precision: Overall 0.718, within IE 0.819, IE-Semitic 0.538					

Table 5. Effect of language relatedness on CLIR

The next set of results are for a repetition of the previous three experiments, but this time with morphological analysis of the Arabic data. These results are shown in Table 6.

As was the case without the additional languages, the overall effect of adding morphological analysis of Arabic is still to increase precision. In all three cases, the net improvement for pairs excluding Arabic is highly significant (0.813 to 0.844 with Latin, 0.789 to 0.852 with Japanese, and 0.819 to 0.850 with Hebrew). For pairs including Arabic, however, the change is again insignificant. This was a consistent but surprising feature of our results, that morphological analysis of Arabic in fact appears to benefit non-Semitic languages more than it benefits Arabic itself, at least with this dataset. The results might possibly have been different if we had included other Semitic languages in the test data, although this appears unlikely as we found the same phenomenon consistently occurring across a wide variety of tests, and regardless of which languages we used in training.

	ES	RU	FR	EN	AR
Latin included in training data					
ES	1.0000	0.6579	0.8684	0.7456	0.4211
RU	0.5614	1.0000	0.7456	0.8509	0.4386
FR	0.8421	0.8158	1.0000	0.8509	0.4211
EN	0.8421	0.8333	0.8947	1.0000	0.4123
AR	0.4123	0.3947	0.5351	0.4825	1.0000
Average precision: Overall 0.721, within IE 0.844, IE-Semitic 0.502					
Japanese included in training data					
ES	1.0000	0.7544	0.8684	0.8070	0.4211
RU	0.4737	1.0000	0.7193	0.8509	0.4123
FR	0.8246	0.8596	1.0000	0.8772	0.4211
EN	0.8421	0.8596	0.8947	1.0000	0.4035
AR	0.3333	0.3509	0.5614	0.4649	1.0000
Average precision: Overall 0.720, within IE 0.852, IE-Semitic 0.485					
Modern Hebrew (no vowels) in training data					
ES	1.0000	0.7018	0.9035	0.7982	0.4561
RU	0.5614	1.0000	0.7105	0.8070	0.4035
FR	0.8421	0.8246	1.0000	0.8596	0.4825
EN	0.8509	0.8509	0.8947	1.0000	0.4123
AR	0.3947	0.4298	0.5351	0.5175	1.0000
Average precision: Overall 0.729, within IE 0.850, IE-Semitic 0.514					

Table 6. Effect of language relatedness and morphology on CLIR

For further verification, we explored what would happen if only the Arabic root were included in morphological analysis. As already mentioned, for languages that combine affixes with the stem, there is a higher token-to-type ratio. Omitting the affix from the morphological analysis of these languages reveals the importance of considering the affixes and their contribution to the semantics of a given sentence. Although LSA is not sentence-structure-aware (as it uses a bag-of-words approach), the importance of considering the affixes as part of the sentence is very crucial. The results in Table 7 demonstrate clearly that ignoring or over-looking the word affixes has a negative effect on the overall performance of the CLIR system. When including only the Arabic stem, a performance degradation is noticeable across all languages, with a larger impact on IE languages. The results which illustrate can be seen by comparing Table 7 with Table 3.

	ES	RU	FR	EN	AR
morphological analysis of Arabic –Stem only-					
ES	1.0000	0.5789	0.8070	0.7807	0.3421
RU	0.4912	1.0000	0.6842	0.8246	0.1842
FR	0.8421	0.7018	1.0000	0.8333	0.4211
EN	0.8333	0.8333	0.9211	1.0000	0.4211
AR	0.4561	0.4386	0.5702	0.4912	1.0000
Average precision: Overall 0.698, within IE 0.821, IE-Semitic 0.481					

Table 7. Effect of Using Stem only

Next, we turn specifically to a comparison of the effect that different Semitic languages have on CLIR precision. Here, we compare the results when the sixth language used in training is Hebrew, Amharic, or Aramaic. However, since our Amharic and Aramaic training data were only partially parallel (we have only the New Testament in Amharic, and only portions of the New Testament in Aramaic), we first considered the effect that partial translations have on precision. Table 8 shows the results we obtained when only the Hebrew Old Testament (with vowels) was used as the sixth parallel version. No morphological analysis was performed.

	ES	RU	FR	EN	AR
without morphological analysis of Arabic					
ES	1.0000	0.6842	0.8421	0.8158	0.3947
RU	0.4211	1.0000	0.6228	0.7982	0.4737
FR	0.8509	0.7719	1.0000	0.8509	0.4737
EN	0.7895	0.8333	0.8684	1.0000	0.4649
AR	0.4561	0.3333	0.6404	0.4561	1.0000
Average precision: Overall 0.714, within IE 0.822, IE-Semitic 0.521					
with morphological analysis of Arabic					
ES	1.0000	0.7105	0.9035	0.8333	0.4737
RU	0.4649	1.0000	0.7456	0.8333	0.4912
FR	0.8421	0.8070	1.0000	0.8860	0.4474
EN	0.8772	0.8421	0.9298	1.0000	0.4298
AR	0.2719	0.3684	0.5088	0.5000	1.0000
Average precision: Overall 0.727, within IE 0.855, IE-Semitic 0.499					

Table 8. Effect of partial translation on CLIR

Although two or more parameters differ from those used for Hebrew in Table 5 (a fully-parallel text in modern Hebrew without vowels, versus a partial text in Ancient Hebrew with vowels), it is worth comparing the two sets of results. In particular, the reductions in average precision from 0.718 to 0.714 and from 0.729 to 0.727 respectively are insignificant. Likewise, the changes for pairs with and without Arabic were insignificant. This appears to show that, at least up to a certain point, even only partially parallel corpora can successfully be used under our LSA-based approach. We now turn to the results we obtained using Aramaic, with the intention of comparing these to our previous results with Hebrew.

	ES	RU	FR	EN	AR
no morphological analysis of Arabic					
ES	1.0000	0.4035	0.8070	0.7368	0.2632
RU	0.3509	1.0000	0.5965	0.6579	0.2281
FR	0.8421	0.6754	1.0000	0.8246	0.2719
EN	0.7018	0.6754	0.8947	1.0000	0.2719
AR	0.4825	0.2807	0.4649	0.3947	1.0000
Average precision: Overall 0.633, within IE 0.760, IE-Semitic 0.406					
morphological analysis of Arabic					
ES	1.0000	0.5351	0.8684	0.7719	0.2895
RU	0.5175	1.0000	0.6930	0.7807	0.3421
FR	0.8947	0.7807	1.0000	0.8684	0.2807
EN	0.8070	0.8158	0.9035	1.0000	0.2982
AR	0.3509	0.2193	0.3772	0.2895	1.0000
Average precision: Overall 0.667, within IE 0.827, IE-Semitic 0.383					

Table 9. Effect of Aramaic on CLIR

Here, there is a noticeable across-the-board decrease in precision from the previous results. We believe that this may have more to do with the fact that the Aramaic training data we have is fairly sparse (2,957 verses of the Bible out of a total of 31,226, compared with 23,269 out of 31,226 for Ancient Hebrew). It is likely that at some point as the parallel translation's coverage drops (somewhere between the coverage of the Hebrew and the Aramaic), there is a severe hit to the performance of CLIR. Accordingly, we discarded Aramaic for further tests.

Next, we considered the addition of two Semitic languages other than Arabic, Modern Hebrew and Amharic, to the training data. In this case, we performed morphological analysis of Arabic.

The results appear to show a significant increase in precision for pairs of IE languages and a significant *decrease* for cross-language-group cases (those where an IE language is paired with Arabic), compared to when just Modern Hebrew was used in the training data (see the relevant part of Table 6). It is not clear why this is the case, but in this case we believe that it is quite possible that the results would have been different if more than one Semitic language had been included in the test data.

	ES	RU	FR	EN	AR
ES	1.0000	0.6930	0.8860	0.7719	0.4649
RU	0.5000	1.0000	0.7456	0.8684	0.5175
FR	0.8772	0.7982	1.0000	0.8772	0.4649
EN	0.8684	0.8596	0.9298	1.0000	0.4386
AR	0.2632	0.2982	0.4386	0.3947	1.0000
Average precision: Overall 0.718, within IE 0.855, IE-Semitic 0.476					

Table 10. CLIR with 7 languages (including Modern Hebrew and Amharic)

We now come to a rare example where we achieved a boost in precision specifically for Arabic. In this case, we repeated the last experiment but removed the vowels from the Hebrew text. The results are shown in Table 11.

	ES	RU	FR	EN	AR
ES	1.0000	0.7018	0.8772	0.8158	0.5088
RU	0.5175	1.0000	0.7632	0.8421	0.4825
FR	0.8596	0.8246	1.0000	0.8860	0.5351
EN	0.8947	0.8158	0.9298	1.0000	0.5088
AR	0.2895	0.3772	0.5526	0.5000	1.0000
Average precision: Overall 0.739, within IE 0.858, IE-Semitic 0.528					

Table 11. Effect of removing Hebrew vowels

Average precision for pairs including Arabic increased from 0.476 to 0.528, an increase which was significant ($p \approx 7.33 \times 10^{-4}$), but for other pairs the change was insignificant. Since the Arabic text in training did not include vowels, we believe that the exclusion of vowels from Hebrew placed the two languages on a more common footing, allowing LSA, for example, to make associations between Hebrew and Arabic roots which otherwise might not have been made. Although Hebrew and Arabic do not always share common stems, it can be seen from Table 2 that the type/token statistics of Hebrew (without vowels) and Arabic are very similar. The inclusion of Hebrew vowels would change the statistics for Hebrew considerably, increasing the number of types (since previously indistinguishable wordforms would now be listed separately). Thus, with the *exclusion* of Hebrew vowels, there should be more instances where Arabic tokens can be paired one-to-one with Hebrew tokens.

Finally, in order to confirm our conclusions and to eliminate any doubts about the results obtained so far, we experimented with more languages. We added Japanese, Afrikaans, Czech, Danish, Dutch, Hungarian and Hebrew in addition to our 5 original languages. Morphological analysis of the Arabic text in training was performed, as in some of the previous experiments. The results of these tests are shown in Table 12.

	ES	RU	FR	EN	AR
11 languages (original 5 + Japanese, Afrikaans, Czech, Danish, Dutch, and Hungarian)					
ES	1.0000	0.6754	0.9035	0.7719	0.5526
RU	0.4737	1.0000	0.7632	0.8772	0.5175
FR	0.8596	0.8070	1.0000	0.8947	0.5088
EN	0.8421	0.8684	0.9035	1.0000	0.4912
AR	0.3772	0.2632	0.6316	0.4912	1.0000
Average precision: Overall 0.739, within IE 0.853, IE-Semitic 0.537					
12 languages (as above plus Hebrew)					
ES	1.0000	0.7018	0.8947	0.7719	0.6404
RU	0.6667	1.0000	0.7105	0.9123	0.6228
FR	0.8772	0.8333	1.0000	0.8421	0.6404
EN	0.6667	0.8684	0.9035	1.0000	0.6316
AR	0.5877	0.4386	0.5965	0.6491	1.0000
Average precision: Overall 0.778, within IE 0.853, IE-Semitic 0.645					

Table 12. Effect of further languages on CLIR

Generally, these results confirm the finding of Chew and Abdelali (2007) about adding more languages; doing so enhances the ability to identify translations across language boundaries. Across the board (for Arabic and other languages), the increase in precision gained by adding Afrikaans, Czech, Danish, Dutch and Hungarian is highly significant (compared to the part of Table 5 which deals with Japanese, overall average precision increased from 0.680 to 0.739, with $p \approx 1.17 \times 10^{-11}$; for cross-language-group retrieval, from 0.485 to 0.537, with $p \approx 9.31 \times 10^{-4}$; for pairs within IE, from 0.789 to 0.853 with $p \approx 2.81 \times 10^{-11}$). In contrast with most previous results, however, with the further addition of Hebrew, precision was boosted primarily for Arabic (0.537 to 0.645 with $p \approx 4.39 \times 10^{-13}$). From this and previous results, it appears that there is no clear pattern to when the addition of a Semitic language in training was beneficial to the Semitic language in testing.

5 Conclusion and future work

Based on our results, it appears that although clear genetic relationships exist between certain languages in our training data, it was less possible than we had anticipated to leverage this to our advantage. We had expected, for example, that by including multiple Semitic languages in the training data within an LSA framework, we would have been able to improve cross-language information retrieval results specifically for Arabic. Perhaps surprisingly, the greatest benefit of including additional Semitic languages in the training data is most consistently to non-Semitic languages. A clear observation is that *any* additional languages in training are generally beneficial, and the benefit of additional languages can be considerably greater than the benefits of linguistic pre-processing (such as morphological analysis). Secondly, it is not necessarily the case that cross-language retrieval with Arabic is helped most by including other Semitic languages, despite the genetic relationship. Finally, as we expected, we were able to rule out script similarity (e.g. between Persian and Arabic) as a factor which might improve precision. Our results appear to demonstrate clearly that language relatedness is much more important in the training data than use of the same script.

Finally, to improve cross-language retrieval with Arabic – the most difficult case in the languages we tested – we attempted to ‘prime’ the training data by including Arabic morphological analysis. This did lead to a statistically significant improvement overall in CLIR, but – perhaps paradoxically – the improvement specifically for cross-language retrieval with Arabic was negligible in most cases. The only two measures which were successful in boosting precision for Arabic significantly were (1) the inclusion of Modern Hebrew in the training data; and (2) the elimination of vowels in the Ancient Hebrew training data – both measures which would have placed the training data for the two Semitic languages (Arabic and Hebrew) on a more common statistical footing. These results appear to confirm our hypothesis that there is value, within the current framework, of ‘pairing’ genetically related languages in the training data. In short, language relatedness does matter in cross-language information retrieval.

6 Acknowledgement

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy’s National Nuclear Security Administration under contract DE-AC04-94AL85000.

7 References

Abdou, S., Ruck, P., and Savoy, J. 2005. Evaluation of Stemming, Query Expansion and Manual Indexing Approaches for the Genomic Task. In *Proceedings of TREC 2005*.

Berry, M. W., Dumais, S. T., and O'Brien, G. W. 1994. Using Linear Algebra for Intelligent Information Retrieval. *SIAM: Review*, 37, 573-595.

Biola University. 2005-2006. *The Unbound Bible*. Accessed at http://www.unboundbible.com/ on February 27, 2007.

Chew, P. A., and Abdelali, A. 2007. *Benefits of the 'Massively Parallel Rosetta Stone': Cross-Language Information Retrieval with over 30 Languages*, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, ACL 2007. Prague, Czech Republic, June 23–30, 2007. pp. 872-879.

Chew, P. A., Kegelmeyer, W. P., Bader, B. W. and Abdelali, A. Forthcoming. *The Knowledge of Good and Evil: Multilingual Ideology Classification with PARAFAC2 and Maching Learning*.

Chew, P. A., Verzi, S. J., Bauer, T. L., and McClain, J. T. 2006. Evaluation of the Bible as a Resource for Cross-Language Information Retrieval. *Proceedings of the Workshop on Multilingual Language Resources and Interoperability*, 68–74.

Darwish, K. 2002. *Building a shallow Arabic morphological analyzer in one day*. In Proceedings of the Association for Computational Linguistics (ACL-02), 40th Anniversary Meeting. pp. 47-54.

Dumais, S. T. 1991. Improving the Retrieval of Information from External Sources. *Behavior Research Methods, Instruments, and Computers* 23 (2), 229-236.

Dumais, S. T., Furnas, G. W., Landauer, T. K., Deerwester, S. and Harshman, R. 1998. Using Latent Semantic Analysis to Improve Access to Textual Information. In *CHI'88: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 281-285. ACM Press.

Frakes, W. B. and Baeza-Yates, R. 1992. *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall: New Jersey.

Habash, N. 2004. Large Scale Lexeme Based Arabic Morphological Generation. In *Proc. of Traitement Automatique du Langage Naturel.*

Larkey, L., Ballesteros, L. and Connell, M. 2002. Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-Occurrence Analysis. *SIGIR 2002*, Finland, pp. 275-282.

Larkey, L. and Connell, M. 2002. Arabic Information Retrieval at Umass in TREC-10. In Voorhees, E.M. and Harman, D.K. (eds.): *The Tenth Text Retrieval Conference, TREC 2001 NIST Special Publication 500-250*, pp. 562-570.

Lavie, A., Peterson, E., Probst, K., Wintner, S., and Eytani, Y. 2004. Rapid Prototyping of a Transfer-Based Hebrew-to-English Machine Translation System. In *Proceedings of the TMI-04*.

Mathieu, B., Besançon, R. and Fluhr, C. 2004. Multilingual Document Clusters Discovery. *Recherche d'Information Assistée par Ordinateur (RIAO) Proceedings*, 1-10.

Oard, D. and Gey, F. 2002. *The TREC 2002 Arabic/English CLIR Track, NIST TREC 2002 Proceedings*, pp. 16-26.

Peters, C. (ed.). 2001. *Cross-Language Information Retrieval and Evaluation: Workshop of the Cross-Language Evaluation Forum, CLEF 2000*. Berlin: Springer-Verlag.

Resnik, P., Olsen, M. B., and Diab, M. 1999. The Bible as a Parallel Corpus: Annotating the "Book of 2000 Tongues". *Computers and the Humanities*, 33, 129-153.

Van Rijsbergen, C. 1979. *Information Retrieval (2nd edition)*. Butterworth: London.

Identifying Similar and Co-referring Documents Across Languages

Pattabhi R K Rao T
AU-KBC Research Centre,
MIT Campus, Anna University,
Chennai-44, India.
pattabhi@au-kbc.org

Sobha L
AU-KBC Research Centre,
MIT Campus, Anna University,
Chennai-44, India.
sobha@au-kbc.org

Abstract

This paper presents a methodology for finding similarity and co-reference of documents across languages. The similarity between the documents is identified according to the content of the whole document and co-referencing of documents is found by taking the named entities present in the document. Here we use Vector Space Model (VSM) for identifying both similarity and co-reference. This can be applied in cross-lingual search engines where users get documents of very similar content from different language documents.

1 Introduction

In this age of information technology revolution, the growth of technology and easy accessibility has contributed to the explosion of text data on the web in different media forms such as online news magazines, portals, emails, blogs etc in different languages. This represents 80% of the unstructured text content available on the web. There is an urgent need to process such huge amount of text using Natural Language Processing (NLP) techniques. One of the significant challenges with the explosion of text data is to organize the documents into meaningful groups according to their content.

The work presented in this paper has two parts a) finding multilingual cross-document similarity and b) multilingual cross-document entity co-referencing. The present work analyzes the documents and identifies whether the documents are similar and co-referring. Two objects are said to be similar, when they have some common properties between them. For example, two geometrical figures are said to be similar if they have the same shape. Hence similarity is a measure of degree of resemblance between two objects.

Two documents are said to be similar if their contents are same. For example a document D1 describes about a bomb blast incident in a city and document D2 also describes about the same bomb blast incident, its cause and investigation details, then D1 and D2 are said to be similar. But if document D3 talks of terrorism in general and explains bomb blast as one of the actions in terrorism and not a particular incident which D1 describes, then documents D1 and D3 are dissimilar. The task of finding document similarity differs from the task of document clustering. Clustering is a task of categorization of documents based on domain/field. In the above example, documents D1, D2, D3 can be said to be in a cluster of crime domain. When documents are similar they share common noun phrases, verb phrases and named entities. While in document clustering, sharing of named entities and noun phrases is not essential but still there can be some noun phrases and named entities in common. Cross-document co-referencing of entities refers to the identification of same entities across the documents. When the named entities present in the documents which are similar and also co-referencing, then the documents are said to be co-referring documents.

The paper is further organized as follows. In section 2, the motivation behind this paper is explained and in 3 the methodology used is described. Results and discussions are dealt in section 4 and conclusion in section 5.

2 Motivation

Dekang Lin (1998) defines similarity from the information theoretic perspective and is applicable if the domain has probabilistic model. In the past decade there has been significant amount of work done on finding similarity of documents and organizing the documents according to their content. Similarity of documents are identified using different methods such as Self-Organizing Maps (SOMs) (Kohonen et al, 2000; Rauber, 1999), based on Ontologies and taxanomy (Gruber, 1993; Resnik, 1995), Vector Space Model (VSM) with similarity measures like Dice similarity, Jaccard's similarity, cosine similarity (Salton, 1989). Bagga (Bagga et al., 1998) have used VSM in their work for finding co-references across the documents for English documents. Chung and Allan (2004) have worked on cross-document co-referencing using large scale corpus, where they have said ambiguous names from the same domain (here for example, politics) are harder to disambiguate when compared to names from different domains. In their work Chung and Allan compare the effectiveness of different statistical methods in cross-document co-reference resolution task. Harabagiu and Maiorano (2000) have worked on multilingual co-reference resolution on English and Romanian language texts. In their system, "SWIZZLE" they use a data-driven methodology which uses aligned bilingual corpora, linguistic rules and heuristics of English and Romanian documents to find co-references. In the Indian context, obtaining aligned bilingual corpora is difficult. Document similarity between Indian languages and English is tough since the sentence structure differs and Indian languages are agglutinative in nature. In the recent years there has been some work done in the Indian languages, (Pattabhi et al, 2007) have used VSM for multilingual cross-document co-referencing, for English and Tamil, where no bilingual aligned corpora is used.

One of the methods used in cross-lingual information retrieval (CLIR) is Latent Semantic Analysis (LSA) in conjunction with multilingual parallel aligned corpus. This approach works well for information retrieval task where it has to retrieve most similar document in one language to a query given in another language. One of the drawbacks of using LSA in multilingual space for the tasks of document clustering, document similarity is that it gives similar documents more based on the language than by topic of the documents in different languages (Chew et al, 2007). Another drawback of LSA is that the reduced dimension matrix is difficult to interpret semantically. The examples in Table 1, illustrate this.

	Before Reduction	After Reduction
1.	{(car),(truck),(flower)}	{(1.2810*car+0.5685*truck),(flower)
2	{(car),(bottle),(flower)}	{(1.2810*car+0.5685*bottle),(flower)

Table 1. LSA Example

In the first example the component *(1.2810*car+0.5685*truck)* can be inferred as "Vehicle" but in cases such as in second example, the component *(1.2810*car+0.5685*bottle)* does not have any interpretable meaning in natural language. In LSA the dimension reduction factor '*k*' has very important role to play and the value of 'k' can be found by doing several experiments. The process of doing dimension reduction in LSA is computationally expensive. When LSA is used, it reduces the dimensions statistically and when there is no parallel aligned corpus, this can not be interpreted semantically.

Hence, in the present work, we propose VSM which is computationally simple, along with cosine similarity measure to find document similarity as well as entity co-referencing. We have taken English and three Dravidian languages viz. Tamil, Telugu and Malayalam for analysis.

3 Methodology

In VSM, each document is represented by a vector which specifies how many times each term occurs in the document (the term frequencies). These counts are weighted to reflect the importance of each term and weighting is the inverse document frequency (idf). If a term t occurs in n documents in the collection then the "*idf*" is the inverse of log n. This vector of weighted counts is called a "bag of words" representation. Words such as "stop words" (or function words) are not included in the representation.

The documents are first pre-processed, to get syntactic and semantic information for each word in the documents. The preprocessing of documents involves sentence splitting, morph analysis, part-of-speech (POS) tagging, text chunking and named entity tagging. The documents in English are preprocessed using Brill’s Tagger (Brill, 1994) for POS tagging and fn-TBL (Ngai and Florian, 2001) for text chunking. The documents in Indian languages are preprocessed, using a generic engine (Arulmozhi et al., 2006) for POS tagging, and text chunking based on TBL (Sobha and Vijay, 2006). For both English and Indian language documents the named entity tagging is done using Named Entity Recognizer (NER) which was developed based on conditional random field (CRF). The tagset used by the NER tagger is a hierarchical tagset, consists of mainly i) ENAMEX, ii) NUMEX and iii) TIMEX. Inside the ENAMEX there are mainly 11 subtype’s viz. a) Person b) Organization c) Location d) Facilities e) Locomotives f) Artifacts g) Entertainment h) Cuisines i) Organisms j) Plants k) Disease. For the task of multilingual cross-document entities co-referencing, the documents are further processed for anaphora resolution where the corresponding antecedents for each anaphor are tagged in the document. For documents in English and Tamil, anaphora resolution is done using anaphora resolution system. For documents in Malayalam and Telugu anaphora resolution is done manually. After the preprocessing of documents, the language model is built by computing the term frequency – inverse document frequency (tf-idf) matrix. For the task of finding multilingual cross-document similarity, we have performed four different experiments. They are explained below:

E1: The terms are taken from documents after removing the stop words. These are raw terms where no preprocessing of documents is done; the terms are unique words in the document collection.

E2: The terms taken are the words inside the noun phrases, verb phrases and NER expressions after removing the stop words.

E3: The whole noun phrase/verb phrase/NER expression is taken to be a single term.

E4: The noun phrase/NER expression along with the POS tag information is taken as a single term.

The first experiment is the standard VSM implementation. The rest three experiments differ in the way the terms are taken for building the VSM. For building the VSM model which is common for all language document texts, it is essential that there should be translation/transliteration tool. First the terms are collected from individual language documents and a unique list is formed. After that, using the translation/transliteration tool the equivalent terms in language L2 for language L1 are found. The translation is done using a bilingual dictionary for the terms present in the dictionary. For most of the NERs only transliteration is possible since those are not present in the dictionary. The transliteration tool is developed based on the phoneme match it is a rule based one. All the Indian language documents are represented in roman notation (wx-notation) for the purpose of processing.

After obtaining equivalent terms in all languages, the VSM model is built. Let S1 and S2 be the term vectors representing the documents D1 and D2, then their similarity is given by equation (1) as shown below.

$$\mathrm{Sim}(S1,S2) = \sum_{tj} (W_{1j} \text{ x } W_{2j}) \qquad \text{-- (1)}$$

Where,

tj is a term present in both vectors S1and S2.
W_{1j} is the weight of term tj in S1 and
W_{2j} is the weight of term tj in S2.

The weight of term tj in the vector S1 is calculated by the formula given by equation (2), below.

$$Wij=(tf*\log(N/df))/[\mathrm{sqrt}(S_{i1}^2+S_{i2}^2+\ldots\ldots+S_{in}^2)] \text{ --(2)}$$

Where,

tf = term frequency of term t_j
N=total number of documents in the collection
df = number of documents in the collection that the term t_j occurs in.
sqrt represents square root

The denominator $[\mathrm{sqrt}(S_{i1}^2+S_{i2}^2+\ldots\ldots+S_{in}^2)]$ is the cosine normalization factor. This cosine normalization factor is the Euclidean length of the vector $S_{i,}$ where ‘i’ is the document number in the collection and S_{in}^2 is the square of the product of (tf*log(N/df)) for term t_n in the vector S_i.

For the task of multilingual cross-document entity co-referencing, the words with-in the anaphor tagged sentences are considered as terms for building the language model.

4 Results and Discussion

The corpus used for experiments is collected from online news magazines and online news portals. The sources in English include “The Hindu”, “Times of India”, “Yahoo News”, “New York Times”, “Bangkok Post”, “CNN”, “WISC”, “The

Independent”. The sources for Tamil include “Dinamani”, “Dinathanthi”, “Dinamalar”, “Dinakaran”, and “Yahoo Tamil”. The work was primarily done using English and Tamil. Later on this was extended for Malayalam and Telugu. The data sources for Malayalam are “Malayala Manorama”, “Mathrubhumi”, “Deshabhimani”, “Deepika” and sources for Telugu include “Eenadu”, “Yahoo Telugu” and “Andhraprabha”. First we discuss about English and Tamil and Later Telugu and Malayalam.

The domains of the news taken include sports, business, politics, tourism etc. The news articles were collected using a crawler, and hence we find in the collection, a few identical news articles because they appear in different sections of the news magazine like in Front page section, in state section and national section.

The dataset totally consists of 1054 English news articles, 390 Tamil news articles. Here we discuss results in two parts; in the first part results pertaining to document similarity are explained. In second part we discuss results on multilingual cross-document entity co-referencing.

4.1 Document Similarity

The data collection was done in four instances, spread in a period of two months. At the first instance two days news was crawled from different news sources in English as well as Tamil. In the first set 1004 English documents and 297 Tamil documents were collected.

In this set when manually observed (human judgment) it was found that there are 90 similar documents forming 31 groups, rest of the documents were not similar. This is taken as gold standard for the evaluation of the system output.

As explained in the previous section, on this set the four experiments were performed. In the first experiment (E1), no preprocessing of the documents was done except that the stop words were removed and the language model was built. In this it was observed that the number of similar documents is 175 forming 25 groups. Here it was observed that along with actual similar documents, system also gives other not similar documents (according to gold standard) as similar ones. This is due to the fact there is no linguistic information given to the system, hence having words alone does not tell the context, or in which sense it is used. And apart from that named entities when split don’t give exact meaning, for example in name of hotels “Leela Palace” and “Mysore Palace”, if split into words yields three words, “Leela”, “Mysore”, and “Palace”. In a particular document, an event at hotel Leela Palace is described and the hotel is referred as Leela Palace or by Palace alone. Another document describes about Dussera festival at Mysore Palace. Now here the system identifies both these documents to be similar even though both discuss about different events. The precision of the system was observed to be 51.4%, where as the recall is 100% since all the documents which were similar in the gold standard is identified. Here while calculating the precision; we are considering the number of documents that are given by the system as similar to the number of documents similar according to the gold standard.

Hence to overcome the above discussed problem, we did the second experiment (E2) where only words which occur inside the noun phrases, verb phrases and named entities are considered as terms for building the language model. Here it is observed that the number of similar documents is 140 forming 30 groups. This gives a precision of 64.2% and 100% recall. Even though we find a significant increase in the precision but still there are large number of false positives given by the system. A document consists of noun phrases and verb phrases, when the individual tokens inside these phrases are taken; it is equivalent to taking almost the whole document. This reduces the noise. The problem of “Leela Palace” and “Mysore Palace” as explained in the previous paragraph still persists here.

In the third experiment (E3) the whole noun phrase, verb phrase and named entity is considered as a single term for building the language model. Here the phrases are not split into individual tokens; the whole phrase is a single term for language model. This significantly reduces the number of false positives given by the system. The system identifies 106 documents as similar documents forming 30 groups. Now the precision of the system is 84.9%. In this experiment, the problem of “Leela Palace” and “Mysore Palace” is solved. Though this problem was solved the precision of the system is low, hence we performed the fourth (E4) experiment.

In the fourth experiment (E4), the part-of-speech (POS) information is given along with the phrase for building the language model. It is observed that the precision of the system increases. The number of similar documents identified is 100 forming 31 groups. This gives a precision of 90% and a recall of 100%.

Another important factor which plays a crucial role in implementation of language model or VSM is the threshold point. What is the threshold point that is to be taken? For obtaining an answer for this question, few experiments were performed by setting the threshold at various points in the range 0.75 to 0.95. When the threshold was set at 0.75 the number of similar documents identified by the system was larger, not true positives but instead false positives. Hence the recall was high and precision was low at 50%. When the threshold was moved up and set at 0.81, the number of similar documents identified was more accurate and the number of false positives got reduced. The precision was found to be 66%. When the threshold was moved up still further and set at 0.90, it was found that the system identified similar documents which were matching with the human judgment. The precision of the system was found to be 90%. The threshold was moved up further to 0.95, thinking that the precision would further improve, but this resulted in documents which were actually similar to be filtered out by the system. Hence the threshold chosen was 0.9, since the results obtained at this threshold point had matched the human judgment. For the experiments E1, E2, E3 and E4 explained above, the threshold is fixed at 0.9.

A new set of data consisting of 25 documents from 5 days news articles is collected. This is completely taken from single domain, terrorism. These news articles describe specifically the Hyderabad bomb blast, which occurred on August 25th 2007. All these 25 documents were only English documents from various news magazines. This data set was collected specifically to observe the performance of the system, when the documents belonging to single domain are given. In the new data set, from terrorism domain, human judgment for document similarity was found to have 13 similar documents forming 3 groups. While using this data set the noun phrases, verb phrases and named entities along with POS information were taken as terms to build the language model and the threshold was set at 0.9, it was observed that the system finds 14 documents to be similar forming 3 groups. Here, out of 14 similar documents, only 12 documents match with the human judgment and one document which ought to be identified was not identified by the system. The document which was not identified described about the current event, that is, bomb blast on 25th August in the first paragraph and then the rest of the document described about the similar events that occurred in the past. Hence the similarity score obtained for this document with respect to other documents in the group was 0.84 which is lower than the threshold fixed. Hence the recall of the system is 92.3% and the precision of the system is 85.7%.

Another data set consisting of 114 documents was taken from tourism domain. The documents were both in Tamil and English, 79 documents in Tamil and 35 documents in English. This data set describes various pilgrim places and temples in Southern India. The human annotators have found 21 similar documents which form a group of three. These similar documents describe about Lord Siva's and Lord Murugan's temples. The system obtained 25 documents as similar and grouped into three groups. Out of 25 documents obtained as similar, four were dissimilar. These dissimilar documents described non-Siva temples in the same place. In these dissimilar documents the names of offerings, festivals performed were referred by the same names as in the rest of the documents of the group, hence these documents obtained similarity score of 0.96 with respect to other documents in the group. Here we get a precision of 84% and a recall of 100%.

A new data set consisting of 46 documents was taken from various news magazines. This set consists of 24 English documents, 11 Tamil documents, 7 Malayalam documents and 4 Telugu documents. This data set describes the earthquake in Indonesia on 12th September 2007 and tsunami warning in other countries. The news articles were collected on two days 13th and 14th September 2007.

The documents collected were in different font encoding schemes. Hence before doing natural language processing such as morph-analysis, POS tagging etc, the documents were converted to a common roman notation (wx-notation) using the font converter for each encoding scheme.

Here we have used multilingual dictionaries of place; person names etc for translation. The language model is built by taking noun phrases and verb phrases along with POS information were as terms. In this set human annotators have found 45 documents to be similar and have grouped them into one group. The document which was identified as dissimilar describes about a Tamil film shooting at Indonesia being done during the quake time. The system had identified all the 46 documents including the film shooting document in the collection to be similar and put into one group. The "*film shooting*" document consisted of two paragraphs about the quake incident, other two paragraphs consisted of statement by the film producer stating that the whole crew is safe and the shooting is temporarily suspended for next few days. Since this document also contained the content describing the earthquake found in other documents of the group, the system identified this *"film shooting"* document to be similar. Here one interesting point which was found was that all the documents gave a very high similarity score greater than 0.95. Hence the precision of the system is 97.8% and recall 100%.

The summary of all these experiments with different dataset is shown in the table 2 below.

SNo	Dataset	Precision %	Recall %
1	English 1004 and Tamil 297 documents	90.0	100.0
2	English 25 – terrorism domain documents	85.7	92.3
3	35 English Docs and Tamil 79 docs - Tourism domain	84.0	100.0
4	46 Docs on Earth Quake incident – 24 English, 11 Tamil, 7 Malayalam, 4 Telugu	97.8	100.0
Average		89.3 %	98.07%

Table 2. Summary of Results for Document similarity for four different data sets

4.2 Document Co-referencing

The documents that were identified as similar ones are taken for entity co-referencing. In this work the identification of co-referencing documents is done for English and Tamil. In this section first we discuss the co-referencing task for English documents in terrorism domain, then for documents in English and Tamil in Tourism domain. In the end of this section we discuss about documents in English and Tamil, which are not domain specific.

In the first experiment, the document collection in terrorism domain is taken for co-referencing task. This data set of 25 documents in terrorism domain consists of 60 unique person names. In this work we consider only person names for entity co-referencing. In this data set, 14 documents are identified as similar ones by the system. These 14 documents consist of 26 unique person names. .

The language model is built using only named entity terms and the noun, verb phrases occurring in the same sentence where the named entity occurs. POS information is also provided with the terms. Here we find that out of 26 entities, the system co-references correctly for 24 entities, even though the last names are same. The results obtained for these named entities is shown in the below table Table 3.

Entity Name	**No. of links containing the entity**	**Correct Responses obtained**	**Total Responses obtained**	**Precision %**	**Recall %**
Y S Rajasekhar Reddy	7	7	7	100	100
Indrasena Reddy	1	1	1	100	100
K Jana Reddy	1	1	1	100	100
Shivaraj Patil	2	2	2	100	100
Manmohan Singh	4	4	4	100	100
Abdul Shahel Mohammad	1	1	2	50	100
Mohammad Abdullah	1	1	2	50	100
Mohammad Amjad	1	1	1	100	100
Mohammad Yunus	1	1	1	100	100
Ibrahim	1	1	1	100	100
Dawood Ibrahim	1	1	1	100	100
Madhukar Gupta	3	3	3	100	100
N Chandrababu Naidu	2	2	2	100	100
Tasnim Aslam	2	2	2	100	100
Mahender Agrawal	1	1	1	100	100
Somnath Chatterjee	2	2	2	100	100
Pervez Musharaff	2	2	2	100	100
Sonia Gandhi	2	2	2	100	100
Taslima	1	1	1	100	100

Nasrin					
Bandaru Dattatreya	1	1	1	100	100
L K Advani	2	2	2	100	100
Average				95.2	100

Table 3. Results for entity co-referencing for English documents in terrorism domain

The system identifies the entity names ending with "Reddy" correctly. These names in the documents occur along with definite descriptions which helps the system in disambiguating these names. For example "*Y S Rajasekhar Reddy*" in most cases is referred to as "*Dr. Reddy*" along with the definite description "*chief minister*". Similarly the other name "*K Jana Reddy*" occurs with the definite description "*Home minister*". Since here we are taking full noun phrases as terms for building language model, this helps obtaining good results. For entities such as "Abdul Shahel Mohammad" and "Mohammad Abdullah", it is observed that the both names are referred in the documents as "Mohammad" and surrounding phrases do not have any distinguishing phrases such as definite descriptions, which differentiate these names. Both these entities have been involved in masterminding of the Hyderabad bomb blast. Hence the system couldn't disambiguate between these two named entities and identifies both to be same, hence it fails here.

In the second experiment, the data set in Tourism domain consisting of 79 Tamil Documents and 35 English documents is taken for the task of co-referencing. In this data set 25 documents were identified as similar. Now these similar documents of 25 are considered for entity co-referencing task. There are 35 unique names of Gods. Here in this domain, one of the interesting points is that, there are different names to refer to a single God. For example Lord Murugan, is also referred by other names such as "Subramanyan", "Saravana", "Karttikeyan", "Arumukan" etc. Simialrly for Lord Siva is referred by "Parangirinathar", "Dharbaraneswara" etc. It is observed that in certain documents the alias names are not mentioned along with common names. In these instances even human annotators found it tough for co-referencing, hence the system could not identify the co-references. This problem of alias names can be solved by having a thesaurus and using it for disambiguation. The results obtained for these named entities are shown in the table 4, below.

Entity Name	**No. of links containing the entity**	**Correct Responses obtained**	**Total Responses obtained**	**Precision %**	**Recall %**
Murugan	7	7	8	87.5	100
Shiva	10	9	9	100	90
Parvathi	10	9	11	81.8	90
Nala	5	5	5	100	100
Damayanthi	2	2	2	100	100
Narada	3	3	3	100	100
Saneeswarar	6	6	7	85.7	100
Deivayani	4	4	4	100	100
Vishnu	2	2	2	100	100
Vinayaka	3	3	3	100	100
Indra	2	2	2	100	100
Thirunavukkarasar	1	1	1	100	100
Mayan	2	2	2	100	100
Average				96.5	98.4

Table 4. Results for entity co-referencing for English and Tamil Documents in Tourism domain

The co-referencing system could disambiguate a document which was identified as similar by the system and dissimilar by the human annotator.

Another experiment is performed where both English and Tamil Documents are taken for entity co-referencing. In this experiment we have taken the data set in which there are 1004 English documents and 297 Tamil documents. The documents are not domain specific. Here 100 documents are identified as similar ones, which contains of 64 English and 36 Tamil documents. Now we consider these 100 similar documents for entity co-referencing. In the 100 similar documents, there are 520 unique named entities. The table (Table 5) below shows results of few interesting named entities in this set of 100 similar documents.

Entity Name	**No. of links containing the entity**	**Correct Responses obtained**	**Total Responses obtained**	**Precision %**	**Recall %**
Karunanidhi	7	7	7	100	100
Manmohan Singh	15	14	16	87.5	93.3
Sonia Gandhi	54	54	58	93.1	100
Shivaraj Patil	8	8	10	80	100
Prathibha Patil	24	24	26	92.3	100
Lalu Prasad	5	5	5	100	100

Atal Bihari Vajpayee	4	4	4	100	100
Abdul Kalam	22	22	22	100	100
Sania Mirza	10	10	10	100	100
Advani	8	8	8	100	100
Average				95.3	99.3

Table 5. Results for entity co-referencing for English and Tamil Documents not of any specific domain

5 Conclusion

The VSM method is a well known statistical method, but here it has been applied for multilingual cross-document similarity, which is a first of its kind. Here we have tried different experiments and found that using phrases with its POS information as terms for building language model is giving good performance. In this we have got an average precision of 89.3 and recall of 98.07% for document similarity. Here we have also worked on multilingual cross-document entity co-referencing and obtained an average precision of 95.6 % and recall of 99.2 %. The documents taken for multilingual cross-document co-referencing are similar documents identified by the similarity system. Considering similar documents, helps indirectly in getting contextual information for co-referencing entities, because obtaining similar documents removes documents which are not in the same context. Hence this helps in getting good precision. Here we have worked on four languages viz. English, Tamil, Malayalam and Telugu. This can be applied for other languages too. Multilingual document similarity and co-referencing, helps in retrieving similar documents across languages.

References

Arulmozhi Palanisamy and Sobha Lalitha Devi. 2006. *HMM based POS Tagger for a Relatively Free Word Order Language*, Journal of Research on Computing Science, Mexico. 18:37-48.

Bagga, Amit and Breck Baldwin. 1998. *Entity-Based Cross-Document Coreferencing Using the Vector Space Model*, Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-ACL'98):79-85.

Brill, Eric. 1994. *Some Advances in transformation Based Part of Speech Tagging,* Proceedings of the Twelfth International Conference on Artificial Intelligence (AAAI-94), Seattle, WA

Peter A. Chew, Brett W. Bader, Tamara G. Kolda, Ahmed Abdelali. 2007. *Cross-Language Information Retrieval Using PARAFAC2,* In the Proceedings Thirteenth International Conference on Knowledge Discovery and Data Mining (KDD' 07), San Jose, California.:143-152.

Chung Heong Gooi and James Allan. 2004. *Cross-Document Coreference on a Large Scale Corpus,* Proceedings of HLT-NAACL: 9-16.

Dekang Lin. 1998. *An Information-Theoretic Definition of Similarity,* Proceedings of International Conference on Machine Learning, Madison, Wisconsin, July.

T. R. Gruber. 1993. *A translation approach to portable ontologies,* Knowledge Acquisition, 5(2):199–220.

Harabagiu M Sanda and Steven J Maiorano. 2000. *Multilingual Coreference Resolution,* Proceedings of 6th Applied Natural Language Processing Conference: 142–149.

Kohonen, Teuvo Kaski, Samuel Lagus, Krista Salojarvi, Jarkko Honkela, Jukka Paatero,Vesa Saarela, Anti. 2000. *Self organisation of a massive document collection*, IEEE Transactions on Neural Networks, 11(3): 574-585.

G. Ngai and R. Florian. 2001. *Transformation-Based Learning in the Fast Lane*, Proceedings of the NAACL'2001, Pittsburgh, PA: 40-47

R K Rao Pattabhi, L Sobha, and Amit Bagga. 2007. *Multilingual cross-document co-referencing,* Proceedings of 6th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC), March 29-30, 2007, Portugal:115-119

Rauber, Andreas Merkl, Dieter. 1999. *The SOMLib digital library system,* In the Proceedings of the 3rd European Conference on Research and Advanced Technology for Digital Libraries (ECDL'99), Paris, France. Berlin: 323-341.

P. Resnik. 1995. *Using information content to evaluate semantic similarity in taxonomy,* Proceedings of IJCAI: 448–453.

Salton, Gerald. 1989. *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer,* Reading, MA: Addison Wesley

Sobha L, and Vijay Sundar Ram. 2006. *Noun Phrase Chunker for Tamil,* Proceedings of the First National Symposium on Modeling and Shallow Parsing of Indian Languages (MSPIL), IIT Mumbai, India: 194-198.

Finding parallel texts on the web using cross-language information retrieval

Achim Ruopp
University of Washington,
Seattle, WA 98195, USA
achimr@u.washington.edu

Fei Xia
University of Washington
Seattle, WA 98195, USA
fxia@u.washington.edu

Abstract

Discovering parallel corpora on the web is a challenging task. In this paper, we use cross-language information retrieval techniques in combination with structural features to retrieve candidate page pairs from a commercial search engine. The candidate page pairs are then filtered using techniques described by Resnik and Smith (2003) to determine if they are translations. The results allow the comparison of efficiency of different parameter settings and provide an estimate for the percentage of pages that are parallel for a certain language pair.

1 Introduction

Parallel corpora are invaluable resources in many areas of natural language processing (NLP). They are used in multilingual NLP as a basis for the creation of translation models (Brown et. al., 1990), lexical acquisition (Gale and Church, 1991) as well as for cross-language information retrieval (Chen and Nie, 2000). Parallel corpora can also benefit monolingual NLP via the induction of monolingual analysis tools for new languages or the improvement of tools for languages where tools already exist (Hwa et. al., 2005; Padó and Lapata, 2005; Yarowsky and Ngai, 2001).

For most of the mentioned work, large parallel corpora are required. Often these corpora have limited availability due to licensing restrictions (Tiedemann and Nygaard, 2004) and/or are domain specific (Koehn, 2005). Also parallel corpora are only available for a limited set of language pairs. As a result, researchers look to the World Wide Web as a source for parallel corpora (Resnik and Smith, 2003; Ma and Liberman, 1999; Chen and Nie, 2000). Because of the web's world-wide reach and audience, many websites are bilingual, if not multilingual. The web is therefore a prime candidate as a source for such corpora especially for language pairs including resource-poor languages.

Resnik and Smith (2003) outlined the following three steps for identifying parallel text on the web:

(1) Locating pages that might have parallel translations
(2) Generating candidate page pairs that might be translations
(3) Structural filtering out of non-translation candidate pairs

In most of the previous work, Step (1) is performed in an ad-hoc manner using structural features that were observed in a limited set of samples of parallel pages. For example a language name in an HTML link is considered a strong indication that the page is also available translated to the language indicated by the link. The reason for this ad-hoc approach is that there aren't any standards as to how web developers structure multilingual web pages on a server. Often developers use language names or identifiers in uniform resource locators (URLs) to distinguish different language versions of a page on a server.

When Step (1) is performed using a commercial search engine, another obstacle to finding candidates for parallel pages comes into play: the results are always relevance-ranked for the end user. In this paper, instead of searching exclusively for structural features of parallel pages, we are adding a dictionary-based sampling technique, based on cross-language information retrieval for Step (1). We compare the URL results from each of our experiments with three different matching methods for Step (2). Finally, for Step (3), we adapted a filtering method from Resnik and Smith (2003) to determine whether or not a page pair is a true translation pair.

To estimate the percentage of parallel pages that are available for a certain language pair in relation to the total number of pages available in each of the two languages, we modified a technique that Bharat and Broder (1998) used to estimate overlaps of search engine indices.

We conducted our experiments on the English-German pair, but the described techniques are largely language-independent. The results of the experiments in this paper would allow researchers to choose the most efficient technique when trying to build parallel corpora from the web and guide research into further optimizing the retrieval of parallel texts from the web.

2 Methodology

The first step in finding parallel text on the web has two parts. The first part, the *sampling* procedure, retrieves a set S_1 of pages in the source language L_1 by sending sampling queries to the search engine. These sampling queries are structured in such a way that they retrieve pages that are likely to have translations. The second part, a *checking* procedure, retrieves a set S_2 of pages in target language L_2 that are likely to contain the translations of pages in S_1. The two procedures are described in Sections 2.1 and 2.2, respectively.

Step (2) *matches up* elements of S_1 and S_2 to generate a set of candidates for page pairs that could be translations of each other. This is explained in Section 2.3.

Step (3), a final *filtering* step, uses features of the pages to eliminate page pairs that are not translations of each other. The detail of the step is described in Sections 2.4. Figure 1 illustrates the different sets of pages and page pairs created by the three steps.

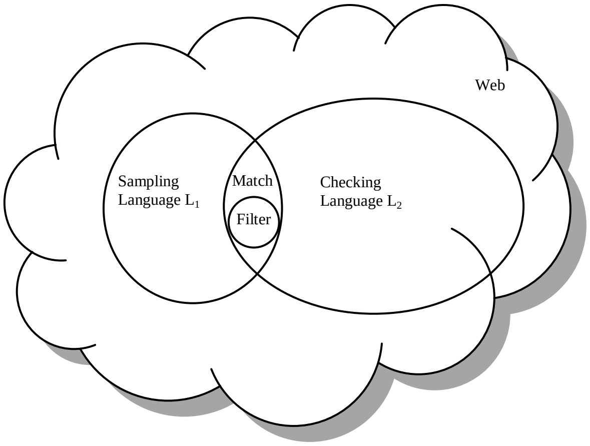


Figure 1. Pages and page pairs involved in the three steps of the algorithm

2.1 Sampling

For the baseline case the sampling should select pages randomly from the search space. To get a random sample of pages from a search engine that we can check for translational equivalents in another language, we select terms at random from a bilingual dictionary.

Instead of using a manually crafted bilingual dictionary, we chose to use a translation lexicon automatically created from parallel data, because the translation probabilities are useful for our experiments. In this study, the translation lexicon was created by aligning part of the German-English portion of the Europarl corpus (Koehn, 2005) using the Giza++ package (Och and Ney, 2003).

The drawback of using this translation lexicon is that the lexicon is domain-specific to parliamentary proceedings. We alleviated this domain-specificity by selecting mainly terms with medium frequency in the lexicon.

We sorted the terms by frequency. According to Zipf's law (Zipf, 1949), the frequency of the terms is roughly inversely proportional to their rank in this list. We choose terms according to a normal distribution whose mean is the midpoint of all ranks. We tuned the deviation to ¼ of the mean, so as to avoid getting very frequent terms into the sample which would just return a large set of unrelated pages, as well as very infrequent terms which would return few or no results.

A single word selected with this normal distribution, together with the `lang:` parameter set to language L_1, is submitted to the search engine to retrieve a sample (in our experiments 100 pages). The search engine automatically performs stemming on the term.

2.1.1 Source Language Expansion

To obtain a sample yielding more translation candidates, it is valuable to use semantically related multi-word queries for the sampling procedure.

To obtain semantically related words, we used the standard information retrieval (IR) technique of query expansion. Part of the sampling result of single-word queries are summaries, delivered back by the search engine. To come up with a ranked list of terms that are semantically related to the original one-word term, we extract and count all unigrams from a concatenation of the summaries. Stopwords are ignored. After the count, the unigrams are ranked by frequency.

For an n-term query, the original single-word query is combined with the first (n-1) terms of this ranked list to form an expanded query that is submitted to the search engine.

The advantage of this form of expansion is that it is largely language independent and often leads to highly relevant terms, due to the ranking algorithms employed by the search engines.

2.1.2 Language Identifiers in URLs

Once the baseline is established with single and multi-word sampling queries, an additional structural `inurl:` search parameter, which allows querying for substrings in URLs, can be added to increase the likelihood of finding pages that do have translations.

For this paper we limited our experiments to use standard (RFC 3066) two-letter language identifiers for this search parameter: “en” for English and “de” for German.

2.2 Checking

The purpose of the checking procedure is to generate a set of web pages in language L_2 that are potentially translations of pages in the sample obtained in the previous section.

2.2.1 Translating the Sampling Query

The natural way to do this is to translate the sampling query from language L_1 into the target language L_2. The sampling query does not necessarily have a unique one-to-one translation in language L_2. This is where the translation lexicon created from the Europarl corpus comes in. Because the lexicon contains probabilities, we can obtain the m-best translations for a single term from the sampling query.

Given a query in L_1 with n terms and each term has up to m translations, the checking procedure will form up to m^n queries in L_2 and sends each of them to the search engine. Because most current commercial search engines set a limit on the maximum number of queries allowed per day, longer sampling queries (i.e., larger n) mean that fewer overall samples can be retrieved per day. The effect of this trade-off on the number of parallel page pairs is evaluated in our experiments.

Source language expansion can lead to sample terms that are not part of the translation lexicon. These are removed during translation.

If the `inurl:` search parameter was used in the sampling query, the corresponding `inurl:` parameter for language L_2 will be used in the checking query.

2.2.2 Target Language Expansion

An alternative to translating all terms in an expanded, multi-word sampling query (see Section 2.1.1) is to translate only the original single sampling word to obtain top m translations in $L_{2,}$ and then for each translation do a term expansion on the target language side with (n-1) expansion terms. The benefit of target language expansion is that it only requires m checking queries, where source language expansion requires m^n checking queries. The performance of this different approach will be evaluated in Section 3.

2.2.3 Site Parameter

Another structural search parameter appropriate for checking is the `site:` parameter, which many search engines provide. It allows limiting the query results to a set of pre-defined sites. In our experiments we use the sites of the top-30 results of the sampling set, which is the maximum allowed by the Yahoo! search engine.

2.3 Matching Methods

To obtain page pairs that might be translations of each other, pages in sampling set S_1 are matched up based on URL similarity with pages in corresponding checking set S_2. We experimented with three methods.

2.3.1 Fixed Language List

In the fixed language list matching method, URLs differing only in the language names and language identifiers (as listed in Table 1) are considered a match and added to the set of page pair candidates.

en	de
en-us	de-de
en	ge
enu	deu
enu	ger
english	german
englisch	deutsch

Table 1. Language identifiers and language names for Fixed Language List and URL Part Substitution

An example for a match in this category is http://ec.europa.eu/education/policies/rec_qual/recognition/diploma_en.html and http://ec.europa.eu/education/policies/rec_qual/recognition/diploma_de.html.

2.3.2 Levenshtein Distance

In the Levenshtein distance matching method, if the Levenshtein distance (also known as edit distance) between a pair of URLs from S_1 and S_2 is larger than zero[1] and is below a threshold, the URL pair is considered a match. In our experiments, we set the threshold to four, because for most standard (RFC 3066) language identifiers the maximum Levenshtein distance would be four (e.g. "en-US" vs. "de-DE" as part of a URL).

2.3.3 URL Part Substitution

The third method that we tried does not require querying a search engine for a checking set. Instead, each URL U_1 in the sampling set S_1 is parsed to determine if it contains a language name or identifier at a word boundary. If so, the language name or identifier is substituted with the corresponding language name or identifier for the target language to form a target language URL U_2 according to the substitutions listed in Table 1.

For each resulting U_2, an HTTP HEAD request is issued to verify whether the page with that URL exists on the server. If the request is successful, the pair (U_1,U_2) is added to the set of page pair candidates. If multiple substitutions are possible for a U_1 all the resulting U_2 will be tested.

2.4 Page Filtering

The goal of this step is to filter out all the page pairs that are not true translations.

2.4.1 Structural Filtering

One method for filtering is a purely structural, language-independent method described in Resnik and Smith (2003). In this method, the HTML structure in each page is linearized and the resulting sequences are aligned to determine the structural differences between the two files. Their paper discussed four scalar values that can be calculated from the alignment. We used two of the values in our experiments, as described below.

The first one is called the difference percentage (dp), which indicates the percentage of nonshared material in the page pair. Given the two linearized sequences for a page pair ($p_{1,}$ p_2), we used Eq (1) to calculate dp, where $length_1$ is the length of the first sequence, and $diff_1$ is the number of lines in the first sequence that do not align to anything in the second sequence; $length_2$ and $diff_2$ are defined similarly.

$$dp\ (p_1, p_2) = \frac{diff_1 + diff_2}{length_1 + length_2} \quad (1)$$

The second value measures the correlation between the lengths of aligned nonmarkup chunks. The idea is that the lengths of corresponding translated sentences or paragraphs usually correlate. The longer a sentence in one language is, the longer its translation in another language should be. For the sake of simplicity, we assume there is a linear correlation between the lengths of the two files, and use the Pearson correlation coefficient as a length correlation metric. From the two linearized sequences, the lengths of nonmarkup chunks are recorded into two arrays. The Pearson correlation coefficient can be directly calculated on these two arrays.

[1] We don't want to match identical URLs.

This metric is denoted as r(p_1,p_2), and its value is in the range of [-1,1]: 1 indicates a perfect positive linear relationship, 0 indicates there is no linear relationship, and -1 indicates a perfect negative linear relationship between the chunk lengths in the two files.

2.4.2 Content Translation Metric

As shown in Resnik and Smith (2003), the structural filtering to judge whether pages are translations of each other leads to very good precision and satisfactory recall.

However, when using the URL part substitution method described in 2.3, many web sites, if they receive a request for a URL that does not exist, respond by returning the most likely page for which there is an existing URL on the server. This is often the page content of the original URL before substitution. Identical pages in the candidate page pair[2] would be judged as translations by the purely structural method and precision would be negatively impacted. There are several solutions for this, one of them is to use a content-based metric to complement the structural metric.

Ma and Liberman (1999) define the following similarity metric between two pages in a page pair (p_1, p_2):

$$c(p_1, p_2) = \frac{Num\ Of\ Translation\ Token\ Pairs}{Num\ Of\ Tokens\ in\ p_1} \quad (2)$$

To calculate this content-based metric, the translation lexicon created in Step (1) comes in handy. For the first 500 words of each page in the page pair candidate, we calculate the similarity metric in Eq (2), using the top two translations of the words in the translation lexicon.

2.4.3 Linear Combination

We combine the structural metrics (*dp* and *r*) and the content-based metric *c* by linear combination:[3]

$$t_{dprc}(p_1, p_2) = \frac{a_{dp} * (1 - dp(p_1, p_2)) + a_r * r(p_1, p_2) + a_c * c(p_1, p_2)}{3} \quad (3)$$

If t_{dprc} is larger than a predefined threshold, the page pair is judged to be a translation.

2.5 Estimating the Percentage of Parallel Pages for a Language Pair

Statistics on what share of web pages in one language have translated equivalents in another language are, to our knowledge, not available. Obtaining these statistics is useful from a web metrics perspective. The statistics allow the calculation of relative language web page counts and serve as a baseline to evaluate methods that try to find parallel pages.

Fortunately there is a statistical method (Bharat and Broder, 1998) that can be adapted to obtain these numbers. Bharat and Broder introduce a method to estimate overlaps in the coverage of a pair of search indices and to calculate the relative size ratio of search indices. They achieve this by randomly sampling pages in one index and then check whether the pages are also present in the other index.

Instead of calculating the overlap of pages in two search engines, we adapted the method to measure the overlap of languages in one search engine. Let P(E) represent the probability that a web page belongs to a set E. Let P(F_E|E) represent the conditional probability that there exist translational equivalents F of E given E. Then

$$P(F_E \mid E) = \frac{Size(F_E)}{Size(E)} \quad (4)$$

$$P(E_F \mid F) = \frac{Size(E_F)}{Size(F)} \quad (5)$$

Size(F_E) and Size(E) can be determined with an experiment using one term samples and checking with a `site:` parameter. Size (F_E) equals the number of page pairs that are determined to be translations by the filtering step. Size(E) is the number of checked sites per sample (30 in the case of Yahoo!) times the number of samples. Size(E_F) and Size(F) are calculated similarly.

3 Experiments

To evaluate the effectiveness of various methods described in Section 2, we ran a range of experiments and the results are shown in Table 2.

[2] The two identical pages could have different URLs.

[3] We use 1-dp(p_1,p_2) to turn a dissimilarity measure into a similarity measure.

The first column is the experiment ID; the second column indicates whether source or target expansion is used in Step (1); the third column shows the length of the queries after the expansion (if any). For instance, in Experiment #3, source expansion is applied, and after the expansion, the new queries always have three words: one is the original query term, and the other two are terms that are most relevant to the original query term according to the documents retrieved by the search engine (see Section 2.1.2).

The fourth and fifth columns indicate whether the `inurl:` and `site:` parameters are used during the search. A blank cell means that the search parameter is not used in that experiment. For each query, the search engine returns the top 100 documents.

The next three columns show the numbers of page pairs produced by each of the three matching methods describe in Section 2.3. The last three columns show the numbers of page pairs after the filtering step. Here, we used the linear combination (see Section 2.4.3). All the numbers are the sum of the results from 100 sampling queries. Let us examine the experimental results in detail.

3.1 Sampling and Checking

The evaluation of the sampling and checking procedures are difficult, because the number of translation page pairs existing on the web is unknown. In this study, we evaluated the module indirectly by looking at the translation pairs found by the following steps: the matching step and the filtering step.

A few observations are worth noting. First, query expansion increases the number of page pairs created in Steps (2) and (3), and source and target query expansion lead to similar results. However, the difference between n=2 and n=3 is not significant. One possible explanation is that the semantic divergence between queries on the source side and on the target side could become more problematic for longer queries.

Second, using the `site:` and `inurl:` search parameters (described in 2.1.2 and 2.2.3) increases the number of discovered page pairs. The potential limitation is that `inurl:` narrows the set of discoverable pages to the ones that contain language identifiers in the URL.

Experiment ID	Expansion type	Query length (n)	inurl: Param	site: Param	Number of page pairs (before filtering)			Number of page pairs (after filtering)		
					List	Levenshtein	Substitution	List	Levenshtein	Substitution
1	none	1			5	13	1108	1	1	97
2	Source	2			8	28	1889	3	4	157
3	Source	3			10	42	1975	1	10	124
4	none	1	en/de		58	84	5083	17	22	285
5	Source	2	en/de		72	132	9279	27	31	433
6	Source	3	en/de		100	160	9200	25	31	347
7	none	1			6	18	1099	1	3	92
8	Target	2			4	24	1771	2	3	143
9	Target	3			4	12	1761	0	0	149
10	none	1	en/de		56	93	5041	24	34	281
11	Target	2	en/de		107	161	9131	27	33	426
12	Target	3	en/de		45	72	8395	12	15	335
13	none	1		30	10	258	n/a	6	9	n/a
14	Source	2		30	22	743	n/a	9	32	n/a
15	Source	3		30	46	1074	n/a	12	41	n/a
16	none	1	en/de	30	59	164	n/a	13	15	n/a
17	Source	2	en/de	30	118	442	n/a	28	50	n/a
18	Source	3	en/de	30	171	693	n/a	46	49	n/a

Table 2. Experiment configurations and results

3.2 Matching Methods

Table 2 shows that among the three matching methods, the URL part substitution method leads to many more translation page pairs than the other two methods.

Notice that although the fixed language list method uses the same language name pair table (i.e., Table 1) as the URL part substitution method, it works much worse than the latter. This is largely due to the different rankings of documents in different languages. For instance, suppose a page p_1 in L_1 is retrieved by a sampling query $q_{1,}$ and p_1 has a translation page p_2 in L_2 , it is possible that p_2 will not be retrieved by the query q_2, a query made up of the translation of the terms in q_1 .[4]

Another observation is that the Levenshtein distance matching method outperforms the fixed language list method. In addition, it has a unique advantage: the results allow the automatic learning of language identifiers that web developers use in URLs to distinguish parallel pages for certain language pairs.

3.3 Parameter Tuning for Linear Combination of Filtering Metrics

Before the combined metrics in Eq (3) can be used to filter page pairs, the combination parameters need to be tuned on a development set. The parameters are a_{dp}, a_r, and a_c as well as the threshold above which the combined metrics indicate a translated page pair vs. an unrelated page pair.

To tune the parameters, we used data from an independent test run for the en→de language direction. We randomly chose 50 candidate pairs from a set created with the URL part substitution method and manually judged whether or not the pages are translations of each other.

We varied the parameters a_{dp}, a_r, a_c and t_{dprc} over a range of empirical values and compared how well the combined metrics judgment correlated with the human judgment for page translation (we calculated the Pearson correlation coefficient). The results of tuning are shown in Table 3.

a_{dp}	a_r	a_c	t_{dprc}
0.5	1.5	1	> 0.8

Table 3: Parameter and threshold values chosen for linear combination

3.4 Evaluation of the Filtering Step

To evaluate the combined filtering method described in Section 2.4.3, we chose 110 page pairs at random from the 433 candidate page pairs in experiment #5 (Language direction en→de, Pairs generated with the URL part substitution method described in 2.3.3). Each of the page pairs was evaluated manually to assess whether it is a true translation pair.

On this set, the combined filter had a precision of 88.9% and a recall of 36.4%. The high precision is encouraging on the noisy test set. The recall is low but is acceptable since one can always submit more sampling queries to the search engine. Resnik and Smith (2003) reported higher precision and recall in their experiments. However, their numbers and ours are not directly comparable because their approach required the existence of parent or sibling pages and consequently their test sets were less noisy.

From the numbers of translation pairs, we can make an estimate of available parallel pages for a language pair, as explained in Section 2.5. For instance, by using the results of experiment #13, the estimate is $P(D_E|E)$=0.03% and $P(E_D|D)$=0.27% (E for English, and D for German). This indicates that the number of English-German parallel pages is small comparing to the total number of English and German web pages.

4 Conclusion

In this paper we show that despite the fact that there are no standardized features to identify parallel web pages and despite the relevance ranking of commercial search engine results, it is possible to come up with reliable methods to gather parallel pages using commercial search engines. It is also possible to calculate an estimate of how many pages are available parallel in relation to the overall number of pages in a certain language.

The number of translation pages retrieved by the current methods is relatively small. In the future, we plan to learn URL patterns from the Levenshtein matching method and add them to the patterns used in the URL part substitution method. Once more translation pages are retrieved, we plan to use these pages as parallel data in a statistical machine translation (MT) system to evaluate the usefulness of this approach to MT.

[4] The search engine returns 100 or fewer documents for each query.

Instead of using narrow query interfaces to a public search engine interface, it also might be advantageous to have access to raw indices or crawl data of the engines. Such access will enable us to take advantage of certain page features that could be good indicators of parallel pages.

References

Krishna Bharat and Andrei Broder , 1998, *A technique for measuring the relative size and overlap of public Web search engines*, Computer Networks and ISDN Systems 30(1-7).

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin, 1990, *A statistical approach to machine translation*, Computational Linguist. **16**(2), pp 79-85.

Jiang Chen and Jian-Yun Nie, 2000, *Parallel Web Text Mining for Cross-Language IR, in* Proceedings of RIAO-2000: Content-Based Multimedia Information Access.

William A. Gale and Kenneth W. Church, 1991, *Identifying word correspondence in parallel texts, in* 'HLT '91: Proceedings of the workshop on Speech and Natural Language', NJ, USA, pp. 152-157.

J.W. Hunt and M.D. McIlroy, M.D., 1976, *An Algorithm for Differential File Comparison*, Bell Laboratories, Computer Science Technical Report **41**.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak, 2005, *Bootstrapping Parsers via Syntactic Projection across Parallel Texts*, Special Issue of the Journal of Natural Language Engineering on Parallel Texts **11**(3), pp 311-325.

Philipp Koehn, 2005, *Europarl: A Parallel Corpus for Statistical Machine Translation, in Proceedings of the* 2005 MT Summit.

Xiaoyi Ma and Mark Y. Liberman, 1999, *BITS: A Method for Bilingual Text Search over the Web,* in Proceedings of the 1999 MT Summit, Singapore.

Franz Josef Och and Hermann Ney, 2003, *A Systematic Comparison of Various Statistical Alignment Models*, Computational Linguistics **29**(1), pp 19-51.

Sebastian Padó and Mirella Lapata, 2005, *Cross-linguistic Projection of Role-Semantic Information, in* Proceedings of HLT/EMNLP 2005.

Philip Resnik and Noah A. Smith, 2003, *The Web as a Parallel Corpus*, Computational Linguistics **29**(3), pp 349-380.

Jörg Tiedemann and Lars Nygaard, 2004, *The OPUS corpus - parallel & free, in* 'Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)'.

David Yarowsky and Grace Ngai, 2001, *Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora, in* 'Proceedings of the Second meeting of the North American Chapter of ACL (NAACL 2001)', NJ, USA, pp. 1-8.

George K. Zipf, 1949, *Human Behavior and the Principle of Least-Effort*, Addison-Wesley

Some Experiments in Mining Named Entity Transliteration Pairs from Comparable Corpora

K Saravanan
Microsoft Research India
Bangalore, India
v-sarak@microsoft.com

A Kumaran
Microsoft Research India
Bangalore, India
kumarana@microsoft.com

Abstract

Parallel Named Entity pairs are important resources in several NLP tasks, such as, CLIR and MT systems. Further, such pairs may also be used for training transliteration systems, if they are transliterations of each other. In this paper, we profile the performance of a mining methodology in mining parallel named entity transliteration pairs in English and an Indian language, Tamil, leveraging linguistic tools in English, and article-aligned comparable corpora in the two languages. We adopt a methodology parallel to that of [Klementiev and Roth, 2006], but we focus instead on mining parallel named entity transliteration pairs, using a well-trained linear classifier to identify transliteration pairs. We profile the performance at several operating parameters of our algorithm and present the results that show the potential of the approach in mining transliterations pairs; in addition, we uncover a host of issues that need to be resolved, for effective mining of parallel named entity transliteration pairs.

1 Introduction & Motivation

Parallel Named Entity (NE) pairs are important resources in several NLP tasks, from supporting Cross-Lingual Information Retrieval (CLIR) systems, to improving Machine Translation (MT) systems. In addition, such pairs may also be used for developing transliteration systems, if they are transliterations of each other. Transliteration of a name, for the purpose of this work, is defined as its transcription in a different language, preserving the phonetics, perhaps in a different orthography [Knight and Graehl, 1997][1]. While traditional transliteration systems have relied on hand-crafted linguistic rules, more recently, statistical machine learning techniques have been shown to be effective in transliteration tasks [Jung et al., 2000] [AbdulJaleel and Larkey, 2003] [Virga and Kudhanpur , 2003] [Haizhou et al., 2004]. However, such data-driven approaches require significant amounts of training data, namely pairs of names in two different languages, possibly in different orthography, referred to as *transliteration pairs*, which are not readily available in many resource-poor languages. It is important to note at this point, that NEs are found typically in news corpora in any given language. In addition, news articles covering the same event in two different languages may reasonably be expected to contain the same NEs in the respective languages. The perpetual availability of news corpora in the world's languages, points to the promise of mining transliteration pairs endlessly, provided an effective identification of such NEs in specific languages and pairing them appropriately, could be devised.

Recently, [Klementiev and Roth, 2006] outlined an approach by leveraging the availability of article-aligned news corpora between English and Russian, and tools in English, for discovering transliteration pairs between the two languages, and progressively refining the discovery process. In this paper, we adopt their basic methodology, but we focus on 3 different issues:

[1] *London* rewritten as லண்டன் in Tamil, or لندن in Arabic (both pronounced as *London*), are considered as transliterations, but not the rewriting of *New Delhi* as புது தில்லி (*puthu thilli*) in Tamil.

1. mining comparable corpora for NE pairs, leveraging a well trained classifier,
2. calibrating the performance of this mining framework, systematically under different parameters for mining, and,
3. uncovering further research issues in mining NE pairs between English and an Indian language, Tamil.

While our analysis points to a promising approach for mining transliteration pairs, it also uncovers several issues that may need to be resolved, to make this process highly effective. As in [Klementiev and Roth, 2006] no language specific knowledge was used to refine our mining process, making the approach broadly applicable.

2 Transliteration Pairs Discovery

In this section, we outline briefly the methodology presented in [Klementiev and Roth, 2006], and refer interested readers to the source for details.

They present a methodology to automatically discover parallel NE transliteration pairs between English and Russian, leveraging the availability of a good-quality Named Entity Recognizer (NER) in English, and article-aligned bilingual comparable corpora, in English and Russian. The key idea of their approach is to extract all NEs in English, and identify a set of potential transliteration pairs in Russian for these NEs using a simple classifier trained on a small seed corpus, and re-ranking the identified pairs using the similarity between the frequency distributions of the NEs in the comparable corpora. Once re-ranked, the candidate pairs, whose scores are above a threshold are used to re-train the classifier, and the process is repeated to make the discovery process more effective.

To discriminate transliteration pairs from other content words, a simple perceptron-based linear classifier, which is trained on *n*-gram features extracted from a small seed list of NE pairs, is employed leveraging the fact that transliteration relies on approximately monotonic alignment between the names in two languages. The potential transliteration pairs identified by this classifier are subsequently re-ranked using a Discrete Fourier Transform based similarity metric, computed based on the frequency of words of the candidate pair, found in the article-aligned comparable corpora. For the frequency analysis, equivalence classes of the words are formed, using a common prefix of 5 characters, to account for the rich morphology of Russian language. The representative prefix of each of the classes are used for classification.

Finally, the high scoring pairs of words are used to re-train the perceptron-based linear classifier, to improve the quality of the subsequent rounds. The quality of the extracted NE pairs is shown to improve, demonstrating viability of such an approach for successful discovery of NE pairs between English and Russian.

3 Adoption for Transliteration Pairs Mining

We adopt the basic methodology presented in [Klementiev and Roth, 2006], but we focus on three specific issues described in the introduction.

3.1 Mining of Transliteration Pairs

We start with comparable corpora in English and Tamil, similar in size to that used in [Klementiev and Roth, 2006], and using the English side of this corpora, first, we extract all the NEs that occur more than a given threshold parameter, F_E, using a standard NER tool. The higher the threshold is, the more will be the evidence for legitimate transliteration pairs, in the comparable corpora, which may be captured by the mining methodology. The extracted list of NEs provides the set of NEs in English, for which we mine for transliteration pairs from the Tamil side of the comparable corpora.

We need to identify all NEs in the Tamil side of the corpora, in order to appropriately pair-up with English NEs. However, given that there is no publicly available NER tool in Tamil (as the case may be in many resource-poor languages) we start with an assumption that all words found in the Tamil corpus are potentially NEs. However, since Tamil is a highly morphologically inflected language, the same NE may occur in its various inflected forms in the Tamil side of the corpora; hence, we collect those words with the same prefix (of fixed size) into a single bucket, called *equivalence class*, and consider a representative prefix, referred to as *signature* of the collection for comparison. The assumption here is that the common prefix would stand for a Tamil NE, and all the members of the equivalence class are the various inflected forms of the NE. We use such a signature to classify a Tamil word as potential transliteration of an English word. Again, we consider only those signatures that have occurred more than a threshold parameter, F_T, in the Tamil side of the comparable corpora, in order to strengthen support for a meaningful similarity in their frequency of occurrence.

We used a linear Support Vector Machine classifier (details given in a later section) trained on a sizable seed corpus of transliterations between English and Tamil, and use it to identify potential Tamil signatures with any of the NEs extracted from the English side. We try to match each of the NEs extracted from the English side, to every signature from the Tamil side, and produce an ordered list of Tamil signatures that may be potential transliterations for a given English NE. Every Tamil signature, thus, would get a score, which is used to rank the signatures in the decreasing order of similarity. Subsequently, we consider only those above a certain threshold for analysis, and in addition, consider only the top-*n* candidates.

3.2 Quality Refinement

Since a number of such transliteration candidates are culled from the Tamil corpus for a given NE in English, we further cull out unlikely candidates, by re-ranking them using frequency cues from the aligned comparable corpora. For this, we start with the hypothesis, that the NEs will have similar normalized frequency distributions with respect to time, in the two corpora. Given that the news corpora are expected to contain same names in similar time periods in the two different languages, the frequency distribution of words in the two languages provides a strong clue about possible transliteration pairs; however, such potential pairs might also include other content words, such as, சோஷலிஸ்ட் (*soshaliSt*), கவனமாக (*kavanamaaka*), கேட்பது (*keetpathu*), etc., which are common nouns, adjectives or even adverbs and verbs. On the other hand, function words are expected to be uniformly distributed in the corpus, and hence may not have high variability like content words. Note that the NEs in English are not usually inflected. Since Tamil NEs usually have inflections, the frequency of occurrence of a NE in Tamil must be normalized across all forms, to make it reasonably comparable to the frequency of the corresponding English NE. This was taken care of by considering the signature and its equivalence class. Hence the frequency of occurrence of a NE (i.e., its signature) in Tamil is the sum of frequencies of all members in its equivalence class.

For identifying the names between the languages, we first create a frequency distribution of every word in English and Tamil, by creating temporal bins of specific duration, covering the entire timeline of the corpus. The frequency is calculated as the number of occurrences of each signature in the bin interval. Once the frequency distributions are formed, they are normalized for every signature. Given the normalized frequencies, two words are considered to have same (or, similar) pattern of occurrence in the corpus, if the normalized frequency vectors of the two words are the same (or, close within a threshold). Figure 1 shows the frequency of the word *Abishek*, and its Tamil version, அபிஷேக் (*apishek*) as a frequency plot, where a high correlation between the frequencies can be observed.

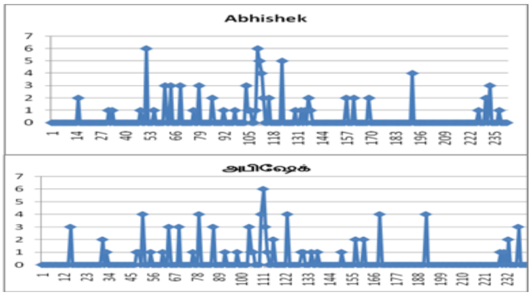


Figure 1: Names Frequency Plot in Comparable Corpora

Hence, to refine the quality of the classifier output, we re-rank the list of candidates, using the distance between the frequency vectors of the English NE, and the Tamil candidate signature. This step moves up those signatures that have similar patterns of occurrence, and moves down those that do not. It is likely that such frequency cues from the comparable corpora will make the quality of matched transliteration pairs better, yielding better mined data.

4 Experimental Setup & Results

In this section, we present the experimental setup and the data that we used for mining transliteration pairs from comparable corpora in two languages: English and the Indian language, Tamil. We evaluate and present the effectiveness of the methodology in extracting NE pairs, between these languages, under various parameters.

4.1 Comparable Corpora

We used a set of news articles from the New Indian Express (in English) and Dinamani (in Tamil) roughly covering similar events in English and Tamil respective, and covering a period of about 8 months, between January and August of 2007. The articles were verified to contain similar set of NEs, though only a fraction of them are expected to be legitimate transliteration pairs. Others related NEs could be translations, for example, *chief minister* in English vs முதல்வர் (*muthalvar*) in Tamil, abbreviation which are not usually transliterated but spelled out , for example, *ICC* in English, and ஐசிசி (*aicici*) in Tamil, or co-references , for example, *New Delhi* in English, and புதுதில்லி (*puthu thilli*) in Tamil. While the number of articles used were roughly the same (~2,400), the number of words in Tamil were only about 70% of that in English. This is partially due to the fact Tamil is a highly agglutinative language, where various affixes (prefixes and suffixes of other content words) stand for function words and prepositions in English, thus do not contribute to the word count. Further, since our focus is on mining names, we expect the same NEs to be covered in both the corpora, and hence we do not expect a severe impact on mining.

Corpus	Time Period	Size	
		Articles	Words
New Indian Express (English)	2007.01.01 to 2007.08.31	2,359	347,050
Dinamani (Tamil)	2007.01.01 to 2007.08.31	2,359	256,456

Table 1: Statistics on Comparable Corpora

From the above corpora, we first extracted all the NEs from the English side, using the Stanford NER tool [Finkel et al, 2005]. No multiword expressions were considered for this experiment. Also, only those NEs that have a frequency count of more than a threshold value of F_E were considered, in order to avoid unusual names that are hard to identify in the comparable corpora. Thus, we extracted from the above corpora, only a subset of NEs found in the English side to be matched with their potential transliteration pairs; for example, for a parameter setting of F_E to 10, we extract only 274 legitimate NEs.

From the Tamil side of the corpora, we extracted all words, and grouped them in to equivalence classes, by considering a prefix of 5 characters. That is, all words that share the same 5 characters were considered to be morphological variations of the same root word or NE in Tamil. After they were grouped, the longest common prefix of the group is extracted, and is used as the signature of the equivalence class. It should be noted here that though the number of unique words in the corpus is about 46,503, the number of equivalence classes to be considered changes depending on the filtering threshold that we use in the Tamil side. For example, at a threshold (F_T) value of 1, the number of equivalence classes is 14,101. It changes to 4,612 at a threshold (F_T) value of 5, to 2,888 at a threshold (F_T) value of 10 and to 1779 at a threshold (F_T) value of 20. However, their signature (i.e., longest common prefix) sizes ranged from 5 to 13 characters. Thus, we had about 14,101 equivalence classes, covering all the words from the Tamil corpus. The equivalence classes thus formed were as shown in Figure 2:

Tamil Signature	Tamil Equiv. Class
ஐஸ்வர்யா (*aiSvaryaa*)	ஐஸ்வர்யா (*aiSvaryaa*), ஐஸ்வர்யாவின் (*aiSvaryaavin*), ஐஸ்வர்யாவுக்கு (*aiSvaryaavukku*), ஐஸ்வர்யாவை (*aiSvaryaavai*), ஐஸ்வர்யாவிற்கும் (*aiSvaryaaviRkum*), ஐஸ்வர்யாவுடன் (*aiSvaryaavutan*)
பிரம் (*piram*)	பிரம்மபுத்திரா (*pirammapuththiraa*), பிரம்மாண்டமான (*pirammaaNdamaana*), பிரம்பு (*pirampu*), பிரம்மா (*pirammaa*)
காவேரி (*kaaveeri*)	காவேரி (*kaaveeri*)
ஐசிசி (*aicici*)	ஐசிசி (*aicici*), ஐசிசியின் (*aicicyin*), ஐசிசிக்க (*aicici kku*), ஐசிசிதான் (*aicicithaan*), ஐசிசியிடம் (*aiciciyidam*)

Figure 2: Signatures and Equivalence Classes

As can be seen in the table, all elements of an equivalence class share the same signature (by definition). However, some signatures, such as ஐஸ்வர்யா (*aiSvaryaa*), correspond to an equivalence class in which every element is a morphological variation of the signature. Such equivalence classes, we name them *pure*. Some signatures represent only a subset of the members, as this set includes some members unrelated to this stem; for example, the signature பிரம் (*piram*), correctly corresponds to பிரம்மா (*pirammaa*), and incorrectly to the noun பிரம்பு (*pirambu*), as well as incorrectly to the adjective பிரம்மாண்டமான (*pirammaandamaana*). We name such equivalence classes *fuzzy*. Some are well formed, but may not ultimately contribute to our mining, being an abbreviation, such as *ICC* (in Tamil, ஐசிசி), even though they are used similar to any NE in Tamil. While most equivalence classes contained inflections of single stems, we also found morphological variations of several compound names in the same equivalence class such as, அகமத்நகர் (*akamathñakar*), அகமதாபாத் (*akamathaapaath*), with அகமத் (*akamath*).

4.2 Classifier for Transliteration Pair Identification

We used SVM-light [Joachims, 1999], a Support-vector Machine (SVM) from Cornell University, to identify near transliterations between English and Tamil. We used a seed corpus consisting of 5000 transliteration pair samples collected from a different resource, unrelated to the experimental comparable corpora. In addition to the 5000 positive examples from this seed corpus, 5000 negative examples were extracted randomly, but incorrectly, aligned names from this same seed corpus and used for the classifier.

The features used for the classification are binary features based on the length of the pair of strings and all aligned unigram and bigram pairs, in each direction, between the two strings in the seed corpus in English and Tamil. The length features include the difference in lengths between them (up to 3), and a separate binary feature if they differ by more than 3. For unigram pairs, the i^{th} character in a language string is matched to $(i\text{-}1)^{st}$, i^{th} and $(i+1)^{st}$ characters of the other language string. Each string is padded with special characters at the beginning and the end, for appropriately forming the unigrams for the first and the last characters of the string. In the same manner, for binary features, every bigram extracted with a sliding window of size 2 from a language string, is matched with those extracted from the other language string. After the classifier is trained on the seed corpus of hand crafted transliteration pairs, during the mining phase, it compares every English NE extracted from the English corpus, to every signature from the Tamil corpus.

While classifier provided ranked list of all the signatures from Tamil side, we consider only the top-30 signatures (and the words in the equivalence classes) for subsequent steps of our methodology. We hand-verified a random sample of about 100 NEs from English side, and report in Table 5, the fraction of the English NEs for which we found at least one legitimate transliteration in the top-30 candidates (for example, the recall of the classifier is 0.56, in identifying a right signature in the top-30 candidates, when the threshold F_E is 10 & F_T is 1).

It is interesting to note that as the two threshold factors are increased, the number of NEs extracted from the English side decreases (as expected), and the average number of positive classifications per English NE reduces (as shown in Table 2), considering all NEs. This makes sense as the classifier for identifying potential transliterations is trained with sizable corpora and is hence accurate; but, as the thresholds increase, it has less data to work with, and possibly a fraction of legitimate transliterations also gets filtered with noise.

Parameters	Extracted English NEs	Ave. Positive Classifications/ English NE
F_E: 10, F_T: 1	274	79.34
F_E: 5, F_T: 5	588	29.50
F_E: 10, F_T: 10	274	17.49
F_E: 20, F_T: 20	125	10.55

Table 2: Threshold Parameters vs Mining Quantity

Table 3 shows some sample results after the classification step with parameter values as (*F_E: 10, F_T: 1*). Right signature for *Aishwarya* (corresponding to all correct transliterations) has been ranked 10 and *Gandhi* (with only a subset of the equivalence class corresponding to the right transliterations) has been ranked at 8. Three different variations of *Argentina* can be found, ranked 2nd, 3rd and 13th. While, in general no abbreviations are found (usually their Tamil equivalents are spelled out), a rare case of abbreviation (*SAARC*) and its right transliteration is ranked 1st.

English Named Entity	Tamil Equivalence Class Signature	Precision	Rank
aishwarya	ஐஸ்வர்யா (*aiSvaryaa*)	1	10
argentina	அர்ஜன்டினாவில (*arjantinaavila*)	1	2
argentina	ஆர்ஜென்டினாவி (*aarjantinaavi*)	1	3
argentina	ஆர்ஜன்டினாவில் (*aarjantinaavil*)	1	13
gandhi	காந்த (*kaañtha*)	0.2121	8
saarc	சார்க் (*saark*)	1	1

Table 3: Ranked List after Classification Step

4.3 Enhancing the Quality of Transliteration-Pairs

For the frequency analysis, we use the frequency distribution of the words in English and Tamil side of the comparable corpora, counting the number of occurrences of NEs in English and the Tamil signatures in each temporal bin spanning the entire corpus. We consider one temporal bin to be equal to two successive days. Thus, each of the English NEs and the Tamil signatures is represented by a vector of dimension approximately 120. We compute the distance between the two vectors, and hypothesize that they may represent the same (or, similar) name, if the difference between them is zero (or, small). Note that, as mentioned earlier, the frequency vector of the Tamil signature will contain the sum of individual frequencies of the elements in the equivalence class corresponding to it. Given that the classifier step outputs a list of English NEs, and associated with each entry, a ranked list of Tamil signatures that are identified as potential transliteration by the classifier, we compute the distance between the frequency vector of every English NE, with each of the top-30 signatures in the ranked list. We re-rank the top-30 candidate strings, using this distance measure. The output is similar to that shown in Table 4, but with possibly a different rank order.

English Named Entity	Tamil Equivalence Class Signature	Precision	Rank
aishwarya	ஐஸ்வர்யா (*aiSvaryaa*)	1	1
argentina	அர்ஜன்டினாவில (*arjantinaavila*)	1	1
argentina	ஆர்ஜென்டினாவி (*aarjantinaavi*)	1	3
argentina	ஆர்ஜன்டினாவில் (*aarjantinaavil*)	1	14
gandhi	காந்த (*kaañtha*)	0.2121	16
saarc	சார்க் (*saark*)	1	1

Table 4: Ranked List after Frequency Analysis Step

On comparing Table 3 and 4, we observe that some of the ranks have moved for the better, and some of them for the worse. It is interesting to note that the ranking of different stems corresponding to Argentina has moved differently. It is quite likely that merging these three equivalence classes corresponding to the English NE *Argentina* might result in a frequency profile that is more closely aligned to that of the English NE.

4.4 Overall Performance of Transliteration Pairs Mining

To find the effectiveness of each step of the mining process in identifying the right signatures (and hence, the equivalence classes) for a given English NE, we computed the Mean Reciprocal Rank (MRR) of the random sample of 100 transliteration pairs mined, in two different ways: First, we computed MRR_{pure}, which corresponded to the first occurrence of a pure equivalence class, and MRR_{fuzzy}, which corresponded to the first occurrence of a fuzzy equivalence class in the random samples. MRR_{fuzzy} captures how successful the mining was in identifying one possible transliteration, MRR_{pure}, captures how successful we were in identifying an equivalence class that contains only right transliterations[2]. In addition, these metrics were computed, corresponding to different frequency thresholds for the occurrence of a English NE (F_E) and a Tamil signature (F_T). The overall quality profile of the mining framework in mining the NE transliteration pairs in English and Tamil is shown in Table 5. Additionally, we also report the *recall* metric (*the fraction of English NEs, for which at least one le-*

[2] However, it should be noted that the current metrics neither capture how pure an equivalence class is (fraction of the set that are correct transliterations), nor the size of the equivalence class. We hope to specify these as part of quality of mining, in our subsequent work.

gitimate Tamil signature was identified) computed on a randomly chosen 100 entity pairs.

Parameters	Classification Step		Frequency Analysis Step		Re-call
	MRR fuzzy	MRR pure	MRR fuzzy	MRR pure	
F_E: 10, F_T: 1	0.3579	0.2831	0.3990	0.3145	0.56
F_E: 5, F_T: 5	0.4490	0.3305	0.5064	0.3529	0.61
F_E: 10, F_T: 10	0.4081	0.2731	0.4930	0.3494	0.57
F_E: 20, F_T: 20	0.3489	0.2381	0.4190	0.2779	0.47

Table 5: Quality Profile of NE Pairs Extraction

First, it should be noted that the recalls are the same for both the steps, since Frequency Analysis step merely re-arranges the output of the Classification step. Second, the recall figures drop, as more filtering is applied to the NEs on both sides. This trend makes sense, since the classifier gets less data to work with, as more legitimate words are filtered out with noise. Third, as can be expected, MRR_{pure} is less than the MRR_{fuzzy} at every step of the mining process. Fourth, we see that the MRR_{pure} and the MRR_{fuzzy} improve between the two mining steps, indicating that the time-series analysis has, in general, made the output better.

Finally, we find that the MRR_{pure} and the MRR_{fuzzy} keep dropping with increased filtering of English NEs and Tamil signatures based on their frequency, in both the classification and frequency analysis steps. The fall of the MRRs after the classification steps is due to the fact that the classifier has less and less data with the increasing threshold, and hence some legitimate transliterations may be filtered out as noise. However, the frequency analysis step critically depends on availability of sufficient words from the Tamil side for similarity testing. In frequency analysis step, the fall of MRRs from threshold 5 to 10 is 0.0134 on MRR_{fuzzy} and 0.0035 on MRR_{pure}. This fall is comparatively less to the fall of MRRs from threshold 10 to 20 which is 0.074 on MRR_{fuzzy} and 0.0715 on MRR_{pure}. This may be due to the fact that the number of legitimate transliterations filtered out from threshold 5 to 10 is less when compared to the number of legitimate transliterations filtered out from threshold 10 to 20. These results show that with less number of words filtered, it can get reasonable recall and MRR values. More profiling experiments may be needed to validate this claim.

5 Open Issues in NE pair Mining

In this paper, we outline our experience in mining parallel NEs between English and Tamil, in an approach similar to the one discussed in [Klementiev and Roth, 2006]. Over and above, we made parameter choices, and some procedural modifications to bridge the underspecified methodology given in the above work. While the results are promising, we find several issues that need further research. We outline some of them below:

5.1 Indistinguishable Signatures

Table 7 shows a signature that offers little help in distinguishing a set of words. Both the words, சென்னை (*cennai*) and morphological variations of சென் (*cen*), share the same 5-character signature, namely, சென்ன (*cenna*), affecting the frequency distribution of the signature adversely.

English Named Entity	Tamil Named Entity	Tamil Equivalent Class
chennai	சென்னை (*cennai*)	சென்னை (*cennai*), சென்னையில் (*cennaiyil*), சென்னையிலிருந்து (*cennaiyilirunthu*), சென்னின் (*cennin*), சென்னுக்கு (*cen-nukku*),சென்னையை (*cennaiyai*)

Table 7: Multiple-Entity Equivalence Class

5.2 Abbreviations

Table 8 shows a set of abbreviations, that are not identified well in our NE pair mining. Between the two languages, the abbreviations may be either expanded, as *BJP* expanded to (the equivalent translation for *Bharatiya Janatha Party* in Tamil), or spelled out, as in *BSNL* referred to as பிஎஸ்என்எல் (*pieSenel*). The last example is very interesting, as each *W* in English is written out as டபிள்யூ (*tapiLyuu*). All these are hard to capture by a simple classifier that is trained on well-formed transliteration pairs.

English Named Entity	Tamil Named Entity
BJP	பாஜக (*paajaka*), பா.ஜ.க. (*paa. ja. ka.*), பாரதீய ஜனதா கட்சி (*paarathiiya janathaa katci*)
BSNL	பிஎஸ்என்எல் (*pieSenel*), பிஎஸ்என்எல்லின் (*pieSenellin*), பிஎஸ்என்எல்லை (*piesenellai*)
WWW	டபிள்யூடபிள்யூடபிள்யூ (*tapiLyuutapiLyuutapiLyuu*)

Table 8: Multiple-Entity Equivalence Class

5.3 Multiword Expressions

This methodology is currently designed for mining only single word expressions. It may be an interesting line of research to mine multiword expressions automatically.

6 Related Work

Our work essentially follows a similar procedure as reported in [Klementiev and Roth, 2006] paper, but applied to English-Tamil language pair. Earlier works, such as [Cucerzan and Yarowsky, 1999] and [Collins and Singer, 1999] addressed identification of NEs from untagged corpora. They relied on significant contextual and morphological clues. [Hetland, 2004] outlined methodologies based on time distribution of terms in a corpus to identify NEs, but only in English. While a large body of literature exists on transliteration, we merely point out that the focus of this work (based on [Klementiev and Roth, 2006]) is not on transliteration, but mining transliteration pairs, which may be used for developing a transliteration system.

7 Conclusions

In this paper, we focused on mining NE transliteration pairs in two different languages, namely English and an Indian language, Tamil. While we adopted a methodology similar to that in [Klementiev and Roth, 2006], our focus was on mining parallel NE transliteration pairs, leveraging the availability of comparable corpora and a well-trained linear classifier to identify transliteration pairs. We profiled the performance of our mining framework on several parameters, and presented the results. Our experiment results are inline with those reported by [Klementiev and Roth, 2006]. Given that the NE pairs are an important resource for several NLP tasks, we hope that such a methodology to mine the comparable corpora may be fruitful, as comparable corpora may be freely available in perpetuity in several of the world's languages.

8 Acknowledgements

We would like to thank Raghavendra Udupa, Chris Quirk, Aasish Pappu, Baskaran Sankaran, Jagadeesh Jagarlamudi and Debapratim De for their help.

References

Nasreen AbdulJaleel and Leah S. Larkey. 2003. Statistical transliteration for English-Arabic cross language information retrieval. In *Proceedings of CIKM*, pages 139–146, New York, NY, USA.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pp. 363-370.

L Haizhou, Z Min and S Jian. 2004. A Joint Source-Channel Model for Machine Transliteration. In *Proceedings of 42nd Meeting of Assoc. of Computational Linguistics*.

Magnus Lie Hetland. 2004. *Data Mining in Time Series Databases*, a chapter in A Survey of Recent Methods for Efficient Retrieval of Similar Time Sequences. World Scientific.

T. Joachims. 1999. 11 in: Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT Press.

Sung Young Jung, SungLim Hong, and Eunok Paek. 2000. An English to Korean transliteration model of extended markov window. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 383–389.

Alexandre Klementiev and Dan Roth. 2006. Named Entity Transliteration and Discovery from Multilingual Comparable Corpora. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, pages 82–88.

Kevin Knight and Jonathan Graehl. 1997. Machine transliteration. In *Proceedings of the Meeting of the European Association of Computational Linguistics*, pages 128–135.

Yusuke Shinyama and Satoshi Sekine. 2004. Named entity discovery using comparable news articles. In *Proceedings the International Conference on Computational Linguistics (COLING)*, pages 848–853.

Richard Sproat, Tao Tao, ChengXiang Zhai. 2006. Named Entity Transliteration with Comparable Corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 73–80, Sydney.

Tao Tao and ChengXiang Zhai. 2005. Mining comparable bilingual text corpora for cross-language information integration. In *KDD'05*, pages 691–696.

Tao Tao, Su-Youn Yoon, Andrew Fister, Richard Sproat, and ChengXiang Zhai. 2006. Unsupervised named entity transliteration using temporal and phonetic correlation. In *EMNLP 2006*, Sydney, July.

Paula Virga and Sanjeev Khudanpur. 2003. Transliteration of Proper Names in Cross-Lingual Information Retrieval. In *Proceedings of Workshop on Multilingual and Mixed-Language Named Entity Recognition*.

Domain-Specific Query Translation for Multilingual Information Access using Machine Translation Augmented With Dictionaries Mined from Wikipedia

Gareth J. F. Jones, Fabio Fantino, Eamonn Newman, Ying Zhang
Centre for Digital Video Processing
Dublin City University
Dublin 9, Ireland
{gjones,enewman,yzhang}@computing.dcu.ie

Abstract

Accurate high-coverage translation is a vital component of reliable cross language information access (CLIA) systems. While machine translation (MT) has been shown to be effective for CLIA tasks in previous evaluation workshops, it is not well suited to specialized tasks where domain specific translations are required. We demonstrate that effective query translation for CLIA can be achieved in the domain of cultural heritage (CH). This is performed by augmenting a standard MT system with domain-specific phrase dictionaries automatically mined from the online *Wikipedia*. Experiments using our hybrid translation system with sample query logs from users of CH websites demonstrate a large improvement in the accuracy of domain specific phrase detection and translation.

1 Introduction

Reliable translation is a key component of effective Cross Language Information Access (CLIA) systems. Various approaches to translation have been explored at evaluation workshops such as TREC[1], CLEF[2] and NTCIR[3]. Experiments at these workshops have been based on laboratory collections consisting of news articles or technical reports with "TREC" style queries with a minimum length of a full sentence. Test collection design at these workshops often ensures that there are a reasonable number of relevant documents available for each query. In such cases general purpose translation resources based on bilingual dictionaries and standard machine translation (MT) have been shown to be effective for translation in CLIA. However, this is less likely to be the case when translating the very short queries typically entered by general users of search engines, particularly when they are seeking information in a specific domain.

Online cultural heritage (CH) content is currently appearing in many countries produced by organisations such as national libraries, museums, galleries and audiovisual archives. Additionally, there are increasing amounts of CH relevant content available more generally on the World Wide Web. While some of this material concerns national or regional content only of local interest, much material relates to items involving multiple nations and languages, for example concerning events or groups encompassing large areas of Europe or Asia. In order to gain a full understanding of such things, including details contained in different collections and exploring different cultural perspectives, often requires effective multilingual search technologies.

CH content encompasses various different media, including of course text documents, but also images, videos, and audio recordings which may only be described by very limited metadata labels. Such metadata may include simple factual details such as date of creation, but also descriptive details relating to the contents of the item and interpretation and contextualization of the content. Multilingual

[1] trec.nist.gov

[2] http://www.clef-campaign.org/

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

searching using metadata content requires that either the metadata be translated into a language with which the user is able to search or that the search query be translated into the language of the metadata. This alternative of document or query translation is a well rehearsed argument in CLIA, which has generally concerned itself with full text document searching. However, the features of metadata require a more careful analysis. Metadata is typically dense in search terms, while lacking the linguistic structure and information redundancy of full text documents. The absence of linguistic structure makes precise translation of content problematic, while the lack of redundancy means that accurate translation of individual words and phrases between the query and document is vital to minimize mismatch between query and document terms. Developing reliable and robust approaches to translation for metadata search is thus an important component of search for many CH archives.

The EU FP6 *MultiMatch*[4] project is concerned with information access for multimedia and multilingual content for a range of European languages. In this paper we report on the MultiMatch query translation methods we are developing to deal with domain-specific language in the CH domain. We demonstrate the effectiveness of these techniques using example query logs from CH sites in English, Spanish and Italian. We translate the queries and examine the quality of these translations using human annotation. We show how a domain-specific phrase dictionary can be used to augment traditional general MT systems to improve the coverage and reliability of translation of these queries. We also show how retrieval performance on CH image metadata is improved with the use of these improved, domain-specific translations.

The remainder of this paper is organized as follows: Section 2 introduces the translation resources used for this study, Section 3 describes our experimental setup and results, Section 4 summarizes our conclusions, and Section 5 gives details of our ongoing work.

2 Query Translation Techniques

The MT approach to query translation for CLIA uses an existing MT system to provide automatic translation. Using MT systems for query translation is widely used in CLIA when such a system is available for the particular language pair under consideration. Results reported at the standard retrieval evaluation workshops have often shown it to be competitive with other translation methods. However, while MT systems can provide reasonable translations for general language expressions, they are often not sufficient for domain-specific phrases that contain personal names, place names, technical terms, titles of artworks, etc. In addition, certain words and phrases hold special meanings in a specific domain. For example, the Spanish phrase “Canto general” is translated into English as “general song”, which is arguably correct. However, in the CH domain, “Canto general” refers to a book title from Pablo Neruda’s book of poems and should be translated directly into English as the phrase “Canto general”. Multiple-word phrases are more information-bearing and more unambiguously represented than single words. They are often domain-specific and typically absent from static lexicons. Effective translation of such phrases is therefore particularly critical for short queries that are typically entered by non-expert users of search engines.

The focus of the research reported in this paper is a method to improve translation effectiveness of phrases previously untranslated or inappropriately translated by a standard MT system. In this work we combine an MT system with domain-specific phrase dictionaries mined from the online *Wikipedia*. The next sections describe the construction of our dictionaries and their combination with the MT system.

2.1 Phrase Dictionary Construction

Our phrase translation system uses domain-specific phrase dictionaries built by mining the online Wikipedia[5]. As a multilingual hypertext medium, Wikipedia has been shown to be a valuable new source of translation information (Adafre and de Rijke, 2005; Adafre and de Rijke, 2006; Bouma et al., 2006; Declerck et al., 2006). Wikipedia is structured as an interconnected network of articles,

[4] www.multimatch.org

[5] http://wikipedia.org

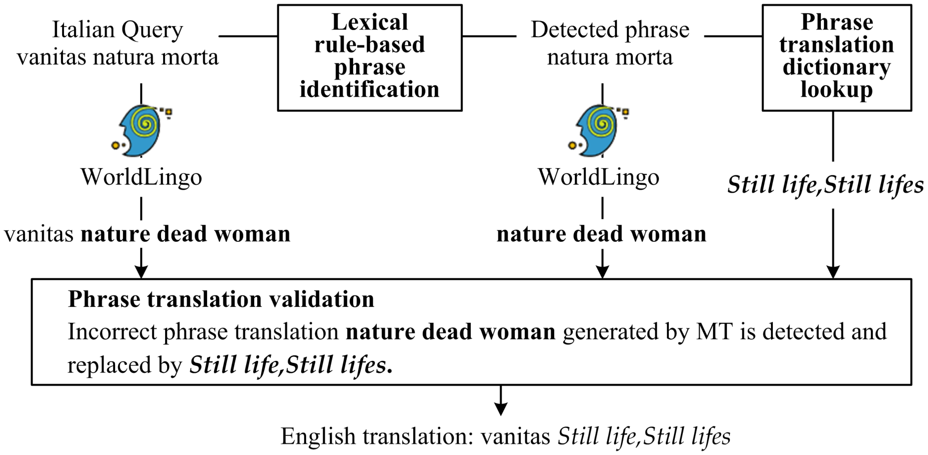

Italian Query
vanitas natura morta
Lexical rule-based phrase identification
Detected phrase
natura morta
Phrase translation dictionary lookup
WorldLingo
WorldLingo
Still life,Still lifes
vanitas nature dead woman
nature dead woman
Phrase translation validation
Incorrect phrase translation nature dead woman generated by MT is detected and replaced by Still life,Still lifes.
English translation: vanitas Still life,Still lifes

Figure 1: An example of Italian–English query translation.

in particular, wikipedia page titles in one language are often linked to a multilingual database of corresponding terms. Unlike the web, most hyperlinks in wikipedia have a more consistent pattern and meaningful interpretation. For example, the English wikipedia page `http://en.wikipedia.org/wiki/Cupid_and_Psyche` hyperlinks to its counterpart written in Italian `http://it.wikipedia.org/wiki/Amore_e_Psiche`, where the basenames of these two URLs ("Cupid and Psyche" and "Amore e Psiche") are an English–Italian translation pair. The URL basename can be considered to be a term (single word or multiple-word phrase) that should be translated as a unit.

Utilizing the multilingual linkage feature of Wikipedia, we implement a three-stage automatic process to mine wikipedia pages as a translation source and construct phrase dictionaries in the culture heritage domain.

1. First, we performed a web crawl from the English wikipedia, Category: Culture. This category contains links to articles and subcategories concerning arts, religions, traditions, entertainment, philosophy, etc. The crawl process is restricted to the category of culture including all of its recursive subcategories. In total, we collected $458,929$ English pages.

2. For each English page obtained, we extracted the hyperlinks to each of the query languages (Italian and Spanish).

3. We then selected the basenames of each pair of hyperlinks (English–Italian, English–Spanish) as translations and added them into our domain-specific dictionaries. The multiple-word phrases were added into the phrase dictionary for each language. These phrase dictionaries are later used for dictionary-based phrase identification.

The dictionaries we compiled contain about $90,000$, $70,000$, and $80,000$ distinct multiple-word phrases in English, Italian, and Spanish respectively. The majority of the phrases extracted are CH domain-specific named entities and the rest of them are general noun-based phrases, such as "Music of Ireland" and "Philosophy of history". We did not apply any classifier to filter out the general noun-based phrases, since such phrases play an equally important role in the query translation process as domain-specific named entities.

2.2 Improved MT-based Translation

Figure 1 shows our query translation process which proceeds as follows:

Lexical rule-based phrase identification Given a query, the first task is to locate phrases. Three methods of multiple-word phrase identification have been commonly used: lexical rule-based (Ballesteros and Croft, 1997; Hull and Grefenstette, 1996), statistical (Coenen et al., 2007; Gao et al., 2001), and syntactical methods (Sharma and Raman, 2003; Gelbukh et al., 2004; Van de Cruys and Villada Moirón, 2007). The lexical rule-based approach with maximum forward matching was adopted in our query translation process due to its robust performance and computational simplicity. The query is sequentially scanned to match the phrase dictionary. The longest matched subsequence is taken as a phrase and translated via a domain-specific dictionary lookup. This process is recursively invoked on the remaining part of the query until no matches are found. The performance of this approach depends strongly on the completeness of the coverage of the adopted dictionary. Our experimental results showed that at least one phrase is detected in 90% of the testing queries, for example, personal names, geographic locations, and titles of various types of artworks. This indicates that the phrase dictionaries we compiled can be used to accurately identify phrases in web queries.

WorldLingo machine translation We translate the original query into the target language using the WorldLingo[6] MT system. WorldLingo was selected for the MultiMatch project because it generally provides good translation between English, Spanish, Italian, and Dutch — the languages relevant to the Multimatch project. In addition, it provides a useful API that can be used to translate queries in real-time via HTTP transfer protocol.

Phrase translation validation For each of the phrases previously recognized, we again pass it to the MT system and the translation T_{mt} of this phrase is returned by WorldLingo. T_{mt} is then replaced in the WorldLingo translation of the query by the translations(s) T_{dict} from our domain-specific dictionary, if $T_{mt} \neq T_{dict}$. This allows us to correct unreliable phrase translations generated by the MT system.

3 Experimental Investigation

The goal of our experiments was to evaluate the usefulness and the accuracy of the domain-specific translation dictionaries. Instead of using queries from a standard information retrieval test collection, we experimented with queries explicitly seeking CH information from real query log data provided by CH organisations.

3.1 Query Log

The query log data used in this investigation was provided by three European CH organisations participating in the MultiMatch project, and is taken from their archives of real user queries. The data consists of 100 English, 1048 Italian, and 1088 Spanish distinct web queries and the number of hits of each query. The top 200 most popular multiple-word queries in Italian and Spanish were selected as the queries for testing. Due to the smaller size of the English query log, we only obtained English 53 phrasal queries.

We used two methods of evaluation: first, the dictionary usefulness and the translation effectiveness are judged extrinsically by human assessment; and second, evaluation using a parallel Italian–English metadata document set explored how translation affects the retrieval performance of an information retrieval system.

	# Detected by dictionaries	# Untranslated by WorldLingo	Proportion
EN–IT	14	11	79%
EN–ES	19	11	58%
IT–EN	83	33	40%
ES–EN	74	33	45%

Table 1: Number of detected phrases using the domain-specific dictionaries.

	Total	# Exactly correct	# + Extra translations	# + Minor noise
EN–IT	14	13	1	0
EN–ES	19	17	1	1
IT–EN	83	40	43	0
ES–EN	74	37	5	32

Table 2: Correctness of the translations of detected domain-specific phrases.

3.2 Human Judgement Evaluation

The WorldLingo MT system was used to translate Spanish and Italian queries into English and vice versa. Our domain-specific dictionaries were used to translate phrases within the queries into the same target languages. It should be noted that it is not possible to directly compare the lexical coverage of our domain-specific dictionaries and the built-in phrase dictionaries of WorldLingo since we don't have access to the internal WorldLingo dictionaries.

To evaluate the usefulness of our dictionaries, we observed the proportion of domain-specific phrases in the various query sets that can be translated using our domain-specific dictionaries mined from the web, but are incorrectly translated by WorldLingo.

[6] http://worldlingo.com

Original Query	WorldLingo Translation	Improved Machine Translation
EN–IT		
turner east sussex	Turner Sussex orientale	Turner *East Sussex*
still life flowers	fiori di vita tranquilla	fiori di *Natura morta*
francis bacon	Francis Bacon	*Francesco Bacone*
pop art	arte di schiocco	*Pop art*
m c escher	escher di m. c	*Maurits Cornelis Escher*
american 60's	americano 60's	americano *Anni 1960*
EN–ES		
vanessa bell	campana del vanessa	*Vanessa Bell*
turner east sussex	Turner sussex del este	Turner *East Sussex*
henry moore	moore del Henrio	*Henry Moore*
still life flowers	flores de la vida inmóvil	flores de *Bodegón*
guerrilla girls	muchachas del guerrilla	*Guerrilla Girls*
IT–EN		
leonardo da vinci	leonardo from you win	*Da Vinci, Leonardo da Vinci,* *Leonardo daVinci, Leonardo de Vinci*
duomo di milano	dome of Milan	*Cathedral of Milan, Duomo di Milan,* *Duomo di Milano, Duomo of Milan, Milan Cathedral*
beni culturali	cultural assets	*Cultural heritage*
arte povera	poor art	*Arte povera*
san lorenzo	saint lorenzo	*Lawrence of Rome, Saint Lawrence, St Lawrence,*
gentile da fabriano	kind from fabriano	*Gentile da Fabriano*
statua della liberta	statue of the freedom	*Statue of Liberty*
aldo rossi	aldo red	*Aldo Rossi*
arnaldo pomodoro	arnaldo tomato	*Arnaldo Pomodoro*
la cattura di cristo di caravaggio	the capture of caravaggio Christ	*The Taking of Christ* caravaggio
ES–EN		
lope de vega	lope of fertile valley	*Lope de Vega*
literatura infantil	infantile Literature	*Children's book, Children's books,Children's literature*
cantar de mio cid	to sing of mine cid	*Cantar de mio Cid, Lay of the Cid, The Lay of the Cid*
el quijote de la mancha	quijote of the spot	quijote of *La Mancha*
dulce maria loynaz	candy Maria loynaz	*Dulce María Loynaz*
andres bello	andres beautiful	*Andrés Bello*
filosofia del derecho	philosophy of the right	*Philosophy of law*
elogio de la locura	praise of madness	*In Praise of Folly, Praise of Folly, The Praise of Folly*
la regenta	it runs it	*La Regenta*
cristobal colon	cristobal colon	*Christopher Colombus, Christopher Columbus,* *Cristopher Columbus*

Table 3: Some examples of improved translations using the domain-specific dictionaries. (The corrected phrase translations are in italic.)

Namely, we tested the ability of our system to detect and correct the presence of unreliable MT translations for domain-specific phrases. Translated phrases for these queries can generally be judged unambiguously as correct or incorrect by a bilingual speaker of the languages involved, and so we are confident that assessment of translation accuracy here does not involve significant degrees of subjectivity.

As shown in Table 1, we can see that 79%, 58%, 40%, and 45% of incorrect MT-translated phrases were able to be corrected using the domain-specific dictionaries mined from wikipedia, in EN–IT, EN–ES, IT–EN, and ES–EN translation tasks, respectively. Our system leads to a large improvement in MT translation for domain-specific phrases. Some examples of improved query translations are shown in Table 3.

We also conducted an investigation on the correctness of the translation mined from wikipedia, as shown in Table 2. *Exact correct translation* is strictly-correct single translation. *Extra translation* refers to strictly-correct multiple translations, for example, "Cathedral of Milan, Duomo di Milan, Duomo di Milano, Duomo of Milan, Milan Cathedral" (Italian: Duomo di Milano). It is interesting to observe that about 50% of Italian phrases are found to have multiple correct English translations due to multiple English wikipedia pages being redirected to the same Italian pages. Some *minor noise* is observed when the correct translation contains some related additional words, such as "Alfonso XII of Spain" (Spanish: Alfonso XII). When used for information retrieval, this additional information can sometimes improve effectiveness.

We are not able to manually evaluate the accuracy of all translation pairs in our bilingual dictionaries due to limited resources. However, our results for sample queries from user logs demonstrate that our translations are generally highly accurate.

3.3 Intrinsic Evaluation Using IR System

Our information retrieval experiments were performed on a database of metadata associated with a collection of 5000 CH photographs. The metadata to describe each artifact in the collection is available in English and in Italian. Each photograph is described identically in both languages. We formed a separate search index for English and Italian. Search was carried out using the Lucene search engine[7]. We carried out an evaluation based on this collection which proceeded as follows:

1. Submit the original queries to the index and record the ranked list of references returned.

2. Submit the translated queries to the appropriate index and record the ranked list of references returned.

3. Find the correlation between the lists returned for the native language queries and the queries translated to that language.

4. The better translation will have the stronger correlation with the native language list.

Due to the fact that the corpus was only complete in the Italian and English versions, we were unable to include the Spanish queries in this part of the evaluation. Also, while this collection is based in the CH domain, some of the queries yield no relevant documents due to their specialist nature. The collection of queries for which meaningful retrieval results are returned is too small to allow for a quantitative analysis of retrieval effectiveness. Therefore, we present a qualitative analysis of some of the more interesting cases.

3.3.1 Italian–English translations

The Italian queries cover a wide range of Italian interests in CH. We present here a sample of some of the more interesting results.

Arnaldo Pomodoro This refers to an Italian artist, but the name "Pomodoro" is translated to "Tomato" in English by WorldLingo. While there were no references to the artist in the collection, all documents returned contained the term "tomato" (referring to the vegetable) which are irrelevant to the query. The dictionary-based translation recognized the name and therefore left it untranslated. It is preferable to retrieve no documents rather than to retrieve irrelevant ones.

Amore e Psiche This refers to the sculpture entitled "Cupid and Psyche" in English. This phrase was matched in our phrase dictionary and translated correctly. The MT system translated this as "Love, Psyche". The dictionary translation was observed to retrieve relevant documents with greater precision since it matched against the more specific term "Cupid", as opposed to the more general term "Love".

David Michaelangelo This query provided a counterexample. The phrase dictionary added the term "statue" to the translated query. This led to retrieval of a large number of non-relevant documents.

3.3.2 English–Italian translations

As with the Italian queries, there was not much overlap between the query log and the document collection. Some of the interesting translations include:

pop art This phrase was recognized by our domain-specific dictionary, and so was left in its original form for searching in Italian. Interestingly, this led to an improvement in search accuracy for the query compared to that in the English language collection. For the English index, this phrase matched many non-relevant documents which contained the word "art". However, when searching in the Italian index, where "art" is not a word encountered in the

[7] http://lucene.apache.org/

general vocabulary, the phrase retrieves only 7 documents, of which 5 were relevant.

Turner East Sussex The place name "East Sussex" was correctly recognized and translated by our phrase dictionary. However the MT system again failed to recognise it and translated the partial term "East" to "Orientale". The presence of the term "Orientale" in the translated query resulted in many non-relevant documents being retrieved, reducing the precision of the query.

The examples given in this section provide anecdotal evidence to support the view that the automatically mined domain-specific phrase dictionary improves the performance of the retrieval system. Query sets and relevance judgements are being created for the MultiMatch document set by domain experts who compiled the original collections. Thus we will be able to ensure that the query sets are a good representative sample of the information needs of the typical user. These test collections will allow us to conduct full quantitative analysis of our system.

4 Conclusions

We have presented an automatic mining system developed for construction of domain-specific phrase dictionaries. Phrases not translated by a general MT system are shown to be translated effectively using these dictionaries. The extracted translations were evaluated by human assessment and shown to be highly accurate. We have also demonstrated a way to combine these dictionaries with MT for topical phrases in the culture heritage domain. Our experimental results show that we were able to detect and correct a large proportion of domain-specific phrases unsuccessfully translated by MT, and thus improve information retrieval effectiveness and facilitate MLIA.

5 Ongoing Work

In our ongoing work we plan to further extend the coverage of our dictionaries by exploring the mining of other translations pairs from within the linked *Wikipedia* pages. While the method described in this paper has been shown to be effective for query translation, we have so far only demonstrated its behavior for a very small number of queries to our CLIA system. We are currently developing test collections based on several CH data sets to evaluate the effectiveness of our hybrid query translation method.

Acknowledgement

Work partially supported by European Community under the Information Society Technologies (IST) programme of the 6th FP for RTD — project MultiMATCH contract IST−033104. The authors are solely responsible for the content of this paper. It does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of data appearing therein.

References

Sisay Fissaha Adafre and Maarten de Rijke. 2005. Discovering missing links in Wikipedia. In *Proceedings of the 3rd International Workshop on Link Discovery*, pages 90–97, Chicago, Illinois, United States. ACM Press.

Sisay Fissaha Adafre and Maarten de Rijke. 2006. Finding similar sentences across multiple languages in Wikipedia. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 62–69, Trento, Italy.

Lisa Ballesteros and W. Bruce Croft. 1997. Phrasal translation and query expansion techniques for cross-language information retrieval. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 84–91, Philadelphia, PA, USA. ACM Press.

Gosse Bouma, Ismail Fahmi, Jori Mur, Gertjan van Noord, Lonneke van der Plas, and Jorg Tiedemann. 2006. The University of Groningen at QA@CLEF 2006 using syntactic knowledge for QA. In *Working Notes for the Cross Language Evaluation Forum 2006 Workshop*, Alicante, Spain.

Frans Coenen, Paul H. Leng, Robert Sanderson, and Yanbo J. Wang. 2007. Statistical identification of key phrases for text classification. In *Machine Learning and Data Mining in Pattern Recognition*, volume 4571 of *Lecture Notes in Computer Science*, pages 838–853. Springer.

Thierry Declerck, Asunciòn Gòmez Pèrez, Ovidiu Vela, Zeno Gantner, and David Manzano-Macho. 2006. Multilingual lexical semantic resources for ontology translation. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, Genoa, Italy. ELDA.

Jianfeng Gao, Jian-Yun Nie, Endong Xun, Jian Zhang, Ming Zhou, and Changning Huang. 2001. Improving query translation for cross-language information retrieval using statistical models. In *Proceedings of the 24th Annual International ACM SIGIR conference on Research and Development in information retrieval*, pages 96–104, New Orleans, Louisiana, United States. ACM Press.

Alexander F. Gelbukh, Grigori Sidorov, Sang-Yong Han, and Erika Hernández-Rubio. 2004. Automatic syntactic analysis for detection of word combinations. In *Proceedings of the 5th International Conference on Computational Linguistics and Intelligent Text Processing*, volume 2945 of *Lecture Notes in Computer Science*, pages 243–247. Springer.

David A. Hull and Gregory Grefenstette. 1996. Querying across languages: a dictionary-based approach to multilingual information retrieval. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49–57, Zurich, Switzerland. ACM Press.

Rupali Sharma and S. Raman. 2003. Phrase-based text representation for managing the web documents. In *Proceedings of the International Conference on Information Technology: Computers and Communications*, page 165, Washington, DC, USA. IEEE Computer Society.

Tim Van de Cruys and Begoña Villada Moirón. 2007. Semantics-based multiword expression extraction. In *Proceedings of the Workshop on A Broader Perspective on Multiword Expressions*, pages 25–32, Prague, Czech Republic, June. Association for Computational Linguistics.

Statistical Transliteration for Cross Langauge Information Retrieval using HMM alignment and CRF

Surya Ganesh, Sree Harsha
LTRC, IIIT
Hyderabad, India
suryag,sreeharshay@students.iiit.net

Prasad Pingali, Vasudeva Varma
LTRC, IIIT
Hyderabad, India
pvvpr,vv@iiit.net

Abstract

In this paper we present a statistical transliteration technique that is language independent. This technique uses Hidden Markov Model (HMM) alignment and Conditional Random Fields (CRF), a discriminative model. HMM alignment maximizes the probability of the observed (source, target) word pairs using the expectation maximization algorithm and then the character level alignments (n-gram) are set to maximum posterior predictions of the model. CRF has efficient training and decoding processes which is conditioned on both source and target languages and produces globally optimal solutions. We apply this technique for Hindi-English transliteration task. The results show that our technique perfoms better than the existing transliteration system which uses HMM alignment and conditional probabilities derived from counting the alignments.

1 Introduction

In cross language information retrieval (CLIR) a user issues a query in one language to search a document collection in a different language. Out of Vocabulary (OOV) words are problematic in CLIR. These words are a common source of errors in CLIR. Most of the query terms are OOV words like named entities, numbers, acronyms and technical terms. These words are seldom found in Bilingual dictionaries used for translation. These words can be the most important words in the query. These words need to be transcribed into document language when query and document languages do not share common alphabet. The practice of transcribing a word or text written in one language into another language is called transliteration.

A source language word can have more than one valid transliteration in target language. For example for the Hindi word below four different transliterations are possible .

गौतम् - gautam, gautham, gowtam, gowtham

Therefore, in a CLIR context, it becomes important to generate all possible transliterations to retrieve documents containing any of the given forms.

Most current transliteration systems use a generative model for transliteration such as freely available GIZA++[1] (Och and Ney , 2000),an implementation of the IBM alignment models (Brown et al., 1993). These systems use GIZA++ (which uses HMM alignment) to get character level alignments (n-gram) from word aligned data. The transliteration system was built by counting up the alignments and converting the counts to conditional probabilities. The readers are strongly encouraged to refer to (Nasreen and Larkey , 2003) to have a detailed understanding of this technique.

In this paper, we present a simple statistical technique for transliteration. This technique uses HMM alignment and Conditional Random Fields (Hanna , 2004) a discriminative model. Based on this technique desired number of transliterations are generated for a given source language word. We also describe the Hindi-English transliteration system built by us. However there is nothing particular to both these languages in the system. We evaluate the transliteration system on a test set of proper names from Hindi-English parallel transliterated word lists. We compare the efficiency of this system with the system that was developed using HMMs (Hidden Markov Models) only.

[1] http://www.fjoch.com/GIZA++.html

2 Previous work

Earlier work in the field of Hindi CLIR was done by Jaleel and Larkey (Larkey et al., 2003). They did this based on their work in English-Arabic transliteration for cross language Information retrieval (Nasreen and Larkey , 2003). Their approach was based on HMM using GIZA++ (Och and Ney , 2000). Prior work in Arabic-English transliteration for machine translation purpose was done by Arababi (Arbabi et al., 1994). They developed a hybrid neural network and knowledge-based system to generate multiple English spellings for Arabic person names. Knight and Graehl (Knight and Graehl , 1997) developed a five stage statistical model to do back transliteration, that is, recover the original English name from its transliteration into Japanese Katakana. Stalls and Knight (Stalls and Knight , 1998) adapted this approach for back transliteration from Arabic to English of English names. Al-Onaizan and Knight (Onaizan and Knight , 2002) have produced a simpler Arabic/English transliterator and evaluates how well their system can match a source spelling. Their work includes an evaluation of the transliterations in terms of their reasonableness according to human judges. None of these studies measures their performance on a retrieval task or on other NLP tasks. Fujii and Ishikawa (Fujii and Ishikawa , 2001) describe a transliteration system for English-Japanese cross language IR that requires some linguistic knowledge. They evaluate the effectiveness of their system on an English-Japanese cross language IR task.

3 Problem Description

The problem can be stated formally as a sequence labelling problem from one language alphabet to other. Consider a source language word $x_1 x_2 .. x_i .. x_N$ where each x_i is treated as a word in the observation sequence. Let the equivalent target language orthography of the same word be $y_1 y_2 .. y_i .. y_N$ where each y_i is treated as a label in the label sequence. The task here is to generate a valid target language word (label suquence) for the source language word (observation sequence).

$$\begin{array}{ccc} x_1 & \text{———————} & y_1 \\ x_2 & \text{———————} & y_2 \\ . & \text{———————-} & . \\ . & \text{———————-} & . \\ . & \text{———————-} & . \\ x_N & \text{———————} & y_N \end{array}$$

Here the valid target language alphabet(y_i) for a source language alphabet(x_i) in the input source language word may depend on various factors like

1. The source language alphabet in the input word.
2. The context(alphabets) surrounding source language alphabet(x_i) in the input word.
3. The context(alphabets) surrounding target language alphabet(y_i) in the desired output word.

4 Transliteration using HMM alignment and CRF

Our approach for transliteration is divided into two phases. The first phase induces character alignments over a word-aligned bilingual corpus, and the second phase uses some statistics over the alignments to transliterate the source language word and generate the desired number of target language words.

The selected statistical model for transliteration is based on HMM alignment and CRF. HMM alignment maximizes the probability of the observed (source, target) word pairs using the expectation maximization algorithm. After the maximization process is complete, the character level alignments (n-gram) are set to maximum posterior predictions of the model. This alignment is used to get character level alignment (n-gram) of source and target language words. From the character level alignment obtained we compare each source language character (n-gram) to a word and its corresponding target language character (n-gram) to a label. Conditional random fields (CRFs) are a probabilistic framework for labeling and segmenting sequential data. We use CRFs to generate target language word (similar to label sequence) from source language word (similar to observation sequence).

CRFs are undirected graphical models which define a conditional distribution over a label sequence given an observation sequence. We define CRFs as conditional probability distributions $P(Y|X)$ of target language words given source language words. The probability of a particular target language word Y given source language word X is the normalized product of potential functions each of the form

$$e^{(\sum_j \lambda_j t_j(Y_{i-1},Y_i,X,i))+(\sum_k \mu_k s_k(Y_i,X,i))}$$

where $t_j(Y_{i-1},Y_i,X,i)$ is a transition feature function of the entire source language word and the target language characters (n-gram) at positions i and $i-1$ in the target language word; $s_k(Y_i,X,i)$ is a state feature function of the target language word at position i and the source language word; and λ_j and μ_k are parameters to be estimated from training data.

$$F_j(Y,X) = \sum_{i=1}^{n} f_j(Y_{i-1},Y_i,X,i)$$

where each $f_j(Y_{i-1},Y_i,X,i)$ is either a state function $s(Y_{i-1},Y_i,X,i)$ or a transition function $t(Y_{i-1},Y_i,X,i)$. This allows the probability of a target language word Y given a source language word X to be written as

$$P(Y|X,\lambda) = (\frac{1}{Z(X)})e^{(\sum \lambda_j F_j(Y,X))}$$

$Z(X)$ is a normalization factor.

The parameters of the CRF are usually estimated from a fully observed training data $\{(x^{(k)},y^{(k)})\}$. The product of the above equation over all training words, as a function of the parameters λ, is known as the likelihood, denoted by $p(\{y^{(k)}\}|\{x^{(k)}\},\lambda)$. Maximum likelihood training chooses parameter values such that the logarithm of the likelihood, known as the log-likelihood, is maximized. For a CRF, the log-likelihood is given by

$$L(\lambda) = \sum_k [log \frac{1}{Z(x^{(k)})} + \sum_j \lambda_j F_j(y^{(k)},x^{(k)})]$$

This function is concave, guaranteeing convergence to the global maximum. Maximum likelihood parameters must be identified using an iterative technique such as iterative scaling (Berger , 1997) (Darroch and Ratcliff, 1972) or gradient-based methods (Wallach , 2002). Finally after training the model using CRF we generate desired number of transliterations for a given source language word.

5 Hindi - English Transliteration system

The whole model has three important phases. Two of them are off-line processes and the other is a run time process. The two off-line phases are preprocessing the parallel corpora and training the model using CRF++[2]. CRF++ is a simple, customizable, and open source implementation of Conditional Random Fields (CRFs) for segmenting/labeling sequential data. The on-line phase involves generating desired number of transliterations for the given Hindi word (UTF-8 encoded).

5.1 Preprocessing

The training file is converted into a format required by CRF++. The sequence of steps in preprocessing are

1. Both Hindi and English words were prefixed with a begin symbol B and suffixed with an end symbol E which correspond to start and end states. English words were converted to lower case.

2. The training words were segmented in to unigrams and the English-Hindi word pairs were aligned using GIZA++, with English as the source language and Hindi as target language.

3. The instances in which GIZA++ aligned a sequence of English characters to a single Hindi unicode character were counted. The 50 most frequent of these character sequences were added to English symbol inventory. There were hardly any instances in which a sequence of Hindi unicode characters were aligned to a single English character. So, in our model we consider Hindi unicode characters, $NULL$, English unigrams and English n-grams.

4. The English training words were re segmented based on the new symbol inventory, i.e., if

[2] http://crfpp.sourceforge.net/

a character was a part of an n-gram, it was grouped with the other characters in the n-gram. If not, it was rendered separately. GIZA++ was used to align the above Hindi and English training word pairs, with Hindi as source language and English as target language.

These four steps are performed to get the character level alignment (n-grams) for each source and target language training words.

5. The alignment file from the GIZA++ output is used to generate training file as required by CRF++ to work. In the training file a Hindi unicode character aligned to a English uni-gram or n-gram is called a token. Each token must be represented in one line, with the columns separated by white space (spaces or tabular characters).Each token should have equal number of columns.

5.2 Training Phase

The preprocessing phase converts the corpus into CRF++ input file format. This file is used to train the CRF model. The training requires a template file which specifies the features to be selected by the model. The training is done using Limited memory Broyden-Fletcher-Goldfarb-Shannon method(LBFGS) (Liu and Nocedal, 1989) which uses quasi-newton algorithm for large scale numerical optimization problem. We used Hindi unicode characters as features for our model and a window size of 5.

5.3 Transliteration

The list of Hindi words that need to be transliterated is taken. These words are converted into CRF++ test file format and transliterated using the trained model which gives the top n probable English words. CRF++ uses forward Viterbi and backward A* search whose combination produce the exact n-best results.

6 Evaluation

We evaluate the two transliteration systems for Hindi - English that use HMM alignment and CRF with the system that uses HMM only in two ways. In first evaluation method we compare transliteration accuracies of the two systems using in-corpus (training data) and out of corpus words. In second method we compare CLIR performance of the two systems using Cross Language Evaluation Forum (CLEF) 2007 ad-hoc bilingual track (Hindi-English) documents in English language and 50 topics in Hindi Language. The evaluation document set consists of news articles and reports from Los Angeles Times of 2002. A set of 50 topics representing the information need were given in Hindi. A set of human relevance judgements for these topics were generated by assessors at CLEF. These relevance judgements are binary relevance judgements and are decided by a human assessor after reviewing a set of pooled documents using the relevant document pooling technique. The system evaluation framework is similar to the Craneld style system evaluations and the measures are similar to those used in TREC[3].

6.1 Transliteration accuracy

We trained the model on 30,000 words containing Indian city names, Indian family names, Male first names and last names, Female first names and last names. We compare this model with the HMM model trained on same training data. We tested both the models using in-corpus (training data) and out of corpus words. The out of corpus words consist of both Indian and foreign place names, person names. We evaluate both the models by considering top 5, 10, 15 and 20 transliterations. Accuracy was calculated using the following equation below

$$Accuracy = \frac{C}{N} * 100$$

C - Number of test words with the correct transliteration appeared in the desired number (5, 10, 15, 20, 25) of transliterations.
N - Total number of test words.

The results for 30,000 in-corpus words and 1,000 out of corpus words are shown in the table 1 and table 2 respectively. In below tables 1 & 2 HMM model refers to the system developed using HMM alignment and conditional probabilities derived from counting the alignments, HMM & CRF model refers to the system developed using HMM

[3] Text Retrieval Conferences, http://trec.nist.gov

Model	Top 5	Top 10	Top 15	Top 20	Top 25
HMM	74.2	78.7	81.1	82.1	83.0
HMM & CRF	76.5	83.6	86.5	88.9	89.7

Table 1: Transliteration accuracy of the two systems for in-corpus words.

Model	Top 5	Top 10	Top 15	Top 20	Top 25
HMM	69.3	74.3	77.8	80.5	81.3
HMM & CRF	72.1	79.9	83.5	85.6	86.5

Table 2: Transliteration accuracy of the two systems for out of corpus words.

alignment and CRF for generating top n transliterations.

CRF models for Named entity recognition, POS tagging etc. have efficiency in high nineties when tested on training data. Here the efficiency (Table 1) is low due to the use of HMM alignment in GIZA++.

We observe that there is a good improvement in the efficiency of the system with the increase in the number of transliterations up to some extent(20) and after that there is no significant improvement in the efficiency with the increase in the number of transliterations.

During testing, the efficiency was calculated by considering only one of the correct transliterations possible for a given Hindi word. If we consider all the correct transliterations the efficiency will be much more.

The results clearly show that CRF model performs better than HMM model for Hindi to English transliteration.

6.2 CLIR Evaluation

In this section we evaluate the transliterations produced by the two systems in CLIR task, the task for which these transliteration systems were developed. We tested the systems on the CLEF 2007 documents and 50 topics. The topics which contain named entities are few in number; there were around 15 topics with them. These topics were used for evaluation of both the systems.

We developed a basic CLIR system which performs the following steps

1. Tokenizes the Hindi query and removes stop words.
2. Performs query translation; each Hindi word is looked up in a Hindi - English dictionary and all the English meanings for the Hindi word were added to the translated query and for the words which were not found in the dictionary, top 20 transliterations generated by one of the systems are added to the query.
3. Retrieves relevant documents by giving translated query to CLEF documents.

We present standard IR evaluation metrics such as precision, mean average precision(MAP) etc.. in the table 3 below for the two systems.

The above results show a small improvement in different IR metrics for the system developed using HMM alignment and CRF when compared to the other system. The difference in metrics between the systems is low because the number of topics tested and the number of named entities in the tested topics is low.

7 Future Work

The selected statistical model for transliteration is based on HMM alignment and CRF. This alignment model is used to get character level alignment (n-gram) of source and target language words. The alignment model uses IBM models, such as Model 4, that resort to heuristic search techniques to approximate forward-backward and Viterbi inference, which sacrifice optimality for tractability. So, we plan to use discriminative model CRF for character level alignment (Phil and Trevor , 2006) of source and target language words. The behaviour of the other discrminative models such as Maximum Entropy models etc., towards the transliteration task

Model	P10	tot_rel	tot_rel_ret	MAP	bpref
HMM	0.3308	13000	3493	0.1347	0.2687
HMM & CRF	0.4154	13000	3687	0.1499	0.2836

Table 3: IR Evaluation of the two systems.

also needs to be verified.

8 Conclusion

We demonstrated a statistical transliteration system using HMM alignment and CRF for CLIR that works better than using HMMs alone. The following are our important observations.

1. With the increase in number of output target language words for a given source language word the efficiency of the system increases.

2. The difference between efficiencies for top n and n-5 where $n > 5$; is decreasing on increasing the n value.

References

A. L. Berger. 1997. *The improved iterative scaling algorithm: A gentle introduction.*

Al-Onaizan Y, Knight K. 2002. *Machine translation of names in Arabic text. Proceedings of the ACL conference workshop on computational approaches to Semitic languages.*

Arababi Mansur, Scott M. Fischthal, Vincent C. Cheng, and Elizabeth Bar. 1994. *Algorithms for Arabic name transliteration. IBM Journal of research and Development.*

D. C. Liu and J. Nocedal. 1989. *On the limited memory BFGS method for large-scale optimization, Math. Programming 45 (1989), pp. 503–528.*

Fujii Atsushi and Tetsuya Ishikawa. 2001. *Japanese/English Cross-Language Information Retrieval: Exploration of Query Translation and Transliteration. Computers and the Humanities, Vol.35, No.4, pp.389-420.*

H. M. Wallach. 2002. *Efficient training of conditional random fields. Masters thesis, University of Edinburgh.*

Hanna M. Wallach. 2004. *Conditional Random Fields: An Introduction.*

J. Darroch and D. Ratcliff. 1972. *Generalized iterative scaling for log-linear models. The Annals of Mathematical Statistics, 43:14701480.*

Knight Kevin and Graehl Jonathan. 1997. *Machine transliteration. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, pp. 128-135. Morgan Kaufmann.*

Larkey, Connell,AbdulJaleel. 2003. *Hindi CLIR in Thirty Days.*

Nasreen Abdul Jaleel and Leah S. Larkey. 2003. *Statistical Transliteration for English-Arabic Cross Language Information Retrieval.*

Och Franz Josef and Hermann Ney. 2000. *Improved Statistical Alignment Models. Proc. of the 38th Annual Meeting of the Association for Computational Linguistics, pp. 440-447, Hong Kong, China.*

P. F. Brown, S. A. Della Pietra, and R. L. Mercer. 1993. *The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics, 19(2):263-311.*

Phil Blunsom and Trevor Cohn. 2006. *Discriminative Word Alignment with Conditional Random Fields.*

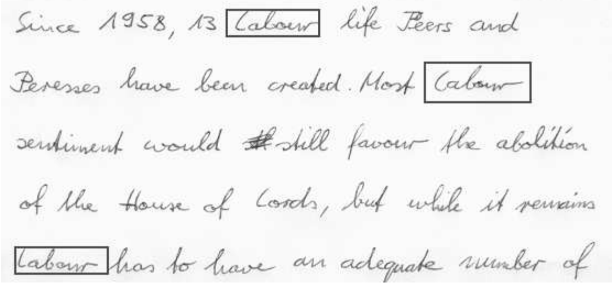
Stalls Bonnie Glover and Kevin Knight. 1998. *Translating names and technical terms in Arabic text.*

Script Independent Word Spotting in Multilingual Documents

Anurag Bhardwaj, Damien Jose and Venu Govindaraju
Center for Unified Biometrics and Sensors (CUBS)
University at Buffalo, State University of New York
Amherst, New York 14228
{ab94,dsjose,govind}@cedar.buffalo.edu

Abstract

This paper describes a method for script independent word spotting in multilingual handwritten and machine printed documents. The system accepts a query in the form of text from the user and returns a ranked list of word images from document image corpus based on similarity with the query word. The system is divided into two main components. The first component known as Indexer, performs indexing of all word images present in the document image corpus. This is achieved by extracting Moment Based features from word images and storing them as index. A template is generated for keyword spotting which stores the mapping of a keyword string to its corresponding word image which is used for generating query feature vector. The second component, Similarity Matcher, returns a ranked list of word images which are most similar to the query based on a cosine similarity metric. A manual Relevance feedback is applied based on Rocchio's formula, which re-formulates the query vector to return an improved ranked listing of word images. The performance of the system is seen to be superior on printed text than on handwritten text. Experiments are reported on documents of three different languages: English, Hindi and Sanskrit. For handwritten English, an average precision of 67% *was obtained for* 30 *query words. For machine printed Hindi, an average precision of* 71% *was obtained for* 75 *query words and for Sanskrit, an average precision of* 87% *with* 100 *queries was obtained.*



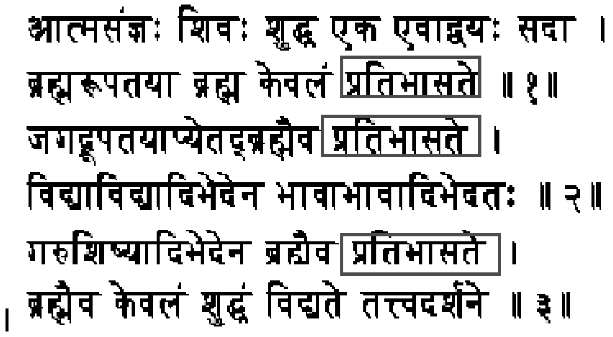
Since 1958, 13 Labour life Peers and
Peresses have been created. Most Labour
sentiment would still favour the abolition
of the House of Lords, but while it remains
Labour has to have an adequate number of

Figure 1: A Sample English Document - Spotted Query word shown in the bounding box.

1 Introduction

The vast amount of information available in the form of handwritten and printed text in different languages poses a great challenge to the task of effective information extraction. Research in this area has primarily focussed on OCR based solutions which are adequate for Roman Language (A sample English document is shown in Figure 1). However efficient solutions do not exist for scripts like Devanagari. One of the main reasons for this is lack of generalisation. OCR solutions tend to be specific to script type. Ongoing research continues to scale these methods to different types and font sizes. Furthermore, non-Latin scripts exhibit complex character classes (like in the Sanskrit document shown in Figure 2) and poor quality documents are common.

The notion of Word spotting [6] has been introduced as an alternative to OCR based solutions. It can be defined as an information retrieval task that finds all occurences of a typed query word in a set of handwritten



आत्मसंज्ञः शिवः शुद्ध एक एवाद्वयः सदा ।
ब्रह्मरूपतया ब्रह्म केवलं प्रतिभासते ॥ १॥
जगद्रूपतयाप्येतद्ब्रह्मैव प्रतिभासते ।
विद्याविद्यादिभेदेन भावाभावादिभेदतः ॥ २॥
गुरुशिष्यादिभेदेन ब्रह्मैव प्रतिभासते ।
। ब्रह्मैव केवलं शुद्धं विद्यते तत्त्वदर्शने ॥ ३॥

Figure 2: A Sample Sanskrit Document - Spotted Query word shown in the bounding box.

or machine printed documents. While spotting words in English has been explored [3, 5, 4, 7, 11], generalising these approaches to multiple scripts is still an ongoing research task. Harish et.al [1] describe a 'Gradient, Structural, Concavity' (GSC) based feature set for word spotting in multiple scripts. However, they do not report the average precision rate for all queries in their experimental results which makes it difficult to estimate the performance of their methodology.

One important factor in finding a script independent solution to word spotting is use of image based features which are invariant to script type, image scale and translations. This paper proposes the use of moment based features for spotting word images in different scripts. We describe a moment-function based feature extraction scheme and use the standard vector space model to represent the word images. Similarity between the query feature vector and the indexed feature set is computed using a cosine similarity metric. We also apply the Rocchio formula based Manual Relevance feedback to improve the ranking of results obtained. We evaluate the performance of our system by conducting experiments on document images of three different scripts: English, Hindi and Sanskrit.

The organization of the rest of the paper is as follows: Section 2 describes the previous work. Section 3 describes the theory of moment functions. Section 4 describes indexing word images and feature extraction. Section 5 describes the Similarity Matching and Relevance Feedback method applied to re-rank results. Section 6 describes the experiments and results. Future work and conclusions are outlined in Section 7.

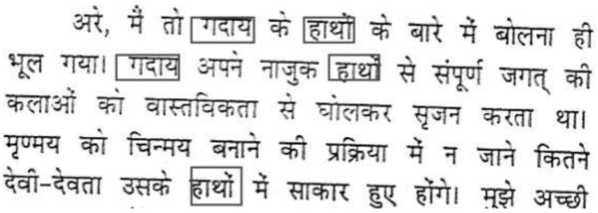
2 Previous Work

Spotting words in English has recently received considerable attention. Manmatha et al. [7], have proposed a combination of feature sets well suited for this application. For finding similarity between a query word image and the document word image, Dynamic Time warping [8] is commonly used. Although the approach has been promising with English handwritten documents, it does not generalise well across scripts. For eg., presence of Shirorekha in Devanagari script (an example shown in Figure 3) renders most of the profile based features ineffective. Also, DTW based approaches are slow. Approaches which use a filter based feature set [2], are efficient with uniform font size and type but are not able to handle font variations and translations.

Harish et al. [1] use a Gradient, Structural and Concavity (GSC) feature set which measures the image characteristics at local, intermediate and large scales. Features are extracted using a $4x8$ sampling window to gather information locally. Since character segmentation points are not perfectly located, local information about stroke orientation and image gradient is not sufficient to characterize the change in font scale and type. Moreover, presence of noise in small regions of the word image lead to inconsistency in the overall feature extraction process. The performance of their approach is presented in terms of percentage of the number of times the correct match was returned, which does not capture the recall rate of system. For English word spotting, their results do not state the size of the dataset and precision recall values have been reported for only 4 query words. For Sanskrit word spotting, the total number of query words is not mentioned which makes understanding of precision recall curve difficult. A comparison of their results against our proposed method is presented in section 6.

3 Moment Functions

Moments and functions of moments have been previously used to achieve an invariant representation of a two-dimensional image pattern [9]. Geometrical moments



अरे, मैं तो गदाय के हाथों के बारे में बोलना ही
भूल गया। गदाय अपने नाजुक हाथों से संपूर्ण जगत् की
कलाओं को वास्तविकता से घोलकर सृजन करता था।
मृण्मय को चिन्मय बनाने की प्रक्रिया में न जाने कितने
देवी-देवता उसके हाथों में साकार हुए होंगे। मुझे अच्छी

Figure 3: A Sample Hindi Document - Spotted Query words shown in the bounding box.

[9] have the desirable property of being invariant under the image translation, scale and stretching and squeezing in either X or Y direction. Mathematically, such affine transformations are of the form of $X^* = aX + b$, and $Y^* = cY + d$ [10]. Geometrical Moments (GM) of order $(p+q)$ for a continuous image function $f(x, y)$ are defined as :

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) \, dx \, dy \tag{1}$$

where $p, q = 0, 1, 2, ..., \infty$. The above definition has the form of the projection of the function $f(x, y)$ onto the mononomial $x^p y^q$. In our case, where the function $f(x, y)$ has only two possible values of 0 and 1, the equation 1 reduces to :

$$M_{pq} = \sum_X \sum_Y x^p y^q f(x, y) \tag{2}$$

where X and Y represent x, y coordinates of the image. The center of gravity of the image has the coordinates :

$$\bar{x} = \frac{M_{10}}{M_{00}}, \bar{y} = \frac{M_{01}}{M_{00}}, \tag{3}$$

If we refer to the center of gravity as origin, we obtain :

$$\bar{M}_{pq} = \sum_X \sum_Y (x - \bar{x})^p (y - \bar{y})^q f(x, y) \tag{4}$$

These moments are also referred to as Central Moments and can be expressed as a linear combination of M_{jk} and the moments of lower order. The variances of the moment are defined as :

$$\sigma_x = \sqrt{\frac{\bar{M}_{20}}{M_{00}}}, \sigma_y = \sqrt{\frac{\bar{M}_{02}}{M_{00}}}, \tag{5}$$

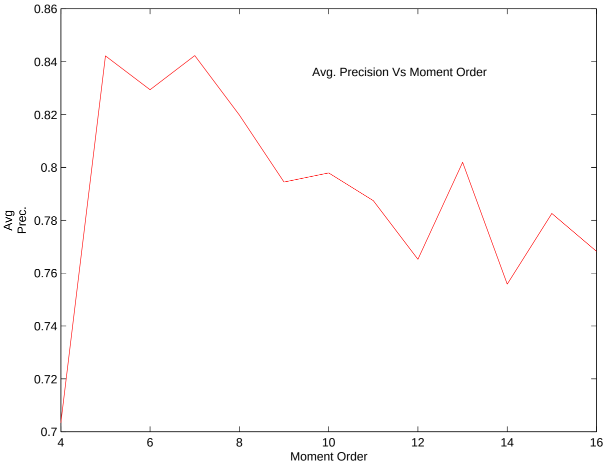


Figure 4: Average Precision curve Vs Moment Order for a Hindi Image Subset.

They are used to normalise the coordinates by setting:

$$x^* = \frac{(x - \bar{x})}{\sigma_x}, y^* = \frac{(y - \bar{y})}{\sigma_y}, \tag{6}$$

Using the normalised values of coordinates as obtained in equation 6 , the moment equation is as follows :

$$m_{pq} = \frac{\sum_X \sum_Y (x^*)^p (y^*)^q f(x, y)}{M_{00}} \tag{7}$$

which is invariant under image translation and scale transformations.

4 Feature Extraction and Indexing

Feature extraction is preceeded by preprocessing of documents prior to computing moment based functions. Firstly, the Horizontal Profile feature of the document image is used to segment into line images. Thereafter, Vertical Profile features of each line image is used to extract individual word images. The word images are normalised to equal height and width of 256 pixels.

Using equation 7, moments up to the 7th order are extracted from the normalised word images. A feature vector consisting of 30 moment values obtained is constructed for each word image and stored in the main index. Experiments were conducted to determine the number of orders up to which moments should be computed. As shown in Figure 4, average precision increases with the rise in moment orders (up to a threshold of 7 orders), after which the precision rate falls. This can be attributed to the nature of higher order Geometrical Moments which are prone to adding noise in the feature set and thereby reduce the overall precision after a certain threshold. After the index has been constructed using the moment features, we create a template which keeps the mapping between a word image and its corresponding text. This template is used to generate a query word image corresponding to the query text input by the user. A similar feature extraction mechanism is performed on the query word image to obtain a query feature vector which is used to find the similarity between the query word image and all other word images present in the corpus.

5 Similarity Matching and Relevance Feedback

5.1 Cosine Similarity

A standard Vector Space Model is used represent the query word and all the candidate words. The index is maintained in the form of a word-feature matrix, where each word image $\vec{w}$ occupies one row of the matrix and all columns in a single row correspond to the moment values computed for the given word image.

When the user enters any query word, a lookup operation is performed in the stored template to obtain the corresponding normalised word image for the input text. Feature extraction is performed on the word image to construct the query feature vector $\vec{q}$. A cosine similarity score is computed for this query feature vector and all the rows of the word-feature matrix. The cosine similarity is calculated as follows:

$$SIM(q,w) = \frac{\vec{q} \ . \ \vec{w}}{\mid \vec{q} \mid * \mid \vec{w} \mid} \tag{8}$$

All the words of the document corpus are then ranked according to the cosine similarity score. The top choice returned by the ranking mechanism represents the word image which is most similar to the input query word.

5.2 Relevance Feedback

Since the word images present in the document corpus may be of poor print quality and may contain noise, the moment features computed may not be effective in ranking relevant word images higher in the obtained result. Also the presence of higher order moments may lead to inconsistency in the overall ranking of word images. To overcome this limitation, we have implemented a Rocchio's formula based manual Relevance Feedback mechanism. This mechanism re-formulates the query feature vector by adjusting the values of the individual moment orders present in the query vector. The relevance feedback mechanism assumes a user input after the presentation of the initial results. A user enters either a 1 denoting a result to be relevant or 0 denoting a result to be irrelevant. The new query vector is computed as follows:

$$q_{new} = \gamma . q_{old} + \frac{\alpha}{|R|} . \sum_{i=1}^{i=R} d_i - \frac{\beta}{|NR|} . \sum_{j=1}^{j=NR} d_j \tag{9}$$

where α , β and γ are term re-weighting constants. R denotes a relevant result set and NR denotes a non-relevant result set. For this experiment, we chose $\alpha = 1$, $\beta = 0.75$ and $\gamma = 0.25$.

6 Experiments and Results

The moment based features seem more robust in handling different image transformations compared to commonly used feature sets for word spotting such as GSC features [1] and Gabor filter based features [2]. This can be obseerved in Figure 5. The first row of the image corresponds to different types of transformations applied to normal English handwritten word images ((a)) such as changing the image scale as in (b) or (c). The second row corresponds to linear ((f)) and scale transformation ((e)), when applied to the normal machine printed Hindi word image ((d)). Even after undergoing such transformations, the cosine similarity score between the moment features extracted from all image pairs is still close to 1, which reflects the strength of invariance of moment based features with respect to image transformations. Table 1 shows the cosine similarity score between all pairs of English word

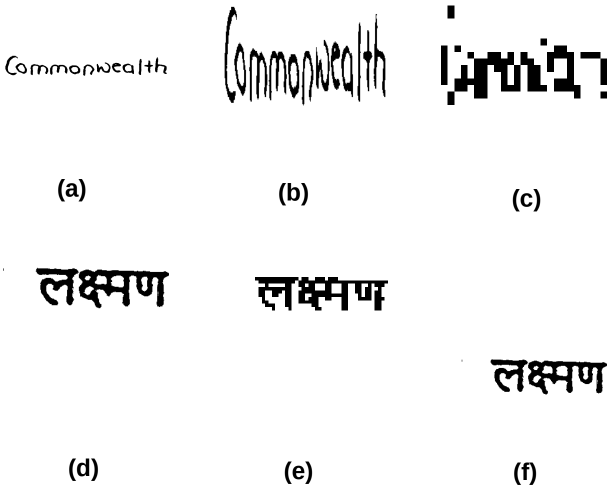


Figure 5: Various forms of Image Transformations. (a) & (d) Sample Word Image . (b),(c) & (e) Scale Transformation Examples (f) Linear Transformation Example .

Table 1: Cosine Similarity Score for English Transformed Word Image Pairs.

Word Image Pair	**(a)**	**(b)**	**(c)**
(a)	1	0.9867	0.9932
(b)	0.9867	1	0.9467
(c)	0.9932	0.9467	1

images. Table 2 shows the similarity score between all pairs of hindi word images.

The data set for evaluating our methodology consists of documents in three scripts, namely English, Hindi and Sanskrit. For English, we used publicly available IAMdb [13] handwritten word images and word images extracted from George Washington's publicly available historical manuscripts [14]. The dataset for English consists of 707 word images. For Hindi, 763 word images were extracted from publicly available Million Book Project documents [12]. For Sanskrit, 693 word images were extracted from 5 Sanskrit documents downloaded from the URL: http://sanskrit.gde.to/ . For public testing and evaluation, we have also made our dataset available at the location: http://cubs.buffalo.edu/ilt/dataset/.

For evaluating the system performance, we use the commonly used Average Precision Metric. Precision for

Table 2: Cosine Similarity Score for Hindi Transformed Word Image Pairs.

Word Image Pair	**(d)**	**(e)**	**(f)**
(d)	1	0.9662	0.9312
(e)	0.9662	1	0.9184
(f)	0.9312	0.9184	1

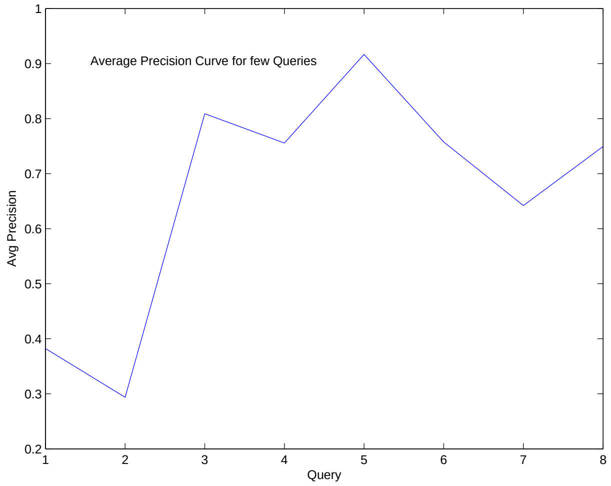


Figure 6: Average Precision curve for English Word Spotting.

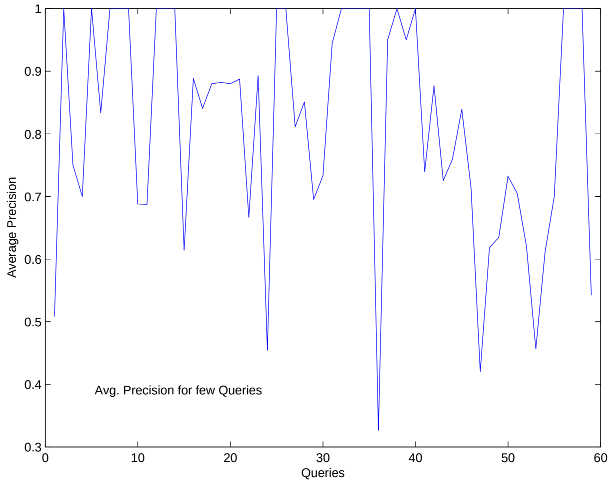


Figure 7: Average Precision curve for Hindi Word Spotting.

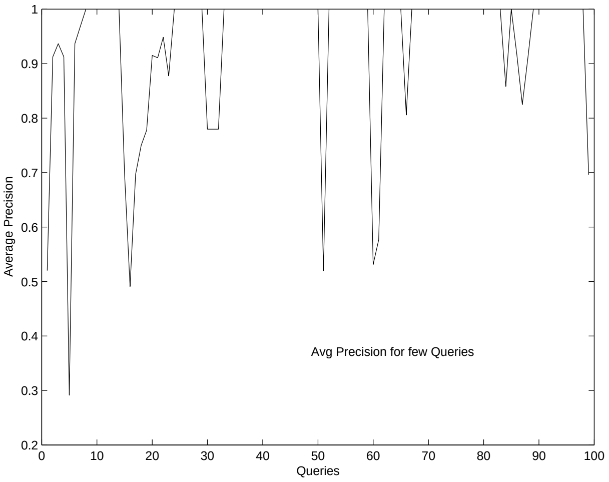


Figure 8: Average Precision curve for Sanskrit Word Spotting.

each query image was calculated at every recall level, and then averaged over to give an Average Precision per query. Figure 6 shows the average precision values for some query words in English. Figure 7 shows the average precision values for query words in Hindi. Figure 8 shows the average precision values for query words in Sanskrit.

The experimental results for all three scripts are summarised in Table 3. The Average Precision rates as shown in the table have been averaged over 30 queries in English, 75 queries in Hindi and 100 queries in Sanskrit. As shown here, the system works better for machine printed text (71.18 and 87.88) as compared to handwritten (67.0). The best performance is seen with Sanskrit script (87.88), which has a variable length words allowing it to be more discriminative in its feature analysis as compared to other two scripts. Table 4 compares the performance of GSC based word spotting as reported in [1] against our methodology. At 50% recall level, Moment based features perform better than GSC based features for both handwritten English and machine printed Sanskrit documents.

We also evaluate the performance of Gabor Feature based word spotting method [2] on our dataset. Features are extracted using an array of Gabor filters having a scale from 4 pixels to 6 pixels and 8 orientations. Table 5 summarizes the performance of Gabor features based method as opposed to our Moment based system. As shown , Moment based features outperform Gabor based features in terms of average precision rates obtained for all 3 scripts used in the experiment.

Table 3: Average Precision rate for word spotting in all 3 Scripts .

Script	Before RF	After RF
English	66.30	69.20
Hindi	71.18	74.34
Sanskrit	87.88	92.33

Table 4: Comparison of GSC and Moments based features at 50% recall level.

Script	GSC	Moments
English	60.0	71.6
Sanskrit	90.0	94.3

7 Summary and Conclusion

In this paper, we have proposed a framework for script independent word spotting in document images. We have shown the effectiveness of using statistical Moment based features as opposed to some of the structural and profile based features which may constrain the approach to few scripts. Another advantage of using moment based features is that they are image scale and translation invariant which makes them suitable for font independent feature analysis. In order to deal with the noise sensitivity of the higher order moments, we use a manual relevance feedback to improve the ranking of the relevant word images. We are currently working on extending our methodology to larger data sets and incorporating more scripts in future experiments.

Table 5: Comparison of Gabor filter based and Moments Features.

Script	Gabor	Moments
English	56.15	66.30
Hindi	67.25	71.18
Sanskrit	79.10	87.88

References

[1] S. N. Srihari, H. Srinivasan, C. Huang and S. Shetty, ”Spotting Words in Latin, Devanagari and Arabic Scripts,” Vivek: Indian Journal of Artificial Intelligence , 2006.

[2] Huaigu Cao, Venu Govindaraju, Template-Free Word Spotting in Low-Quality Manuscripts, the Sixth International Conference on Advances in Pattern Recognition (ICAPR), Calcutta, India, 2007.

[3] S. Kuo and O. Agazzi, Keyword spotting in poorly printed documents using 2-d hidden markov models, in IEEE Trans. Pattern Analysis and Machine Intelligence, 16, pp. 842848, 1994.

[4] M. Burl and P.Perona, Using hierarchical shape models to spot keywords in cursive handwriting, in IEEE-CS Conference on Computer Vision and Pattern Recognition, June 23-28, pp. 535540, 1998.

[5] A. Kolz, J. Alspector, M. Augusteijn, R. Carlson, and G. V. Popescu, A line oriented approach to word spotting in hand written documents, in Pattern Analysis and Applications, 2(3), pp. 153168, 2000.

[6] R. Manmatha and W. B. Croft, ”Word spotting: Indexing handwritten archives,”' In M. Maybury, editor, Intelligent Multimedia Information Retrieval Collection , AAAI/MIT Press, Menlo Park, CA, 1997.

[7] T. Rath and R. Manmatha, .Features for word spotting in historical manuscripts,. in Proc. International Conference on Document Analysis and Recognition, pp. 218.222, 2003.

[8] T. Rath and R. Manmatha, .Word image matching using dynamic time warping,. in Proceeding of the Conference on Computer Vision and Pattern Recognition (CVPR), pp. 521.527, 2003.

[9] Teh C.-H. and Chin R.T., ”‘On Image Analysis by the Methods of Moments,”' in IEEE Trans. Pattern Analysis and Machine Intelligence, 10, No. 4 , pp. 496513, 1988.

[10] Franz L. Alt , ”‘Digital Pattern Recognition by Moments,”' in The Journal of the ACM , Vol. 9 , Issue 2 , pp. 240-258 , 1962.

[11] Jeff L. Decurtins and Edward C. Chen ,”' Keyword spotting via word shape recognition ”' in Proc. SPIE Vol. 2422, p. 270-277, Document Recognition II, Luc M. Vincent; Henry S. Baird; Eds. , vol. 2422 , pp. 270-277, March 1995.

[12] Carnegie Mellon University - Million book project, URL: http://tera-3.ul.cs.cmu.edu/, 2007.

[13] IAM Database for Off-line Cursive Handwritten Text, URL: http://www.iam.unibe.ch/ zimmerma/iamdb/iamdb.html .

[14] Word Image Dataset at CIIR - UMass , URL: http://ciir.cs.umass.edu/cgi-bin/downloads/downloads.cgi .

A Document Graph Based Query Focused Multi-Document Summarizer

Sibabrata Paladhi
Department of Computer Sc. & Engg.
Jadavpur University, India
sibabrata_paladhi@yahoo.com

Sivaji Bandyopadhyay
Department of Computer Sc. & Engg.
Jadavpur University, India
sivaji_cse_ju@yahoo.com

Abstract

This paper explores the research issue and methodology of a query focused multi-document summarizer. Considering its possible application area is Web, the computation is clearly divided into offline and online tasks. At initial preprocessing stage an offline document graph is constructed, where the nodes are basically paragraphs of the documents and edge scores are defined as the correlation measure between the nodes. At query time, given a set of keywords, each node is assigned a query dependent score, the initial graph is expanded and keyword search is performed over the graph to find a spanning tree identifying relevant nodes satisfying the keywords. Paragraph ordering of the output summary is taken care of so that the output looks coherent. Although all the examples, shown in this paper are based on English language, we show that our system is useful in generating query dependent summarization for non- English languages also. We also present the evaluation of the system.

1 Introduction

With the proliferation of information in the Internet, it is becoming very difficult for users to identify the exact information. So many sites are providing same piece of information and a typical query based search in Google results in thousands of links if not million. Web Search engines generally produce query dependent snippets for each result which help users to explore further. An automated query focused multi-document summarizer, which will generate a query based short summary of web pages will be very useful to get a glimpse over the complete story. Automated multi-document summarization has drawn much attention in recent years. Most multi-document summarizers are query independent, which produce majority of information content from multiple documents using much less lengthy text. Each of the systems fall into two different categories: either they are sentence extraction based where they just extract relevant sentences and concatenate them to produce summary or they fuse information from multiple sources to produce a coherent summary.

In this paper, we propose a query focused multi-document summarizer, based on paragraph extraction scheme. Unlike traditional extraction based summarizers which do not take into consideration the inherent structure of the document, our system will add structure to documents in the form of graph. During initial preprocessing, text fragments are identified from the documents which constitute the nodes of the graph. Edges are defined as the correlation measure between nodes of the graph. We define our text fragments as paragraph rather than sentence with the view that generally a paragraph contains more correlated information whereas sentence level extraction might lead to loss of some coherent information.

Since the system produces multi-document summary based on user's query, the response time of the system should be minimal for practical purpose. With this goal, our system takes following steps: First, during preprocessing stage (offline) it performs some query independent tasks like identifying seed summary nodes and constructing graph over them. Then at query time (online), given a set of keywords, it expands the initial graph and performs keyword search over the graph to find a spanning tree identifying relevant nodes (paragraphs) satisfying the keywords. The performance of the system depends much on the identification of the initial query independent nodes (seed nodes). Although, we have presented all the examples in the current discussion for English language only, we argue that our system can be adapted to work in multilingual environment (i.e. Hindi, Bengali, Japanese etc.) with some minor changes in implementation of the system like incorporating language dependent stop word list, stemmer, WodrNet like lexicon etc.

In section 2, related works in this field is presented. In section 3 the overall approach is described. In section 4 query independent preprocessing steps are explained. In section 5 query dependent summary generation and paragraph ordering scheme is presented. Section 6 presents the evaluation scheme of the system. In section 7 we discuss how our system can be modified to work in multilingual scenario. In section 8 we have drawn conclusion and discussed about future work in this field.

2 Related Work

A lot of research work has been done in the domain of multi-document summarization (both query dependent/independent). MEAD (Radev et al., 2004) is centroid based multi-document summarizer which generates summaries using cluster centroids produced by topic detection and tracking system. NeATS (Lin and Hovy, 2002) selects important content using sentence position, term frequency, topic signature and term clustering. XDoX (Hardy et al., 2002) identifies the most salient themes within the document set by passage clustering and then composes an extraction summary, which reflects these main themes.

Graph based methods have been proposed for generating query independent summaries. Websumm (Mani and Bloedorn, 2000) uses a graph-connectivity model to identify salient information. Zhang et al (2004) proposed the methodology of correlated summarization for multiple news articles. In the domain of single document summarization a system for query-specific document summarization has been proposed (Varadarajan and Hristidis, 2006) based on the concept of document graph.

In this paper, the graph based approach has been extended to formulate a framework for generating query dependent summary from related multiple document set describing same event.

3 Graph Based Modeling

The proposed graph based multi-document summarization method consists of following steps: (1) The document set D = $\{d_1,d_2, \ldots\ d_n\}$ is processed to extract text fragments, which are paragraphs in our case as it has been discussed earlier. Here, we assume that the entire document in a particular set are related i.e. they describe the same event. Some document clustering techniques may be adopted to find related documents from a large collection. Document clustering is out of the scope of our current discussion and is itself a research interest. Let for a document d_i, the paragraphs are $\{p_{i1},p_{i2},\ldots p_{im}\}$. But the system can be easily modified to work with sentence level extraction. Each text fragment becomes a node of the graph. (2) Next, edges are created between nodes across the document where edge score represents the degree of correlation between inter documents nodes. (3) Seed nodes are extracted which identify the relevant paragraphs within D and a search graph is built offline to reflect the semantic relationship between the nodes. (4) At query time, each node is assigned a query dependent score and the search graph is expanded. (5) A query dependent multi-document summary is generated from the search graph which is nothing but constructing a total minimal spanning tree T (Varadarajan and Hristidis, 2006). For a set of keywords Q = $\{q_1,q_2, .. q_n\}$, T is total if $\forall q \in Q$, T consists of at least one node satisfying q and T is minimal if no node can be removed from T while getting the total T.

4 Building Query Independent Components

Mainly there are two criteria for the performance evaluation of such systems: First it's accuracy i.e. the quality of output with respect to specific queries and next of course the turn around time i.e., how fast it can produce the result. Both are very important aspects of such system, and we will show how these aspects are taken care of in our system. Runtime of such system greatly depends on how well the query independent graph is constructed. At one extreme, offline graph can be built connecting all the nodes from each of the documents, constituting a total document graph. But keyword search over such large graph is time consuming and practically not plausible. On the other hand, it is possible to select query specific nodes at runtime and to create a graph over those nodes. But if the number of such nodes is high, then calculating similarity scores between all the nodes will take large computing time, thus resulting in slower performance.

We will take an intermediate approach to attack the problem. It can be safely assumed that significant information for a group of keywords can be found in "relevant/topic paragraphs" of the documents. So, if relevant/topic nodes can be selected from document set D during offline processing, then the significant part of the search graph can be constructed offline which greatly reduce the online processing time. For example, if a user wants to find the information about the IBM Hindi speech recognition system, then the keywords are likely to be {IBM, speech recognition, accuracy}. For a set of news articles about this system, the topic paragraphs, identified offline, naturally satisfy first two keywords and theoretically, they are the most informative paragraphs for those keywords. The last term 'accuracy' (relevant for accuracy of the system) may not be satisfied by seed nodes. So, at run time, the graph needs to be expanded purposefully by including nodes so that the paragraphs, relevant to 'accuracy of the system' are included.

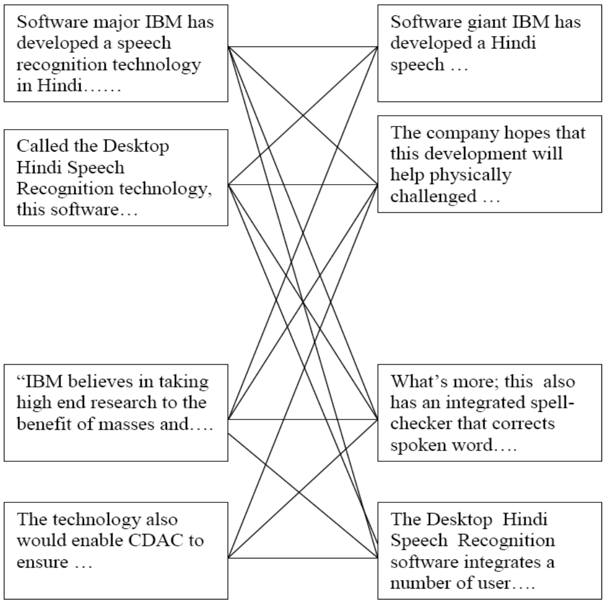
4.1 Identification of Seed/ Topic Nodes

At the preprocessing stage, text is tokenized, stop words are eliminated, and words are stemmed (Porter, 1980). The text in each document is split into paragraphs and each paragraph is represented with a vector of constituent words. If we consider pair of related document, then the inter document graph can be represented as a set of nodes in the form of bipartite graph. The edges connect two nodes corresponding to paragraphs from different documents. The similarity between two nodes is expressed as the edge weight of the bipartite graph. Two nodes are related if they share common words (except stop words) and the degree of relationship can be measured by adapting some traditional IR formula (Varadarajan and Hristidis, 2006).

$$Score(e) = \frac{\sum ((tf(t(u),w) + tf(t(v),w)).idf(w))}{size(t(u)) + size(t(v))}$$

Where $tf(d,w)$ is number of occurrence of w in d, $idf(w)$ is the inverse of the number of documents containing w, and $size(d)$ is the size of the documents in words. The score can be accurately set if stemmer and lexicon are used to match the equivalent words. With the idea of page ranking algorithms, it can be easily observed that a paragraph in a document is relevant if it is highly related to many relevant paragraphs of other document. If some less stringent rules are adopted, then a node from a document is selected as seed/topic node if it has high edge scores with nodes of other document. Actually for a particular node, total edge score is defined as the sum of scores of all out going edges from that node. The nodes with higher total edge scores than some predefined threshold are included as seed nodes. In Figure 1. correlation between two news articles is shown as a bipartite graph.

But the challenge for multi-document summarization is that the information stored in different documents inevitably overlap with each other. So, before inclusion of a particular node (paragraph), it has to be checked whether it is being repeated or not. Two paragraphs are said to be similar if they share for example, 70% words (non stop words) in common.



Software major IBM has developed a speech recognition technology in Hindi……

Called the Desktop Hindi Speech Recognition technology, this software…

"IBM believes in taking high end research to the benefit of masses and….

The technology also would enable CDAC to ensure …

Software giant IBM has developed a Hindi speech …

The company hopes that this development will help physically challenged …

What's more; this also has an integrated spell-checker that corrects spoken word….

The Desktop Hindi Speech Recognition software integrates a number of user….

Figure 1. A bipartite graph representing correlation among two news articles on same event.

4.2 Offline Construction of Search Graph

After detection of seed/topic nodes a search graph is constructed. For nodes, pertaining to different documents, edge scores are already calculated, but for intra document nodes, edge scores are calculated in the similar fashion as said earlier. Since, highly dense graph leads to higher search/execution time, only the edges having edge scores well above the threshold value might be considered. The construction of query independent part of the search graph completes the offline processing phase of the system.

5 Building Query Dependent Components

At query time, first, the nodes of the already constructed search graph are given a query dependent score. The score signifies the relevance of the paragraph with respect to given queries. During evaluation if it is found that any keyword is not satisfied by the seed nodes, then system goes back to individual document structure and collects relevant nodes. Finally, it expands the offline graph by adding those nodes, fetched from individual documents. Next, the expanded search graph is processed to find the total minimum spanning tree T over the graph.

5.1 Expanding Search Graph

When query arrives, system evaluates nodes of the offline search graph and computes query dependent score. This computation is based on ranking principals from IR community. The most popular IR ranking is okapi equation (Varadarajan and Hristidis, 2006) which is based on tf-idf principle.

$$\sum_{t\in Q,d} \ln \frac{N - df + 0.5}{df + 0.5} \cdot \frac{(k_{1}+1).tf}{(k_{1}(1-b) + b\frac{dl}{avdl}) + tf} \cdot \frac{(k_{3}+1).qtf}{k_{3} + qtf}$$

tf is the term’s frequency in document, qtf is term’s frequency in the query, N is the total no. of documents in collection, df is the number of documents that contain the term, dl is the document length (number of words), avdl is the average document length and k1 (1.0 – 2.0), b (0.75), k3 (0 -1000) are constants.

During node score computation, the system intelligently partitions the query set Q into two parts. One part consists of q_i’s which are satisfied by at least one node from offline search graph. The other part consists of q_i’s which are not satisfied by any node from offline search graph. The system then computes query dependent scores for the nodes of all the individual documents for the unsatisfied keyword set and relevant nodes (having score above threshold) are added to the search graph. Edge scores are computed only for edges connecting newly added nodes with the existing ones and between the new nodes. In this way, the offline graph is expanded by adding some query dependent nodes at runtime. Query dependent scoring can be made faster using a full text indexing which is a mapping $K_i \rightarrow (D_i , N_i)$; where K_i’s are content words (i.e., not stop words) and D_i’s and Ni’s are respectively the document ids and the node ids within the document set. Since, the node score is calculated at runtime, it needs to be accelerated. Thus a full text index developed offline will be of great help.

5.2 Summary Generation

Summary generation is basically a keyword search technique in the expanded search graph. This is to mention that the search technique discussed here is basically based on AND semantic, i.e. it requires all the keywords to be present in the summary, but the algorithm can be modified to take care of OR semantic also. Keyword search in graph structure is itself a research topic and several efficient algorithms are there to solve the problem. DBXplorer (Agrawal et al., 2002), BANKS (Bhalotia et al., 2002), are popular algorithms in this field which consider relational database as graph and devise algorithms for keyword based search in the graph. Finally, Varadarajan and Hristidis (2006) has proposed Top-k Enumeration and MultiResultExpanding search for constructing total minimum spanning tree over a document graph. Any of the above popular algorithms can be adapted to use within our framework.

In our system we have used a search algorithm which finds different combinations of nodes that represent total spanning tree. For each of the combination we compute score of the summary based on some IR principle (Varadarajan and Hristidis, 2006). Then we take the one having best score (minimal in our case). If the graph is not too dense, then the response time will be small enough. The equation given below is used to compute the score of individual spanning tree T.

$$T_{score} = a\sum_{e\in T}\frac{1}{e_{score}} + b\,\frac{1}{\sum_{n\in T} n_{score}}$$

Where T_{score} the score of the spanning tree, e and n is are edge and node of T respectively, e_{score} and n_{score} are edge score and individual node score respectively. a and b are non zero positive constants in the range of [0 – 1]. For a particular search graph, it is possible to find many total spanning trees, having different summary scores. In our system, the summary with the best score is considered.

In Figure 2 two sample news stories are shown along with system identified seed nodes, shown in bold. A query based summary from that related document set is shown in Figure 3.

5.3 Paragraph Ordering Scheme

In the previous sections, the techniques for generation of summary nodes have been discussed. Here, we will investigate the method for ordering them into a coherent text. In case of single document summarization, sentence/paragraph ordering is done based on the position of extracted paragraphs/ sentences in the original document. But in multi-document scenario, the problem is non trivial since information is extracted from different documents and no single document can provide ordering. Besides, the ordering of information in two different documents may be significantly varying because

P_0: Software giant IBM has developed a speech recognition software in Hindi.
P_1 : The company hopes that this development will help physically challenged and less literate Hindi speakers to access information using a variety of applications.
P_2 : **The Desktop Hindi Speech Recognition Technology developed by the IBM India Software Lab in collaboration with Centre for Development of Advanced Computing would provide a natural interface for human-computer interaction.**
P_3 : The new IBM technology could help to provide a natural interface for human-computer interaction.
P_4: According to Dr. Daniel Dias, Director, IBM Indian Research Laboratory, the technology which helps transcribe continuous Hindi speech instantly into text form, could find use in a variety of appli In Figure 1. correlation between two news articles is shown as a bipartite graph. cations like voice-enabled ATMs, car navigation systems, banking, telecom, railways, and airlines.
P_5: **Besides, the technology could also enable C-DAC to ensure a high level of accuracy in Hindi translation in a number of domains like administration, finance, agriculture and the small-scale industry.**
P_6: The IBM Desktop Hindi Speech Recognition software is capable of recognizing over 75,000 Hindi words with dialectical variations, providing an accuracy of 90 to 95%.
P_7: What's more; this software also has an integrated spellchecker that corrects spoken-word errors, enhancing the accuracy to a great extent.
P_8: The Desktop Hindi Speech Recognition Technology also integrates a number of user-friendly features such as the facility to convert text to digits and decimals, date and currency format, and into fonts which could be imported to any Windows-based application.
P_9: "IBM believes in taking high-end research to the benefit of the masses and bridging the digital divide through a faster diffusion process," concluded Dias.

P_0: **Software major IBM has developed a speech recognition technology in Hindi which would help physically challenged and less literate Hindi speakers access information through a variety of systems.**
P_1 : Called the Desktop Hindi Speech Recognition technology, this software was developed by the IBM India Software Lab jointly with the Centre for Development of Advanced Computing.
P_2 : **The technology, which helps transcribe continuous Hindi speech instantly into text form, could find use in a variety of applications like voice-enabled ATMs, car navigation systems, banking, telecom, railways and airlines, said Dr Daniel Dias, Director, IBM India Research Laboratory.**
P_3 : The system can recognize more than 75,000 Hindi words with dialectical variations, providing an accuracy level of 90-95 per cent, he said.
P_4: A spellchecker to correct spoken-word errors also enhances the accuracy of the system.
P_5: **The technology also has integrated many user-friendly features such as facility to convert text to digits and decimals, date and currency format, and into fonts which could be imported to any windows-based application.**
P_6: "IBM believes in taking high-end research to the benefit of the masses and bridging the digital divide through a faster diffusion process", Dias said.
P_7: The technology also would enable C-DAC to ensure high-level accuracy in Hindi translation in a host of domains, including administration, finance, agriculture and small scale industry.

Figure 2. Paragraphs of two news articles with five extracted seed/ topic paragraphs (in bold). Underlined paragraphs are added later during graph expansion phase.

Software major IBM has developed a **speech recognition** technology in Hindi which would help physically challenged and less literate Hindi speakers access information through a variety of systems. [Doc-2, Para - 0]
Besides, the technology could also enable C-DAC to ensure a high level of **accuracy** in Hindi translation in a number of domains like administration, finance, agriculture and the small-scale industry. [Doc-1, Para-5]
A **spellchecker** to correct spoken-word errors also enhances the **accuracy** of the system. [Doc-2, Para - 4]

Figure 3. Automatic summary based on {speech recognition, accuracy, spellchecker} query

the writing styles of different authors are different. In case of news event summarization, chronological ordering is a popular choice which considers the temporal sequence of information pieces, when deciding the ordering process.

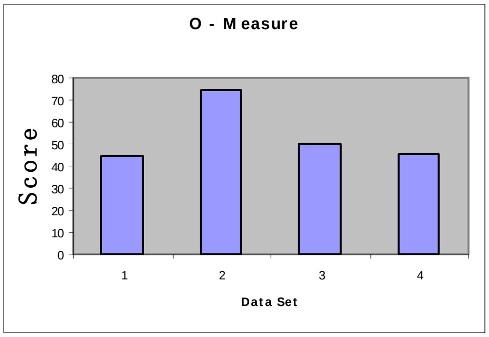
In this paper, we will propose a scheme of ordering which is different from the above two approaches in that, it only takes into consideration the semantic closeness of information pieces (paragraphs) in deciding the ordering among them. First, the starting paragraph is identified which is the paragraph with lowest positional ranking among selected ones over the document set. Next for any source node (paragraph) we find the summary node that is not already selected and have (correlation value) with the source node. This node will be selected as next source node in ordering. This ordering process will continue until the nodes are totally ordered. The above ordering scheme will order the nodes independent of the actual ordering of nodes in the original document, thus eliminating the source bias due to individual writing style of human authors. Moreover, the scheme is logical because we select a paragraph for position p at output summary, based on how coherent it is with the (p-1)th paragraph.

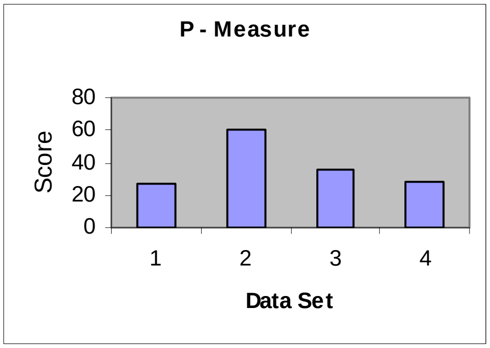
6 Evaluation

Evaluation of summarization methods is generally performed in two ways. Evaluation measure based on information retrieval task is termed as the *extrinsic* method, while the evaluation based on user judgments is called the *intrinsic* measure. We adopted the latter, since we concentrated more on user's satisfaction. We measure the quality of output based on the percentage of overlap of system generated output with the manual extract. Salton et al (1997) observed that an extract generated by one person is likely to cover 46% of the information that is regarded as most important by another person. Mitra et. al. (1998) proposed an interesting method for evaluation of paragraph based automatic summarization and identified the following four quality-measures – Optimistic (O), Pessimistic (P), Intersection (I) and Union (U) based evaluation. For evaluation purpose, we identify different related document set (D) from different domains like technical, business etc and keyword (query) list for each domain. Users are asked to manually prepare the multi-document summarization based on the given queries. They prepared it by marking relevant paragraphs over D. Based on the excerpts prepared by the users; the above scores are calculated as O: Percentage overlap with that manual extract for which the number of common paragraphs is highest, P: Percentage overlap with that manual extract for which the number of common paragraphs is lowest; I: Percentage overlap with the intersection of manual extracts; U: Percentage overlap with the union of manual extracts. The results are shown in Table 1. A comparative survey of quality measures for the set of articles is shown in Figure 3.

D	$O_{measure}$	$P_{measure}$	$I_{measure}$	$U_{measure}$
article1 & article2	44.4	27	33.3	66.6
article3 & article4	75	60	50	100
article5 & article6	50	35.5	25	66
article7 & article8	45.5	28.7	33.3	56.4

Table 1. Evaluation score





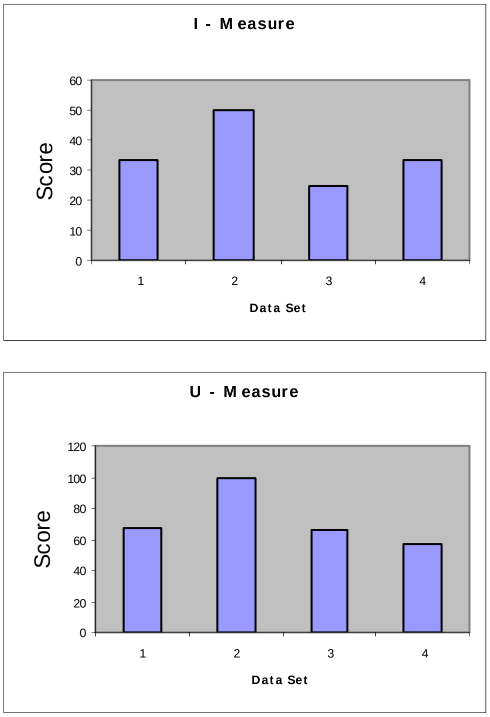


Figure 3. Comparative measure scores for set of articles

7 Baseline Approach to Multilingual Summarization

Our baseline approach to multilingual multidocument summarization is to apply our English based multi-document summarization system to other non-English languages like Hindi, Bengali, Japanese etc. We have initially implemented the system for English language only, but it can be modified to work in multilingual scenario also. To work with other languages, the system requires some language dependent tools for that particular language:

1) A stop word list of that language is required because they have no significance in finding similarity between the paragraphs and need to be removed during initial preprocessing stage.

2) A language dependent stemmer is required. In most of the languages, stemmer is yet to be developed. Another problem is that suffix stripping is not the only solution for all languages because some languages have affix, circumfix etc. in their inflected form. A morphological analyzer to find the root word may be used for those languages.

3) A lexicon for that language is required to match the similar words. For English, WordNet is widely available. For other languages also, similar type of lexicons are required.

If these tools are available then our system can be tuned to generate query dependent multilingual multi-document summary.

8 Conclusion and Future Work

In this work we present a graph based approach for query dependent multi-document summarization system. Considering its possible application in the web document, we clearly divided the computation into two segments. Extraction of seed/topic summary nodes and construction of offline graph is a part of query independent computation. At query time, the precomputed graph is processed to extract the best multi-document summary. We have tested our algorithm with news articles from different domains. The experimental results suggest that our algorithm is effective. Although we experimented with pair of articles, the proposed algorithm can be improved to handle more than two articles simultaneously.

The important aspect of our system is that it can be modified to compute query independent summary which consists of topic nodes, generated during preprocessing stage. The paragraph ordering module can be used to define ordering among those topic paragraphs. Another important aspect is that our system can be tuned to generate summary with custom size specified by users. The spanning tree generation algorithm can be so modified that it produces not only total spanning tree but also takes care of the size requirement of user. Lastly, it is shown that our system can generate summary for other non-English documents also if some language dependent tools are available.

The performance of our algorithm greatly depends on quality of selection of topic nodes. So if we can improve the identification of topic paragraphs and shared topics among multiple documents it would surely enhance the quality of our system.

9 References

A. Singhal , M. Mitra, and C. Buckley. 1997. Automatic Text Summarization by Paragraph Extraction. *Proceedings of* ACL/EACL Workshop.

C.-Y. Lin and E.H. Hovy. 2002. From Single to Multi-document Summarization: A Prototype System and its Evaluation. *Proceedings of ACL:* 457–464.

D.R. Radev, H. Jing, M. Styś and D. Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing and Management*, Vol.40:919–938.

G. Salton , A. Singhal , M. Mitra, and C. Buckley. 1997. Automatic text structuring and summarization. *Information Processing and Management*: Vol. 33, No. 2: 193-207.

G. Bhalotia, C. Nakhe, A. Hulgeri, S. Chakrabarti and S.Sudarshan. 2002. Keyword Searching and Browsing in Databases using BANKS. *Proceedings of ICDE* : 431-440.

H. Hardy, N. Shimizu, T. Strzalkowski, L. Ting, G. B. Wise and X. Zhang. 2002. Cross-document summarization by concept classification. *Proceedings of SIGIR.02*: 65-69 .

I. Mani and E. Bloedorn. 2000. Summarizing Similarities and Differences Among Related Documents. *Information Retrieval*, Vol. 1(1): 35-67.

M. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

R. Varadarajan,. V. Hristidis. 2006. A system for query-specific document summarization. *Proceedings of CIKM 2006*: 622-631.

S. Agrawal, S. Chaudhuri, and G. Das.2002. DBXplorer: A System for Keyword-Based Search over Relational Databases. *Proceedings of ICDE:* 5-16.

Y. Zhang, X. Ji, C. H. Chu, and H. Zha. 2004. Correlating Summarization of Multisource News with K-Way Graph Biclustering. *SIGKDD Explorations 6*(2): 34-42.

CLIR in Indian Languages - Invited Talks

Hindi and Marathi to English Cross Language Information

Manoj Kumar Chinnakotla, Sagar Ranadive, Om P. Damani and Pushpak Bhattacharyya

Abstract:

In this paper, we present our Hindi ->English and Marathi ->English CLIR systems developed as part of our participation in the CLEF 2007 Ad-Hoc Bilingual task. We take a query translation based approach using bi-lingual dictionaries. Query words not found in the dictionary are transliterated using a simple lookup table based transliteration approach. The resultant transliteration is then compared with the index items of the corpus to return the `k' closest English index words of the given Hindi/Marathi word. The resulting multiple translation/transliteration choices for each query word are disambiguated using an iterative page-rank style algorithm, proposed in the literature, which makes use of term-term co-occurrence statistics to produce the final translated query. Using the above approach, for Hindi, we achieve a Mean Average Precision (MAP) of 0.2366 in title which is 61.36\% of monolingual performance and a MAP of 0.2952 in title and description which is 67.06\% of monolingual performance. For Marathi, we achieve a MAP of 0.2163 in title which is 56.09\% of monolingual performance.

Bengali and Hindi to English CLIR Evaluation

Debasis Mandal, Sandipan Dandapat, Mayank Gupta, Pratyush Banerjee, Sudeshna Sarkar

Abstract

Our participation in CLEF 2007 consisted of two Cross-lingual and one monolingual text retrieval in the Ad-hoc bilingual track. The cross-language task includes the retrieval of English documents in response to queries in two Indian languages, Hindi
and Bengali. The Hindi and Bengali queries were first processed using a morphological analyzer (Bengali), a stemmer (Hindi) and a set of 200 Hindi and 273 Bengali stop words. The refined hindi queries were then looked into the Hindi-English bilingual lexicon, 'Shabdanjali' (approx. 26K Hindi words) and all of the corresponding translations were considered for the equivalent English query generation, if a match was found. Rest of the query words were transliterated using the ITRANS scheme. For the Bengali query, we had to depend mostly on the translietrations due to the lack of any effective Bengali-English bilingual lexicon. The final equivalent English query was then fed into the Lucene Search engine for the monolingual retrieval of the English documents. The CLEF evaluations suggested the need for a rich bilingual lexicon, a good Named Entity Recognizer and a better transliterator for CLIR involving Indian languages. The best MAP values for Bengali and Hindi CLIR for our experiment were 7.26 and 4.77 which are 0.20 and 0.13 of our monolingual retrieval, respectively.

Bengali, Hindi and Telugu to English Ad-hoc Bilingual task

Sivaji Bandyopadhyay, Tapabrata Mondal, Sudip Kumar Naskar,
Asif Ekbal, Rejwanul Haque, Srinivasa Rao Godavarthy

Abstract

This paper presents the experiments carried out at Jadavpur University as part of participation in the CLEF 2007 ad-hoc bilingual task. This is our first participation in the CLEF evaluation task and we have considered Bengali, Hindi and Telugu as query languages for the retrieval from English document collection. We have discussed our Bengali, Hindi and Telugu to English CLIR system as part of the ad-hoc bilingual task, English IR system for the ad-hoc monolingual task and the associated experiments at CLEF. Query construction was manual for Telugu-English ad-hoc bilingual task, while it was automatic for all other tasks.

Cross-Lingual Information Retrieval System for Indian Languages

Jagadeesh Jagarlamudi and A Kumaran

Abstract

This paper describes our first participation in the Indian language sub-task of the main Adhoc monolingual and bilingual track in CLEF competition. In this track, the task is to retrieve relevant documents from an English corpus in response to a query expressed in different Indian languages including Hindi, Tamil, Telugu, Bengali and Marathi. Groups participating in this track are required to submit a English to English monolingual run and a Hindi to English bilingual run with optional runs in rest of the languages. We had submitted a monolingual English run and a Hindi to English cross-lingual run.

We used a word alignment table that was learnt by a Statistical Machine Translation (SMT) system trained on aligned parallel sentences, to map a query in source language into an equivalent query in the language of the target document collection. The relevant documents are then retrieved using a Language Modeling based retrieval algorithm. On CLEF 2007 data set, our official cross-lingual performance was 54.4\% of the monolingual performance and in the post submission experiments we found that it can be significantly improved up to 73.4\%.

Hindi and Telugu to English CLIR using Query Expansion

Prasad Pingali and Vasudeva Varma

Abstract

This paper presents the experiments of Language Technologies Research Centre (LTRC) as part of their participation in CLEF2 2007 Indian language to English ad-hoc cross language document retrieval task. In this paper we discuss our Hindi and Telugu to English CLIR system and the experiments using CLEF 2007 dataset. We used a variant of TFIDF algorithm in combination with a bilingual lexicon for query translation. We also explored the role of a document summary in fielded queries and two different boolean formulations of query translations. We find that a hybrid boolean formulation using a combination of boolean AND and boolean OR operators improves ranking of documents. We also find that simple disjunctive combination of translated query keywords results in maximum recall.

FIRE: Forum for Information Retrieval Evaluation

Mandar Mitra, Prasenjit Majumder

Abstract

This talk will present our plans for organizing FIRE, a forum for Information Retrieval evaluation, focused on Indian languages. We will start by reviewing the basic experimental framework and metrics as represented by the Cranfield paradigm. An overview of the main evaluation fora for IR -- TREC, CLEF and NTCIR, which are all based on this paradigm, and from which FIRE draws inspiration -- will be given. The components of the evaluation framework, viz. corpora, search topics, and relevance judgments will also be discussed.

Author Index