Coling 2008

22nd International Conference on Computational Linguistics

Proceedings of the

2nd workshop on Multi-source, Multilingual Information Extraction and Summarization

23 August 2008
Manchester, UK

Order copies of this and other Coling proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-905593-51-4

Design by Chimney Design, Brighton, UK
Production and manufacture by One Digital, Brighton, UK

Editors' Foreword

Information extraction (IE) and text summarization (TS) are key technologies aiming at extracting relevant information from texts and presenting the information to the user in a condensed form. The ongoing information explosion makes IE and TS particularly critical for successful functioning within the information society. These technologies, however, face new challenges with the adoption of the Web 2.0 paradigm (e.g., blogs, wikis) due to their inherent multi-source nature. These technologies must no longer deal only with isolated texts or narratives, but with large-scale repositories or sources—possibly in several languages—containing a multiplicity of views, opinions, and commentaries on particular topics, entities and events. There is thus a need to adapt and/or develop new techniques to deal with these new phenomena.

Recognising similar information across different sources and/or in different languages is of paramount importance in this multi-source, multi-lingual context. In information extraction, merging information from multiple sources can lead to increased accuracy, as compared to extraction from a single source. In text summarization, similar facts found across sources can inform sentence scoring algorithms. In question answering, the distribution of answers in similar contexts can inform answer-ranking components. Often, it is not the similarity of information that matters, but its complementary nature. In a multi-lingual context, information extraction and text summarization can provide solutions for cross-lingual access: key pieces of information can be extracted from different texts in one or many languages, merged, and then conveyed in natural language in concise form. Applications need to be able to cope with the idiosyncratic nature of the new Web 2.0 media: mixed input, new jargon, ungrammatical and mixed-language input, emotional discourse, etc. In this context, synthesizing or inferring opinions from multiple sources is a new and exciting challenge for NLP. On another level, profiling of individuals who engage in the new social Web, and identifying whether a particular opinion is appropriate/relevant in a given context are important topics to be addressed.

The objective of this second *Multi-source Multilingual Information Extraction and Summarization* (MMIES) workshop is to bring together researchers and practitioners in information-access technologies, to discuss recent approaches for dealing with multi-source and multi-lingual challenges. Each paper submitted to the workshop was reviewed by three members of an international Programme Committee. The selection process resulted in this volume of eight papers, covering the following key topics:

- Multilingual Named Entity Recognition,
- Automatic Construction of Multilingual Dictionaries for Information Retrieval,
- Multi-document Summaries for Geo-referenced Images,
- Keyword Extraction for Single-Document Summarization,
- Recognizing Similar News over Time and across Languages,
- Speech-to-Text Summarization,
- Automatic Annotation of Bibliographical References.

We are grateful to the members of the programme committee for their invaluable work, as well as to Roger Evans, Mark Stevenson and Harold Somers for their support.

We thank Robert Gaizauskas for giving the invited talk at the workshop.

July 2008.

Sivaji Bandyopadhyay, Jadavpur University (India)
Thierry Poibeau, CNRS / Université Paris 13 (France)
Horacio Saggion, University of Sheffield (UK)
Roman Yangarber, University of Helsinki (Finland)

Organizers

- Sivaji Bandyopadhyay, Jadavpur University (India)
- Thierry Poibeau, CNRS and University of Paris 13 (France)
- Horacio Saggion, University of Sheffield (United Kingdom)
- Roman Yangarber, University of Helsinki (Finland)

Programme Committee

- Javier Artiles, UNED (Spain)
- Kalina Bontcheva, University of Sheffield (UK)
- Nathalie Colineau, CSIRO (Australia)
- Nigel Collier, NII (Japan)
- Hercules Dalianis, KTH/Stockholm University (Sweden)
- Thierry Declerk, DFKI (Germany)
- Michel Généreux, LIPN-CNRS (France)
- Julio Gonzalo, UNED (Spain)
- Brigitte Grau, LIMSI-CNRS (France)
- Ralph Grishman, New York University (USA)
- Kentaro Inui, NAIST (Japan)
- Min-Yen Kan, National University of Singapore (Singapore)
- Guy Lapalme, University of Montreal (Canada)
- Diana Maynard, University of Sheffield (UK)
- Jean-Luc Minel, Modyco-CNRS (France)
- Constantin Orasan, University of Wolverhampton (UK)
- Cecile Paris, CSIRO (Australia)
- Maria Teresa Pazienza, University of Roma 'Tor Vergata' (Italy)
- Bruno Pouliquen, European Commission – Joint Research Centre (Italy)
- Patrick Saint-Dizier, IRIT-CNRS (France)

- Agnes Sandor, Xerox XRCE (France)
- Satoshi Sekine, NYU (USA)
- Ralf Steinberger, European Commission – Joint Research Centre (Italy)
- Stan Szpakowicz, University of Ottawa (Canada)
- Lucy Vanderwende, Microsoft Research (USA)
- José Luis Vicedo, Universidad de Alicante (Spain)

Table of Contents

Conference Programme

Wednesday, August 23, 2008

	Invited Talk
9:30–10:30	*Generating Image Captions using Topic Focused Multi-document Summarization* Robert Gaizauskas
10:30–11:00	Coffee break
	Session 1: Named Entity and Lexical Resources for IE and Summarization
11:00–11:30	*Learning to Match Names Across Languages* Inderjeet Mani, Alex Yeh and Sherri Condon
11:30–12:00	*Automatic Construction of Domain-specific Dictionaries on Sparse Parallel Corpora in the Nordic languages* Sumithra Velupillai and Hercules Dalianis
12:00–12:30	*Graph-Based Keyword Extraction for Single-Document Summarization* Marina Litvak and Mark Last
12:30–14:00	Lunch
	Session 2: Multi-document Summarization
14:00–14:30	*MultiSum: Query-Based Multi-Document Summarization* Mike Rosner and Carl Camilleri
14:30–15:00	*Mixed-Source Multi-Document Speech-to-Text Summarization* Ricardo Ribeiro and David Martins de Matos
15:00–15:30	*Evaluating automatically generated user-focused multi-document summaries for geo-referenced images* Ahmet Aker and Robert Gaizauskas
15:30–16:00	Coffee break

Wednesday, August 23, 2008 (continued)

	Session 3: Applications
16:00–16:30	*Story tracking: linking similar news over time and across languages* Bruno Pouliquen, Ralf Steinberger and Olivier Deguernel
16:30–17:00	*Automatic Annotation of Bibliographical References with target Language* Harald Hammarström
17:00–17:30	Open Discussion

Generating Image Captions using Topic Focused Multi-document Summarization

Robert Gaizauskas
Natural Language Processing Group
Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello, Sheffield, S1 4DP, UK
R.Gaizauskas@sheffield.ac.uk

In the near future digital cameras will come standardly equipped with GPS and compass and will automatically add global position and direction information to the metadata of every picture taken. Can we use this information, together with information from geographical information systems and the Web more generally, to caption images automatically?

This challenge is being pursued in the TRIPOD project (http://tripod.shef.ac.uk/) and in this talk I will address one of the subchallenges this topic raises: given a set of toponyms automatically generated from geo-data associated with an image, can we use these toponyms to retrieve documents from the Web and to generate an appropriate caption for the image?

We begin assuming the toponyms name the principal objects or scene contents in the image. Using web resources (e.g. Wikipedia) we attempt to determine the types of these things – is this a picture of church? a mountain? a city? We have constructed a taxonomy of such image content types using on-line collections of captioned images and for each type in the taxonomy we have constructed several collections of texts describing that type. For example, we have a collection of captions describing churches and a collection of Wiki pages describing churches. The intuition here is that these collections are examples of, e.g. the sorts of things people say in captions or in descriptions of churches. These collections can then be used to derive models of objects or scene types which in turn can be used to bias or focus multi-document summaries of new images of things of the same type.

In the talk I report results of work we have carried out to explore the hypothesis underlying this approach, namely that brief multi-document summaries generated as image captions by using models of object/scene types to bias or focus content selection will be superior to generic multi-document summaries generated for this purpose. I describe how we have constructed an image content taxonomy, how we have derived text collections for object/scene types, how we have derived object/scene type models from these collections and how these have been used in multi-document summarization. I also discuss the issue of how to evaluate the resulting captions and present preliminary results from one sort of evaluation.

Learning to Match Names Across Languages

Inderjeet Mani
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730, USA
imani@mitre.org

Alex Yeh
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730, USA
asy@mitre.org

Sherri Condon
The MITRE Corporation
7515 Colshire Drive
McLean, VA 22102, USA
scondon@mitre.org

Abstract

We report on research on matching names in different scripts across languages. We explore two trainable approaches based on comparing pronunciations. The first, a cross-lingual approach, uses an automatic name-matching program that exploits rules based on phonological comparisons of the two languages carried out by humans. The second, monolingual approach, relies only on automatic comparison of the phonological representations of each pair. Alignments produced by each approach are fed to a machine learning algorithm. Results show that the monolingual approach results in machine-learning based comparison of person-names in English and Chinese at an accuracy of over 97.0 F-measure.

1 Introduction

The problem of matching pairs of names which may have different spellings or segmentation arises in a variety of common settings, including integration or linking database records, mapping from text to structured data (e.g., phonebooks, gazetteers, and biological databases), and text to text comparison (for information retrieval, clustering, summarization, coreference, etc.). For named entity recognition, a name from a gazetteer or dictionary may be matched against text input; even within monolingual applications, the forms of these names might differ. In multi-document summarization, a name may have different forms across different sources. Systems that address this problem must be able to handle variant spellings, as well as abbreviations, missing or additional name parts, and different orderings of name parts.

In multilingual settings, where the names being compared can occur in different scripts in different languages, the problem becomes relevant to additional practical applications, including both multilingual information retrieval and machine translation. Here special challenges are posed by the fact that there usually aren't one-to-one correspondences between sounds across languages. Thus the name *Stewart*, pronounced / s t u w ə r t / in IPA, can be mapped to Mandarin "斯图尔特", which is Pinyin "si tu er te", pronounced /s i t^h u a ɻ t^h e/, and the name *Elizabeth* / I l I z ə b ɛ θ/ can map to "伊丽莎白", which is Pinyin "yi li sha bai", pronounced /I l I ʂ ɑ p aI/. Further, in a given writing system, there may not be a one-to-one correspondence between orthography and sound, a well-known case in point being English. In addition, there may be a variety of variant forms, including dialectical variants, (e.g., *Bourguiba* can map to *Abu Ruqayba*), orthographic conventions (e.g., Anglophone *Wasim* can map to Francophone *Ouassime*), and differences in name segmentation (*Abd Al Rahman* can map to *Abdurrahman*). Given the high degree of variation and noise in the data, approaches based on machine learning are needed.

The considerable differences in possible spellings of a name also call for approaches which can compare names based on pronunciation. Recent work has developed pronunciation-based models for name comparison, e.g., (Sproat, Tao and Zhai 2006) (Tao et al. 2006). This paper explores trainable pronunciation-based models further.

<table>
<tr><th></th><th>Roman</th><th>Chinese (Pinyin)</th><th>Alignment</th><th>Score</th></tr>
<tr><td><b>LEV</b></td><td>ashburton</td><td>ashenbodu</td><td>| a s h b u r t o n |
| a s h e n b o d u |</td><td>0.67</td></tr>
<tr><td><b>MLEV</b></td><td>ashburton</td><td>ashenbodu</td><td>| a s h - - b u r t o n |
| a s h e n b o - d u - |</td><td>0.72</td></tr>
<tr><td><b>MALINE</b></td><td>asVburton</td><td>aseCnpotu</td><td>| a sV - b < u r t o | n
| a s eC n p o - t u | -</td><td>0.48</td></tr>
</table>

Table 1: Matching "Ashburton" and "阿什伯顿"

Consider the problem of matching Chinese script names against their English (Pinyin) Romanizations. Chinese script has nearly 50,000 characters in all, with around 5,000 characters in use by the well-educated. However, there are only about 1,600 Pinyin syllables when tones are counted, and as few as 400 when they aren't. This results in multiple Chinese script representations for a given Roman form name and many Chinese characters that map to the same Pinyin forms. In addition, one can find multiple Roman forms for many names in Chinese script, and multiple Pinyin representations for a Chinese script representation.

In developing a multilingual approach that can match names from any pair of languages, we compare an approach that relies strictly on **monolingual** knowledge for each language, specifically, grapheme-to-phoneme rules for each language, with a method that relies on **cross-lingual** rules which in effect map between graphemic and/or phonemic representations for the specific pair of languages.

The monolingual approach requires finding data on the phonemic representations of a name in a given language, which (as we describe in Section 4) may be harder than finding more graphemic representations. But once the phonemic representation is found for names in a given language, then as one adds more languages to a system, no more work needs to be done in that given language. In contrast, with the cross-lingual approach, whenever a new language is added, one needs to go over all the existing languages already in the system and compare each of them with the new language to develop cross-lingual rules for each such language pair. The engineering of such rules requires bilingual expertise, and knowledge of differences between language pairs. The cross-lingual approach is thus more expensive to develop, especially for applications which require coverage of a large number of languages.

Our paper investigates whether we can address the name-matching problem without requiring such a knowledge-rich approach, by carrying out a comparison of the performance of the two approaches. We present results of large-scale machine-learning for matching personal names in Chinese and English, along with some preliminary results for English and Urdu.

2 Basic Approaches

2.1 Cross-Lingual Approach

Our cross-lingual approach (called MLEV) is based on (Freeman et al. 2006), who used a modified Levenshtein string edit-distance algorithm to match Arabic script person names against their corresponding English versions. The Levenshtein edit-distance algorithm counts the minimum number of insertions, deletions or substitutions required to make a pair of strings match. Freeman et al. (2006) used (1) insights about phonological differences between the two languages to create rules for equivalence classes of characters that are treated as identical in the computation of edit-distance and (2) the use of normalization rules applied to the English and transliterated Arabic names based on mappings between characters in the respective writing systems. For example, characters corresponding to low diphthongs in English are normalized as "w", the transliteration for the Arabic "و"character, while high diphthongs are mapped to "y", the transliteration for the Arabic "ي" character.

Table 1 shows the representation and comparison of a Roman-Chinese name pair (shown in the title) obtained from the Linguistic Data Consortium's LDC Chinese-English name pairs corpus (LDC 2005T34). This corpus provides name part pairs, the first element in English (Roman characters) and the second in Chinese characters, created by the LDC from Xinhua Newswire's proper name and who's who databases. The name part can be a first, middle or last name. We compare the English form of the name with a Pinyin Romanization of the Chinese. (Since the Chinese is being compared with English, which is toneless, the tone part of Pinyin is being ignored throughout this paper.) For this study, the Levenshtein edit-distance score (where a perfect match scores zero) is normalized to a similarity score as in (Freeman et al. 2006), where the score ranges from 0 to 1, with 1 being a perfect match. This edit-distance score is shown in the LEV row.

The MLEV row, under the Chinese Name column, shows an "Englishized" normalization of the Pinyin for *Ashburton*. Certain characters or character sequences in Pinyin are pronounced differently than in English. We therefore apply certain transforms to the Pinyin; for example, the following substitutions are applied at the start of a Pinyin syllable, which makes it easier for an English speaker to see how to pronounce it and renders the Pinyin more similar to English orthography: "u:" (umlaut "u") => "u", "zh" => "j", "c" => "ts", and "q" => "ch" (so the Pinyin "Qian" is more or less pronounced as if it were spelled as "Chian", etc.). The MLEV algorithm uses equivalence classes that allow "o" and "u" to match, which results in a higher score than the generic score using the LEV method.

2.2 Monolingual Approach

Instead of relying on rules that require extensive knowledge of differences between a language pair[2], the monolingual approach first builds phonemic representations for each name, and then aligns them. Earlier research by (Kondrak 2000) used dynamic programming to align strings of phonemes, representing the phonemes as vectors of phonological features, which are associated with scores to produce similarity values. His program ALINE includes a "skip" function in the alignment operations that can be exploited for handling epenthetic segments, and in addition to 1:1 alignments, it also handles 1:2 and 2:1 alignments. In this research, we made extensive modifications to ALINE to add the phonological features for languages like Chinese and Arabic and to normalize the similarity scores, producing a system called MALINE.

In Table 1, the MALINE row[3] shows that the English name has a palato-alveolar modification on the "s" (expressed as "sV"), so that we get the sound corresponding to "sh"; the Pinyin name inserts a centered "e" vowel, and devoices the bilabial plosive /b/ to /p/. There are actually sixteen different Chinese 'pinyinizations' of *Ashburton*, according to our data prepared from the LDC corpus.

3 Experimental Setup

3.1 Machine Learning Framework

Neither of the two basic approaches described so far use machine learning. Our machine learning framework is based on learning from alignments produced by either approach. To view the learning problem as one amenable to a statistical classifier, we need to generate labeled feature vectors so that each feature vector includes an additional class feature that can have the value 'true' or 'false.' Given a set of such labeled feature vectors as training data, the classifier builds a model which is then used to classify unlabeled feature vectors with the right labels.

A given set of attested name pairs constitutes a set of positive examples. To create negative pairs, we have found that randomly selecting elements that haven't been paired will create negative examples in which the pairs of elements being compared are so different that they can be trivially separated from the positive examples. The experiments reported here used the MLEV score as a threshold to select negatives, so that examples below the threshold are excluded. As the threshold is raised, the negative examples should become harder to discriminate from positives (with the harder problems mirroring some of the "confusable name" characteristics of the real-world name-matching problems this technology is aimed at). Positive examples below the threshold are also eliminated. Other criteria, including a MALINE score, could be used, but the MLEV scores seemed adequate for these preliminary experiments.

Raising the threshold reduces the number of negative examples. It is highly desirable to balance the number of positive and negative examples in training, to avoid the learning being

[2] As (Freeman et al., 2006) point out, these insights are not easy to come by: "These rules are based on first author Dr. Andrew Freeman's experience with reading and translating Arabic language texts for more than 16 years" (Freeman et al., 2006, p. 474).

[3] For the MALINE row in Table 1, the ALINE documentation explains the notation as follows: "every phonetic symbol is represented by a single lowercase letter followed by zero or more uppercase letters. The initial lowercase letter is the base letter most similar to the sound represented by the phonetic symbol. The remaining uppercase letters stand for the feature modifiers which alter the sound defined by the base letter. By default, the output contains the alignments together with the overall similarity scores. The aligned subsequences are delimited by '|' signs. The '<' sign signifies that the previous phonetic segment has been aligned with two segments in the other sequence, a case of compression/expansion. The '-' sign denotes a "skip", a case of insertion/deletion."

biased by a skewed distribution. However, when one starts with a balanced distribution of positive and negatives, and then excludes a number of negative examples below the threshold, a corresponding number of positive examples must also be removed to preserve the balance. Thus, raising the threshold reduces the size of the training data. Machine learning algorithms, however, can benefit from more training data. Therefore, in the experiments below, thresholds which provided woefully inadequate training set sizes were eliminated.

One can think of both the machine learning method and the basic name comparison methods (MLEV and MALINE) as taking each pair of names with a known label and returning a system-assigned class for that pair. Precision, Recall, and F-Measure can be defined in an identical manner for both machine learning and basic name comparison methods. In such a scheme, a threshold on the similarity score is used to determine whether the basic comparison match is a positive match or not. Learning the best threshold for a dataset can be determined by searching over different values for the threshold.

In short, the methodology employed for this study involves two types of thresholds: the MLEV threshold used to identify negative examples and the threshold that is applied to the basic comparison methods, MLEV and MALINE, to identify matches. To avoid confusion, the term *negative threshold* refers to the former, while the term *positive threshold* is used for the latter.

The basic comparison methods were used as baselines in this research. To be able to provide a fair basic comparison score at each negative threshold, we "trained" each basic comparison matcher at twenty different positive thresholds on the same training set used by the learner. For each negative threshold, we picked the positive threshold that gave the best performance on the training data, and used that to score the matcher on the same test data as used by the learner.

3.2 Feature Extraction

Consider the MLEV alignment in Table 1. It can be seen that the first three characters are matched identically across both strings; after that, we get an "e" inserted, an "n" inserted, a "b" matched identically, a "u" matched to an "o", a "r" deleted, a "t" matched to a "d", an "o" matched to a "u", and an "n" deleted. The match unigrams are thus "a:a", "s:s", "h:h", "-:e", "-:n", "b:b", "u:o", "r:-", "t:d", "o:u", and "n:-". Match bigrams were generated by considering any insertion, deletion, and (non-identical) substitution unigram, and noting the unigram, if any, to its left, prepending that left unigram to it (delimited by a comma). Thus, the match bigrams in the above example include "h:h,-:e", "-:e,-:n", "b:b,u:o", "u:o,r:-", "r:-,t:d", "t:d,o:u", "o:u,n:-".

These match unigram and match bigram features are generated from just a single MLEV match. The composite feature set is the union of the complete match unigram and bigram feature sets. Given the composite feature set, each match pair is turned into a feature vector consisting of the following features: string1, string2, the match score according to each of the basic comparison matchers (MLEV and MALINE), and the Boolean value of each feature in the composite feature set.

3.3 Data Set

Our data is a (roughly 470,000 pair) subset of the Chinese-English personal name pairs in LDC 2005T34. About 150,000 of the pairs had more than 1 way to pronounce the English and/or Chinese. For these, to keep the size of the experiments manageable from the point of view of training the learners, one pronunciation was randomly chosen as the one to use. (Even with this restriction, a minimum negative threshold results in over half a million examples). Chinese characters were mapped into Hanyu Pinyin representations, which are used for MLEV alignment and string comparisons. Since the input to MALINE uses a phonemic representation that encodes phonemic features in one or more letters, both Pinyin and English forms were mapped into the MALINE notation.

There are a number of slightly varying ways to map Pinyin into an international pronunciation system like IPA. For example, (Wikipedia 2006) and (Salafra 2006) have mappings that differ from each other and also each of these two sources have changed its mapping over time. We used a version of Salafra from 2006 (but we ignored the ejectives). For English, the CMU pronouncing dictionary (CMU 2008) provided phonemic representations that were then mapped into the MALINE notation. The dictionary had entries for 12% of our data set. For the names not in the CMU dictionary, a simple grapheme to phoneme script provided an approximate phonemic form. We did not use a monolingual mapping of Chinese characters (Mandarin pronunciation) into IPA because we did not find any.

Note that we could insist that all pairs in our dataset be distinct, requiring that there be exactly one match for each Roman name and exactly one match for each Pinyin name. This in our view is unrealistic, since large corpora will be skewed towards names which tend to occur frequently (e.g., international figures in news) and occur with multiple translations. We included attested match pairs in our test corpora, regardless of the number of matches that were associated with a member of the pair.

4 Results

A variety of machine learning algorithms were tested. Results are reported, unless otherwise indicated, using SVM Lite, a Support Vector Machine (SVM[4]) classifier[5] that scales well to large data sets.

Testing with SVM Lite was done with a 90/10 train-test split. Further testing was carried out with the weka SMO SVM classifier, which used built-in cross-validation. Although the latter classifier didn't scale to the larger data sets we used, it did show that cross-validation didn't change the basic results for the data sets it was tried on.

4.1 Machine Learning with Different Feature Sets

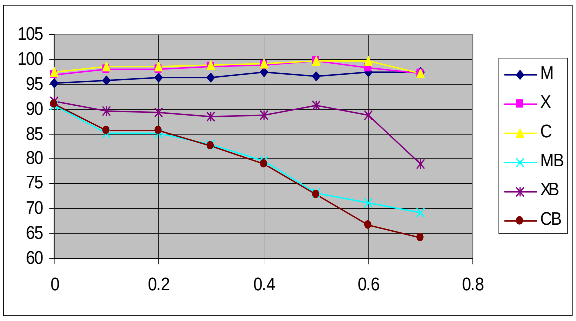


Figure 1: F-measure with Different Feature Sets

Figure 1 shows the F-measure of learning for monolingual features (M, based on MALINE), cross-lingual features (X, based on MLEV), and a combined feature set (C) of both types of features[6] at different negative thresholds (shown on the horizontal axis). Baselines are shown with the suffix B, e.g., the basic MALINE without learning is MB. When using both monolingual and cross-lingual features (C), the baseline (CB) is set to a system response of "true" only when both the MALINE and MLEV baseline systems by themselves respond "true". Table 2 shows the number of examples at each negative threshold and the Precision and Recall for these methods, along with baselines using the basic methods shown in square brackets.

The results show that the learning method (i) outperforms the baselines (basic methods), and (ii) the gap between learning and basic comparison widens as the problem becomes harder (i.e., as the threshold is raised).

For separate monolingual and cross-lingual learning, the increase in accuracy of the learning over the baseline (non-learning) results[7] was statistically significant at all negative thresholds except 0.6 and 0.7. For learning with combined monolingual and cross-lingual features (C), the increase over the baseline (non-learning) combined results was statistically significant at all negative thresholds except for 0.7.

In comparing the mono-lingual and cross-lingual learning approaches, however, the only statistically significant differences were that the cross-lingual features were more accurate than the monolingual features at the 0 to 0.4 negative thresholds. This suggests that (iii) the monolingual learning approach is as viable as the cross-lingual one as the problem of confusable names becomes harder.

However, using the combined learning approach (C) is better than using either one. Learning accuracy with both monolingual and cross-lingual features is statistically significantly better than learning with monolingual features at the 0.0 to 0.4 negative thresholds, and better than learning with cross-lingual features at the 0.0 to 0.2, and 0.4 negative thresholds.

[4] We used a linear kernel function in our SVM experiments; using polynomial or radial basis kernels did not improve performance.

[5] From svmlight.joachims.org.

[6] In Figure 1, the X curve is more or less under the C curve.

[7] Statistical significance between F-measures is not directly computable since the overall F-measure is not an average of the F-measures of the data samples. Instead, we checked the statistical significance of the increase in accuracy (accuracy is not shown for reasons of space) due to learning over the baseline. The statistical significance test was done by assuming that the accuracy scores were binomials that were approximately Gaussian. When the Gaussian approximation assumption failed (due to the binomial being too skewed), a looser, more general bound was used (Chebyshev's inequality, which applies to all probability distributions). All statistically significant differences are at the 1% level (2-sided).

4.2 Feature Set Analyses

The unigram features reflect common correspondences between Chinese and English pronunciation. For example, (Sproat, Tao and Zhai 2006) note that Chinese /l/ is often associated with English /r/, and the feature l:r is among the most frequent unigram mappings in both the MLEV and MALINE alignments. At a frequency of 103,361, it is the most frequent unigram feature in the MLEV mappings, and it is the third most frequent unigram feature in the MALINE alignments (56,780).

Systematic correspondences among plosives are also captured in the MALINE unigram mappings. The unaspirated voiceless Chinese plosives /p,t,k/ contrast with aspirated plosives /p^h,t^h,k^h/, whereas the English voiceless plosives (which are aspirated in predictable environments) contrast with voiced plosives /b,d,g/. As a result, English /b,d,g/ phonemes are usually transliterated using Chinese characters that are pronounced /p,t,k/, while English /p,t,k/ phonemes usually correspond to Chinese /p^h,t^h,k^h/. The examples of *Stewart* and *Elizabeth* in Section 1 illustrate the correspondence of English /t/ and Chinese / t^h/ and of English /b/ with Chinese /p/ respectively. All six of the unigram features that result from these correspondences occur among the 20 most frequent in the MALINE alignments, ranging in frequency from 23,602 to 53,535.

Negative Threshold	Examples	Monolingual (M)		Cross-Lingual (X)		Combined (C)	
		P	R	P	R	P	R
0	538,621	94.69 [90.6]	95.73 [91.0]	96.5 [90.0]	97.15 [93.4]	97.13 [90.8]	97.65 [91.0]
0.1	307,066	95.28 [87.1]	96.23 [83.4]	98.06 [89.2]	98.25 [89.9]	98.4 [87.6]	98.64 [84.1]
0.2	282,214	95.82 [86.2]	96.63 [84.4]	97.91 [88.4]	98.41 [90.3]	98.26 [86.7]	98.82 [84.7]
0.3	183,188	95.79 [80.6]	96.92 [85.3]	98.18 [86.3]	98.8 [90.7]	98.24 [80.6]	99.27 [84.8]
0.4	72,176	96.31 [77.1]	98.69 [82.3]	97.89 [91.8]	99.61 [86.2]	98.91 [77.1]	99.64 [80.9]
0.5	17,914	94.62 [64.6]	98.63 [84.3]	99.44 [89.4]	100.0 [91.9]	99.46 [63.8]	99.89 [84.7]
0.6	2,954	94.94 [66.1]	100 [77.0]	98.0 [85.2]	98.66 [92.8]	99.37 [61.3]	100.0 [73.1]
0.7	362	95.24 [52.8]	100 [100.0]	94.74 [78.9]	100.0 [78.9]	100.0 [47.2]	94.74 [100.0]

Table 2: Precision and Recall with Different Feature Sets (Baseline scores in square brackets)

4.3 Comparison with other Learners

To compare with other machine learning tools, we used the WEKA toolkit (from www.weka.net.nz). Table 3 shows the comparisons on the MLEV data for a fixed size at one threshold. Except for SVM Light, the results are based on 10-fold cross validation. The other classifiers appear to perform relatively worse at that setting for the MLEV data, but the differences in accuracy are not statistically significant even at the 5% level. A large contributor to the lack of significance is the small test set size of 66 pairs (10% of 660 examples) used in the SVM Light test.

4.4 Other Language Pairs

Some earlier experiments for Arabic-Roman comparisons were carried out using a Conditional Random Field learner (CRF), using the Carafe toolkit (from sourceforge.net/projects/carafe). The method computes its own Levenshtein edit-distance scores, and learns edit-distance costs from that. The scores obtained, on average, had only a .6 correlation with the basic comparison Levenshtein scores. However, these experiments did not return accuracy results, as ground-truth data was not specified for this task.

Several preliminary machine learning experiments were also carried out on Urdu-Roman comparisons. The data used were Urdu data extracted from a parallel corpus recently produced by the LDC (LCTL_Urdu.20060408). The results are shown in Table 4. Here a .55 MALINE score and a .85 MLEV score were used for selecting positive examples by basic comparison, and negative examples were selected at random. Here the MALINE method (row 1) using the weka SMO SVM made use of a threshold based on a MALINE score. In these earlier experiments, machine learning does not really improve the system performance (F-measure decreases with learning on one test and only increases by 0.1% on the other test). However, since these earlier experiments did not benefit from the use of different negative thresholds, there was no control over problem difficulty.

5 Related Work

While there is a substantial literature employing learning techniques for record linkage based on the theory developed by Fellegi and Sunter (1969), researchers have only recently developed applications that focus on name strings and that employ methods which do not require features to be independent (Cohen and Richman 2002). Ristad and Yianilos (1997) have developed a generative model for learning string-edit distance that learns the cost of different edit operations during string alignment. Bilenko and Mooney (2003) extend Ristad's approach to include gap penalties (where the gaps are contiguous sequences of mismatched characters) and compare this generative approach with a vector similarity approach that doesn't carry out alignment. McCallum et al. (2005) use Conditional Random Fields (CRFs) to learn edit costs, arguing in favor of discriminative training approaches and against generative approaches, based in part on the fact that the latter approaches "cannot benefit from negative evidence from pairs of strings that (while partially overlapping) should be considered dissimilar". Such CRFs model the conditional probability of a label sequence (an alignment of two strings) given a sequence of observations (the strings).

A related thread of research is work on automatic transliteration, where training sets are typically used to compute probabilities for mappings in weighted finite state transducers (Al-Onaizan and Knight 2002; Gao et al. 2004) or source-channel models (Knight and Graehl 1997; Li et al. 2004). (Sproat et al. 2006) have compared names from comparable and contemporaneous English and Chinese texts, scoring matches by training a learning algorithm to compare the phonemic representations of the names in the pair, in addition to taking into account the frequency distribution of the pair over time. (Tao et al. 2006) obtain similar results using frequency and a similarity score based on a phonetic cost matrix

The above approaches have all developed special-purpose machine-learning architectures to address the matching of string sequences. They take pairs of strings that haven't been aligned, and learn costs or mappings from them, and once trained, search for the best match given the learned representation

Positive Threshold	Examples	Method	P	R	F	Accuracy
.65	660	SVM Light	90.62	87.88	89.22	89.39
.65	660	WEKA SMO	80.6	83.3	81.92	81.66
.65	660	AdaBoost M1	84.9	78.5	81.57	82.27

Table 3: Comparison of Different Classifiers

Method	Positive Threshold	Examples	P	R	F
WEKA SMO	.55 (MALINE)	206 (MALINE)	84.8 [81.5]	86.4 [93.3]	85.6 [87.0]
WEKA SMO	.85 (MLEV)	584 (MLEV)	89.9 [93.2]	94.7 [91.2]	92.3 [92.2]

Table 4: Urdu-Roman Name Matching Results with Random Negatives (Baseline scores in square brackets)

Our approach, by contrast, takes pairs of strings *along with an alignment*, and using features derived from the alignments, trains a learner to derive the best match given the features. This offers the advantage of modularity, in that any type of alignment model can be combined with SVMs or other classifiers (we have preferred SVMs since they offer discriminative training). Our approach allows leveraging of any existing alignments, which can lead to starting the learning from a higher baseline and less training data to get to the same level of performance. Since the learner itself doesn't compute the alignments, the disadvantage of our approach is the need to engineer features that communicate important aspects of the alignment to the learner.

In addition, our approach, as with McCallum et al. (2005), allows one to take advantage of both positive and negative training examples, rather than positive ones alone. Our data generation strategy has the advantage of generating negative examples so as to vary the difficulty of the problem, allowing for more fine-grained performance measures. Metrics based on such a control are likely to be useful in understanding how well a name-matching system will work in particular applications, especially those involving confusable names.

6 Conclusion

The work presented here has established a framework for application of machine learning techniques to multilingual name matching. The results show that machine learning dramatically outperforms basic comparison methods, with F-measures as high as 97.0 on the most difficult problems. This approach is being embedded in a larger system that matches full names using a vetted database of full-name matches for evaluation.

So far, we have confined ourselves to minimal feature engineering. Future work will investigate a more abstract set of phonemic features. We also hope to leverage ongoing work on harvesting name pairs from web resources, in addition applying them to less commonly taught languages, as and when appropriate resources for them become available.

References

Al-Onaizan, Y. and K. Knight, K. 2002. Machine Transliteration of Names in Arabic Text. Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages.

Bilenko, M. and Mooney, R.J. 2003. Adaptive duplicate detection using learnable string similarity measures. In *Proc. of SIGKDD-2003.*

CMU. 2008. The CMU Pronouncing nary. ftp://ftp.cs.cmu.edu/project/speech/dict/

Cohen, W. W., and Richman, J. 2002. Learning to match and cluster large high-dimensional data sets for data integration. In *Proceedings of The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002).*

Fellegi, I. and Sunter, A. 1969. A theory for record linkage. *Journal of the American Statistical Society,* 64:1183-1210, 1969.

Freeman, A., Condon, S. and Ackermann, C. 2006. Cross Linguistic Name Matching in English and Arabic. *Proceedings of HLT.*

Gao, W., Wong, K., and Lam, W. 2004. Phoneme-based transliteration of foreign names for OOV problem. In *Proceedings of First International Joint Conference on Natural Language Processing.*

Kondrak, G. 2000. A New Algorithm for the Alignment of Phonetic Sequences. Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL 2000), 288-295.

Knight, K. and Graehl, J., 1997. Machine Transliteration, In *Proceedings of the Conference of the Association for Computation Linguistics (ACL).*

Li, H., Zhang, M., & Su, J. 2004. A joint source-channel model for machine transliteration. In *Proceedings of Conference of the Association for Computation Linguistics (ACL).*

McCallum, A., Bellare, K. and Pereira, F. 2005. A Conditional Random Field for Discriminatively-trained Finite-state String Edit Distance. *Conference on Uncertainty in AI (UAI).*

Ristad, E. S. and Yianilos, P. N. 1998. Learning string edit distance. *IEEE Transactions on Pattern Recognition and Machine Intelligence.*

Salafra. 2006. http://www.safalra.com /science /linguistics/pinyin-pronunciation/

Sproat, R., Tao, T. and Zhai, C. 2006. Named Entity Transliteration with Comparable Corpora. In *Proceedings of the Conference of the Association for Computational Linguistics*. New York.

Tao, T., Yoon, S. Fister, A., Sproat, R. and Zhai, C. 2006. Unsupervised Named Entity Transliteration Using Temporal and Phonetic Correlation. In *Proceedings of the ACL Empirical Methods in Natural Language Processing Workshop*.

Wikipedia. 2006. http://en.wikipedia.org/wiki/Pinyin

Automatic Construction of Domain-specific Dictionaries on Sparse Parallel Corpora in the Nordic Languages

Sumithra Velupillai
DSV/KTH-Stockholm University
SE-164 40 Kista
Sweden
sumithra@dsv.su.se

Hercules Dalianis [1, 2]
[1] DSV/KTH-Stockholm University
SE-164 40 Kista
Sweden
[2] Euroling AB
Igeldammsgatan 22c
112 49 Stockholm, Sweden
hercules@dsv.su.se

Abstract

Hallå Norden is a web site with information regarding mobility between the Nordic countries in five different languages; Swedish, Danish, Norwegian, Icelandic and Finnish. We wanted to create a Nordic cross-language dictionary for the use in a cross-language search engine for Hallå Norden. The entire set of texts on the web site was treated as one multilingual parallel corpus. From this we extracted parallel corpora for each language pair. The corpora were very sparse, containing on average less than 80 000 words per language pair. We have used the Uplug word alignment system (Tiedemann 2003a), for the creation of the dictionaries. The results gave on average 213 new dictionary words (frequency > 3) per language pair. The average error rate was 16 percent. Different combinations with Finnish had a higher error rate, 33 percent, whereas the error rate for the remaining language pairs only yielded on average 9 percent errors. The high error rate for Finnish is possibly due to the fact that the Finnish language belongs to a different language family. Although the corpora were very sparse the word alignment results for the combinations of Swedish, Danish, Norwegian and Icelandic were surprisingly good compared to other experiments with larger corpora.

1 Introduction

Hallå Norden (Hello Scandinavia) is a web site with information regarding mobility between the Nordic countries and is maintained by the Nordic Council. Mobility information concerns issues such as how employment services, social services, educational systems etc. work in the different countries. The web site has information in five different languages; Swedish, Danish, Norwegian, Icelandic and Finnish. In this paper Nordic languages are defined as Swedish, Danish, Norwegian, Icelandic and Finnish. Scandinavian languages are defined as the Nordic languages excluding Finnish.

The texts on the web site were almost parallel and there were also ten minimal dictionaries with on average 165 words available for the different languages. The dictionaries consisted of domain-specific words regarding mobility information in the Nordic countries. The Nordic Council wanted to extend the dictionaries so they would cover a larger part of the specific vocabulary, in order to help the people in the Nordic countries to find and learn the concepts in their neighboring countries.

The entire set of texts on the web site was treated as one multilingual parallel corpus. From this we extracted parallel corpora for each language pair. We discovered, as expected, that the corpora were very sparse, containing on average less than 80 000 words per language pair. We needed to construct 10 different dictionaries and therefore we processed 10 pairs of parallel text sets. We have used the Uplug word alignment system (Tiedemann 2003a), for the creation of the dictionaries. The system and motivation for the choice of system is further discussed in Section 2.1.

We also discovered that the texts were not completely parallel. Therefore, we made a small experiment on attempting to enhance the results by deleting texts that were not parallel. Multilingual parallel corpora covering all Nordic languages are very rare. Although the corpora created in this work are domain-specific, they are an important contribution for further research on Nordic multilingual issues. Moreover, many large governmental, industrial or similar web sites that contain information in several languages may profit from compiling multilingual dictionaries automatically in order to enhance their search engines and search results.

In this project, our two main goals were to compile parallel corpora covering the Nordic languages, and to evaluate the results of automatically creating dictionaries using an existing tool with basic settings, in order to find out where more work would need to be done and where performance is actually acceptable. We have limited the work by only testing one system (Uplug) with basic settings. Our experiments and results are described in further detail in the following sections. Conclusions and future work are discussed in the final section.

2 Related Work

Word alignment systems have been used in previous research projects for automatically creating dictionaries. In Charitakis (2007) Uplug was used for aligning words in a Greek-English parallel corpus. The corpus was relatively sparse, containing around 200 000 words for each language, downloaded from two different bilingual web sites. A sample of 498 word pairs from Uplug were evaluated by expert evaluators and the result was 51 percent correctly translated words (frequency > 3). When studying high frequent word pairs (>11), there were 67 percent correctly translated words. In Megyesi & Dahlqvist (2007) an experiment is described where they had 150 000 words in Swedish and 126 000 words in Turkish that gave 69 percent correct translations (Uplug being one of the main tools used). In this work the need for parallel corpora in different language combinations is also discussed.

The ITools' suite for word alignment that was used in Nyström et al (2006) on a medical parallel corpus, containing 174 000 Swedish words and 153 000 English words, created 31 000 word pairs with 76 percent precision and 77 percent recall. In this work the word alignment was produced interactively.

A shared task on languages with sparse resources is described in Martin et al (2005). The language pairs processed were English-Inuktitut, Romanian-English and English-Hindi, where the English-Inuktitut parallel corpus contained around 4 million words for English and 2 millions words for Inuktitut. English-Hindi had less words, 60 000 words and 70 000 words respectively. The languages with the largest corpora obtained best word alignment results, for English-Inuktitut over 90 percent precision and recall and for English-Hindi 77 percent precision and 68 percent recall. One conclusion from the shared task was that it is worth using additional resources for languages with very sparse corpora improving results with up to 20 percent but not for the languages with more abundant corpora such as for instance English-Inuktitut.

2.1 Word Alignment: Uplug

We have chosen to use the Uplug word alignment system since it is a non-commercial system which does not need a pre-trained model and is easy to use. It is also updated continuously and incorporates other alignment models, such as GIZA++ (Och & Ney 2003). We did not want to evaluate the performance of different systems in the work presented here, but rather evaluate the performance of only one system applied on different language combinations and on sparse corpora. Evaluating the performance of different systems is an important and interesting research problem, but is left for future work. An evaluation of two word alignment systems Plug (Uplug) and Arcade is described in Ahrenberg et al (2000).

The Uplug system implements a word alignment process that combines different statistical measures for finding word alignment candidates and is fully automatic. It is also possible to combine statistical measures with linguistic information, such as part-of-speech tags. In the preprocessing steps the corpora are converted to an xml-format and they are also sentence aligned.

We have chosen to use basic settings for all corpora in the different language pairs, in order to evaluate the effect of this. The default word alignment settings in Uplug works in the following way:

- create basic clues (Dice and LCSR)
- run GIZA++ with standard settings (trained on plain text)

Language pair	No. texts	No. words	Word distribution, first language in language pair, %
sw-da	191	83871	49.2
sw-no	133	62554	49.7
sw-fi	196	73933	57.6
sw-ice	187	82711	48.5
da-no	156	68777	50.2
da-fi	239	84194	58.4
da-ice	232	97411	49.5
no-fi	156	58901	58.2
no-ice	145	64931	49.6
Average	182	75254	52.3

Table 1: General corpora information, initial corpora

- learn clues from GIZA's Viterbi alignments
- "radical stemming" (take only the 3 initial characters of each token) and run GIZA++ again
- align words with existing clues
- learn clues from previous alignment
- align words again with all existing clues[1]

This approach is called the *clue alignment* approach and is described further in Tiedemann (2003b). In the work presented here, we have not included any linguistic information, as we wanted to evaluate the performance of applying the system on sparse, raw, unprocessed corpora for different (Nordic) language pairs, using default settings.

3 Experiments and Results

For the project presented in this paper we wanted to see if it was possible to create domain-specific dictionaries on even smaller corpora. (compared to the ones described in Section 2) for all the Nordic language pairs. We did not have the possibility to evaluate the results for Icelandic-Finnish, since we did not find any evaluator having knowledge in both Icelandic and Finnish. Therefore we present the results for the remaining nine language pairs. In total we had four evaluators for the other language combinations. Each evaluator evaluated those language pairs she or he had fluent or near-fluent knowledge in. The domain was very restricted containing only words about mobility between the Nordic countries.

The Scandinavian languages are closely related. Swedish, Danish, and Norwegian are comprehensible for Scandinavians. A typical Swede will for instance understand written and to a certain degree spoken Danish, but is not able to speak Danish. Typical Swedes will, for instance, have a passive understanding of Danish (and vice versa for the other languages). Finnish on the other hand belongs to the Finno-Ugric group of the Uralic languages, while the Scandinavian languages are North-Germanic Indo-European languages. We wanted to investigate if, and how, these differences affect the word alignment results. We also wanted to experiment with different frequency thresholds, in order to see if this would influence the results.

The first step was to extract the web pages from the web site and obtain the web pages in plain text format. We obtained help for that work from Euroling AB,[2] our contractor.

In Table 1 we show general information about the corpora. We see that the distribution of words is even for the Scandinavian languages, but not for the combinations with Finnish. It is interesting to observe that Finnish has fewer word tokens than the Scandinavian languages.

All Nordic languages, both Scandinavian and Finnish, have very productive word compounding. In Finnish word length is longer, on average,

[1] Steps taken from the Quickstart guidelines for the Uplug system, which can be downloaded here: http://uplug.sourceforge.net/

[2] See: http://www.euroling.se/

and the number of words per clause lower, on average, due to its extensive morphology.

In Dalianis et al (2007) lemmatizing the text set before the alignment process did not improve results. In the work presented here, we have also made some experiments on lemmatizing the corpora before the alignment process. We have used the CST lemmatizer[3] for the Scandinavian languages and Fintwol[4] for Finnish. Unfortunately, the results were not improved. The main reason for the decrease in performance is probably due to the loss of sentence formatting during the lemmatization process. The sentence alignment is a crucial preprocessing step for the word alignment process, and a lot of the sentence boundaries were lost in the lemmatization process. However, the resulting word lists from Uplug have been lemmatized using the same lemmatizers, in order to obtain normalized dictionaries.

The corpora were to some extent non-parallel containing some extra non-parallel paragraphs. We found that around five percent of the corpora were non-parallel. In order to detect non-parallel sections we have used a simpler algorithm than in for instance Munteanu & Marcu (2006). The total number of paragraphs and sentences in each parallel text pair were counted. If the total number for each language in some language pair differed more than 20 percent these files were deleted. The refined corpora have been re-aligned with Uplug and evaluated. In Table 2 we show the general information for the refined corpora.

3.1 Evaluation

Our initial plan was to use the manually constructed dictionaries from the web site as an evaluation resource, but the words in these dictionaries were rare in the corpus. Therefore we used human evaluators to evaluate the results from Uplug.

The results from the Uplug execution gave on average 213 new dictionary words (frequency > 3) per language, see Table 3. The average error rate[5] was 16 percent. We delimited the word amount by removing words shorter than six characters, and also multiword expressions[6] from the resulting word lists. The six character strategy is efficient for the Scandinavian languages as an alternative to stop word removal (Dalianis et al 2003) since the Scandinavian languages, as well

Language pair	No. parallel texts	Deleted files, %	No. words, parallel	Word distribution, first language in language pair, %
sw-da	179	6.3	78356	49.7
sw-no	128	3.8	59161	49.8
sw-fi	189	3.6	69525	58.1
sw-ice	175	5.9	76056	48.3
da-no	147	5.8	64946	50.2
da-fi	222	7.1	77849	58.6
da-ice	210	3.4	89093	49.0
no-fi	145	7.1	55409	58.3
no-ice	130	2.1	59622	49.0
Average	169	5.0	70002	52.3

Table 2: General corpora information, refined parallel corpora (non-parallel texts deleted)

[3] See: http://cst.dk/download/cstlemma/current/doc/

[4] See: http://www2.lingsoft.fi/cgi-bin/fintwol

[5] The error rate is in this paper defined as the percentage of wrongly generated entries compared to the total number of generated entries.

[6] A multiword expression is in this paper defined as words (sequences of characters, letters or digits) separated by a blank or a hyphen.

as Finnish, mostly produce compounds that are formed into one word (i.e. without blanks or hyphens). In Tiedemann (2008), a similar strategy of removing words with a word length shorter than five characters was carried out but in that case for English, Dutch and German.

	Initial		Deleting non-parallel	
Language pair	No. dictionary words	Erroneous translations, %	No. dictionary words	Erroneous translations, %
sw-da	322	7.1	305	7.2
sw-no	269	6.3	235	9.4
sw-fi	138	29.0	133	34.6
sw-ice	151	18.5	173	16.2
da-no	322	3.7	304	4.3
da-fi	169	34.3	244	33.2
da-ice	206	6.8	226	10.2
no-fi	185	27.6	174	30.0
no-ice	159	14.5	181	14.4
Average	213	16.4	219	16.1

Table 3: Produced dictionary words and error rate

Different combinations with Finnish had a higher error rate, 30 percent, whereas the error rate for the combinations of the Scandinavian languages only yielded on average 9 percent errors.

The high error rate for Finnish is possibly due to the fact that the Finnish language belongs to a different language family. We can see the same phenomena for Greek (Charitakis, 2007) and Turkish (Megyesi & Dahlqvist, 2007) combined with English and Swedish respectively, with 33 and 31 percent erroneously translated words.

However, one might expect even higher error rates due to the differences in the different language pairs (and the sparseness of the data). Finnish has free word order and is typologically very different from the Scandinavian languages, and the use of form words differs between the languages. On the other hand, both Finnish and the Scandinavian languages produce long, complex compounds somewhat similarly, and the word order in Finnish share many features with the word order in the Scandinavian languages. One important aspect is the cultural similarities that the languages share.

The main errors that were produced for the combinations of Finnish and the Scandinavian languages consisted of either errors with particles or compounds where the head word or attribute were missing in the Finnish alignment. For instance, the Swedish word *invånare* (inhabitant) was aligned with the Finnish word *asukasluku* (number of inhabitants). Another error which was produced for all combinations with Finnish was *lisätieto* (more information) which was aligned with *ytterligere* (additional, more) in Norwegian (and equivalent words in Swedish and Danish), an example of an error where the head word is missing. Many texts had sentences pointing to further information, which might explain this type of error.

The lemmatizers produced some erroneous word forms. In Dalianis & Jongejan (2006) the CST lemmatizer was evaluated and reported an average error rate of nine percent. Moreover, since the lemmatization process is performed on the resulting word lists, and not within the original context in which the words occur, the automatic lemmatization is more difficult for the two lemmatizers used in this project. These errors have not been included in our evaluation since they are not produced by the Uplug alignment procedure.

We can also see in Table 3 that deleting non-parallel texts using our simple algorithm did not improve the overall results significantly. Perhaps our simple algorithm was too coarse for these corpora. The texts were in general very short and simple frequency information on paragraph and sentence amounts might not have captured non-parallel fragments on such texts.

The produced dictionary words were of high domain-specific quality. The majority of the correct and erroneous word pairs were covered by both the initial and the refined corpus. Deleting non-parallel texts produced some new, valuable words that were not included in the initial results. However, since these dictionaries were generally smaller, this did not improve the overall results, and the error rate was somewhat higher for most language pairs. Improved dictionary in this work means as many word pairs as possible with domain-specific significance.

Since the texts were about different country-specific issues they could contain sections in another language (names of ministries, offices etc). This produced some errors in the alignment results. These errors might have been avoided by applying a language checker while processing the texts.

The errors for the Scandinavian languages were also mainly of the same type, and mostly due to the fact that the texts were not completely parallel, or due to form words or compounds. For instance, the Swedish word *exempelvis* (for example) was aligned with the Norwegian word *eksempel* (example), which was counted as an error, but which, in its context, is not completely erroneous.

Even at a relatively low frequency threshold the results were very good for the Scandinavian languages. We tried to increase the frequency threshold in order to see if this would improve the results for Finnish, which it unfortunately did not. However, as stated above, the errors were mainly of the same type, and probably constant over different frequencies. We also see that for Icelandic, unlike the other languages, deleting non-parallel fragments yielded larger dictionaries. Uplug produced more multiword units for the initial corpora containing Icelandic, single word pairs were more frequent in the refined corpus. However, the overall results were not improved.

4 Conclusions and Future Work

Although the corpora were very sparse the word alignment results for Swedish-Danish, Swedish-Norwegian and Danish-Norwegian were surprisingly good with on average 93.1 percent correct results. The results for Finnish were worse with on average only 67.4 percent correct results.

However, as discussed above, the main errors were of the same type. Creating dictionaries for non-related languages might need more elaborate alignment approaches. In the special case of Finnish combined with one (or several) of the Scandinavian languages, simple preprocessing steps might improve the results. For instance, removing stop words before running the corpora through a word alignment system might handle the errors where particles and form words are included. Also, tagging the corpora with part-of-speech tags and lemmatizing as a preprocessing step might improve results.

An important aspect of automatically creating multilingual dictionaries is the need for preprocessing tools covering all languages. This is often difficult to obtain, and different tools use different formatting and tagging schemes. Moreover, they might differ in robustness, which also affects the end results. In this project, we encountered such problems during the lemmatization process for instance, but we did not have the opportunity to explore and evaluate alternative tools. In the future, evaluating the performance of the preprocessing steps might be desirable.

Evaluating translated words is not easy. Many words may be related without being direct translations. Manual evaluation has the advantage of taking such issues into account, but this also means that the results might differ depending on the evaluator. Furthermore, evaluating translations without contextual information is problematic. Also, the criteria for judging a translation as correct or not depend on the goal for the use of the word lists. For instance, the errors for the combinations with Finnish might not be problematic in a real-world search engine setting, depending on which demands there are on the search results. The errors produced in the work presented here would probably yield acceptable search results. Such user and search engine result aspects have not been evaluated here, but are interesting research questions for future work.

The Nordic languages are highly inflectional. Combining compound splitting and lemmatizing before the alignment process might improve the results. Especially compound splitting could probably handle the errors produced for the combinations of Finnish with the Scandinavian languages. Cross-combining the different language pairs might enhance the results and create more specific and errorless dictionaries. Other word alignment systems should also be tested, in order to compare different approaches and their results. Perhaps results from different systems could also be combined, in order to produce more extensive dictionaries. Furthermore, other approaches to detect non-parallel fragments should be investigated.

Finding the boundary for the minimum size of parallel corpora in order to obtain acceptable dictionaries is also an interesting research issue which should be explored.

Automatically creating multilingual dictionaries is not trivial. Many aspects need to be considered. Especially, the final use of the produced results influences both the preprocessing steps required and the evaluation of the results. Also, the languages in consideration affect the steps that need to be made. However, in this paper we have shown that using state-of-the-art tools on sparse, raw, unprocessed domain-specific corpora in both related and non-related languages yield acceptable and even commendable results. Depending on the purposes for the use of the dictionaries, simple adjustments would probably yield even better results.

In a real-world setting, parallel (or near-parallel) corpora covering several (small) languages are difficult to obtain and compile. Most resources are found on the Internet, and the quality of the corpora may vary depending on many aspects. Formatting, translations, text length and style may differ considerably depending on the type of texts. Freely available text sets for small languages are often sparse. Despite this, we have shown that it is possible to compile valuable resources from available data.

There are very few sources of dictionaries covering the Nordic language pairs. The created corpora will be made publicly available for further research and evaluation.

References

Ahrenberg, L., M. Merkel, A. Sågvall Hein and J. Tiedemann 2000. Evaluation of word alignment systems. Lars Ahrenberg, Magnus Merkel, Anna Sågvall Hein and Jörg Tiedemann. Proceedings of the Second International Conference on Linguistic Resources and Evaluation (LREC-2000), Athens, Greece, 31 May - 2 June, 2000, Volume III: 1255-1261.

Charitakis, K. 2007. Using parallel corpora to create a Greek-English dictionary with Uplug, in Proc. 16th Nordic Conference on Computational Linguistics - NODALIDA ‘07.

Dalianis, H. and B. Jongejan 2006. Hand-crafted versus Machine-learned Inflectional Rules: the Euroling-SiteSeeker Stemmer and CST's Lemmatiser, in Proc. of the International Conference on Language Resources and Evaluation, LREC 2006.

Dalianis, H., M. Rimka and V. Kann 2007. Using Uplug and SiteSeeker to construct a cross language search engine for Scandinavian. Workshop: The Automatic Treatment of Multilinguality in Retrieval, Search and Lexicography, Copenhagen, April 2007.

Dalianis, H., M. Hassel, J. Wedekind, D. Haltrup, K. de Smedt and T.C. Lech. 2003. Automatic text summarization for the Scandinavian languages. In Holmboe, H. (ed.) Nordisk Sprogteknologi 2002: Årbog for Nordisk Språkteknologisk Forskningsprogram 2000-2004, pp. 153-163. Museum Tusculanums Forlag.

Martin, J and R. Mihalcea and T. Pedersen. 2005. Word Alignment for Languages with Scarce Resources. Proceedings of the ACL 2005 Workshop on *Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, Ann Arbor, MI, June 2005.

Megyesi, B. and B. Dahlqvist, 2007. The Swedish-Turkish Parallel Corpus and Tools for its Creation, in Proc. 16th Nordic Conference on Computational Linguistics - NODALIDA ‘07.

Munteanu, D.S. and D. Marcu 2006. Extracting Parallel Sub-sentential Fragments from Non-parallel Corpora. ACL ’06: Proceedings of the 21st International Conference on Computational Linguistics, pp. 81-88, Sydney, Australia.

Nyström, M., M. Merkel, L. Ahrenberg, P. Zweigenbaum, H. Petersson and H. Åhlfeldt. 2006. Creating a Medical English-Swedish Dictionary using Interactive Word Alignment, in BMC medical informatics and decision making, 6:35.

Franz Josef Och, Hermann Ney. A Systematic Comparison of Various Statistical Alignment Models, Computational Linguistics, volume 29, number 1, pp. 19-51 March 2003.

Tiedemann, J. 2003a. Recycling Translations: Extraction of Lexical Data from Parallel Corpora and their Application in Natural Language Processing. Acta Universitatis Upsaliensis: Studia linguistica upsaliensia, ISSN 1652-1366, ISBN 91-554-5815-7.

Tiedemann, J. 2003b. Combining clues for word alignment. In *Proceedings of the Tenth Conference on European Chapter of the Association For Computational Linguistics - Volume 1* (Budapest, Hungary, April 12 - 17, 2003). European Chapter Meeting of the ACL. Association for Computational Linguistics, Morristown, NJ, 339-346. DOI= http://dx.doi.org/10.3115/1067807.1067852.

Tiedemann, J. 2008. Synchronizing Translated Movie Subtitles. In the Proceedings of the Sixth International Conference on Language Resources and Evaluation, LREC 2008, Marrakech, Morocco, May 28-30, 2008.

Graph-Based Keyword Extraction for Single-Document Summarization

Marina Litvak
Department of
Information System Engineering
Ben-Gurion University of the Negev
Beer-Sheva 84105, Israel
litvakm@bgu.ac.il

Mark Last
Department of
Information System Engineering
Ben-Gurion University of the Negev
Beer-Sheva 84105, Israel
mlast@bgu.ac.il

Abstract

In this paper, we introduce and compare between two novel approaches, supervised and unsupervised, for identifying the keywords to be used in extractive summarization of text documents. Both our approaches are based on the graph-based syntactic representation of text and web documents, which enhances the traditional vector-space model by taking into account some structural document features. In the supervised approach, we train classification algorithms on a summarized collection of documents with the purpose of inducing a keyword identification model. In the unsupervised approach, we run the HITS algorithm on document graphs under the assumption that the top-ranked nodes should represent the document keywords. Our experiments on a collection of benchmark summaries show that given a set of summarized training documents, the supervised classification provides the highest keyword identification accuracy, while the highest F-measure is reached with a simple degree-based ranking. In addition, it is sufficient to perform only the first iteration of HITS rather than running it to its convergence.

1 Introduction

Document summarization is aimed at all types of electronic documents including HTML files with the purpose of generating the summary - main document information expressed in ”a few words”.

In this paper, we introduce and compare between two approaches: supervised and unsupervised, for the cross-lingual keyword extraction to be used as the first step in extractive summarization of text documents. Thus, according to our problem statement, the keyword is a word presenting in the document summary.

The supervised learning approach for keywords extraction was first suggested in (Turney, 2000), where parametrized heuristic rules were combined with a genetic algorithm into a system - GenEx - that automatically identified keywords in a document.

For both our approaches, we utilize a graph-based representation for text documents. Such representations may vary from very simple, syntactic ones like words connected by edges representing co-occurrence relation (Mihalcea and Tarau, 2004) to more complex ones like concepts connected by semantic relations (Leskovec et al., 2004). The main advantage of a syntactic representation is its language independency, while the semantic graphs representation provide new characteristics of text such as its captured semantic structure that itself can serve as a document surrogate and provide means for document navigation. Authors of (Leskovec et al., 2004) reduce the problem of summarization to acquiring machine learning models for mapping between the document graph and the graph of a summary. Using deep linguistic analysis, they extract sub-structures (subjectpredicateobject triples) from document semantic graphs in order to get a summary. Contrary to (Leskovec et al., 2004), both our approaches work with a syntactic representation that does not require almost any language-specific linguistic processing. In

Coling 2008: Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization, pages 17–24
Manchester, August 2008

this paper, we perform experiments with directed graphs, where the nodes stand for words/phrases and the edges represent syntactic relationships between them, meaning ̈followed by ̈ (Schenker et al., 2005).

Some of the most successful approaches to extractive summarization utilize supervised learning algorithms that are trained on collections of ”ground truth” summaries built for a relatively large number of documents (Mani and Maybury, 1999). However, in spite of the reasonable performance of such algorithms they cannot be adapted to new languages or domains without training on each new type of data. Our first approach also utilizes classification algorithms, but, thanks to the language-independent graph representation of documents, it can be applied to various languages and domains without any modifications of the graph construction procedure (except for the technical upgrade of implementation for multilingual processing of text, like reading Unicode or language-specific encodings, etc.) (Markov et al., 2007; Last and Markov, 2005). Of course, as a supervised approach it requires high-quality training labeled data.

Our second approach uses a technique that does not require any training data. To extract the summary keywords, we apply a ranking algorithm called HITS (Kleinberg, 1999) to directed graphs representing source documents. Authors of (Mihalcea and Tarau, 2004) applied the PageRank algorithm (Brin and Page, 1998) for keyword extraction using a simpler graph representation (undirected unweighted graphs), and show that their results compare favorably with results on established benchmarks of manually assigned keywords. (Mihalcea and Tarau, 2004) are also using the HITS algorithm for automatic sentence extraction from documents represented by graphs built from sentences connected by similarity relationships. Since we work with directed graphs, HITS is the most appropriate algorithm for our task as it takes into account both in-degree and out-degree of nodes. We show in our experiments that running HITS till convergence is not necessary, and initial weights that we get after the first iteration of algorithm are good enough for rank-based extraction of summary keywords. Another important conclusion that was infered from our experimental results is that, given the training data in the form of annotated syntactic graphs, supervised classification is the most accurate option for identifying the salient nodes in a document graph, while a simple degree-based ranking provides the highest F-measure.

2 Document representation

Currently, we use the ”simple” graph representation defined in (Schenker et al., 2005) that holds unlabeled edges representing order-relationship between the the words represented by nodes. The stemming and stopword removal operations of basic text preprocessing are done before graph building. Only a single vertex for each distinct word is created even if it appears more than once in the text. Thus each vertex label in the graph is unique. If a word ***a*** immediately precedes a word ***b*** in the same sentence somewhere in the document, then there is a directed edge from the vertex corresponding to term ***a*** to the vertex corresponding to term ***b***. Sentence terminating punctuation marks (periods, question marks, and exclamation points) are taken by us into account and an edge is not created when these are present between two words. This definition of graph edges is slightly different from co-occurrence relations used in (Mihalcea and Tarau, 2004) for building undirected document graphs, where the order of word occurrence is ignored and the size of the co-occurrence window is varied between 2 and 10. Sections defined for HTML documents are: *title*, which contains the text related to the document's title and any provided keywords (meta-data) and *text*, which comprises any of the readable text in the document. This simple representation can be extended to many different variations like a semantic graph where nodes stand for concepts and edges represent semantic relations between them or a more detailed syntactic graph where edges and nodes are labeled by significant information like frequency, location, similarity, distance, etc. The syntactic graph-based representations were shown in (Schenker et al., 2005) to outperform the classical vector-space model on several clustering and classification tasks. We choose the ”simple” representation as a representation that saves processing time and memory resources as well as gives nearly the best results for the two above text mining tasks.

3 Keywords extraction

In this paper, we deal with the first stage of extractive summarization where the most salient words (”keywords”) are extracted in order to generate a summary. Since each distinct word in a text is represented by a node in the document graph, the keywords extraction problem is reduced to the salient nodes extraction in graphs.

3.1 The Supervised approach

In this approach, we try to identify the salient nodes of document graphs by training a classification algorithm on a repository of summarized documents such as (DUC, 2002) with the purpose of inducing a keyword identification model. Each node of every document graph belongs to one of two classes: YES if the corresponding word is included in the document extractive summary and NO otherwise. We consider the graph-based features (e.g., degree) characterizing graph structure as well as statistic-based features (Nobata et al., 2001) characterizing text content represented by a node. The complete list of features, along with their formal definitions, is provided below:

- **In Degree** - number of incoming edges
- **Out Degree** - number of outcoming edges
- **Degree** - total number of edges
- **Frequency** - *term frequency* of word represented by node[1]
- **Frequent words distribution** $\in \{0,1\}$, equals to 1 iff **Frequency**$\geq$*threshold*[2]
- **Location Score** - calculates an average of location scores between all sentences[3] containing the word N represented by node (denote these sentences as *S(N)*):

$$Score\left(N\right) = \frac{\sum_{S_i \in S(N)} Score\left(S_i\right)}{\left|S\left(N\right)\right|}$$

- **Tfidf Score** - calculates the *tf-idf* score (Salton, 1975) of the word represented by node[4].
- **Headline Score** $\in \{0,1\}$, equals to 1 iff document headline contains word represented by node.

3.2 The Unsupervised approach

Ranking algorithms, such as Kleinberg's HITS algorithm (Kleinberg, 1999) or Google's PageRank (Brin and Page, 1998) have been elaborated and used in Web-link analysis for the purpose of optimizating the search performance on the Web. These algorithms recursively assign a numerical weight to each element of a hyperlinked set of documents, determining how important each page is. A hyperlink to a page counts as a vote of support. A page that is linked to by many important pages (with high rank) receives a high rank itself. A similar idea can be applied to lexical or semantic graphs extracted from text documents, in order to extract the most significant blocks (words, phrases, sentences, etc.) for the summary (Mihalcea and Tarau, 2004; Mihalcea, 2004). In this paper, we apply the HITS algorithm to document graphs and evaluate its performance on automatic unsupervised text unit extraction in the context of the text summarization task. The HITS algorithm distinguishes between "authorities" (pages with a large number of incoming links) and "hubs" (pages with a large number of outgoing links). For each node, HITS produces two sets of scores - an "authority" score, and a "hub" score:

$$HITS_A\left(V_i\right) = \sum_{V_j \in In(V_i)} HITS_H\left(V_j\right) \quad (1)$$

$$HITS_H\left(V_i\right) = \sum_{V_j \in Out(V_i)} HITS_A\left(V_j\right) \quad (2)$$

For the total rank (H) calculation we used the following four functions:

1. rank equals to the authority score

$$H\left(V_i\right) = HITS_A\left(V_i\right)$$

2. rank equals to the hub score

$$H\left(V_i\right) = HITS_H\left(V_i\right)$$

3. rank equals to the average between two scores

$$H\left(V_i\right) = avg\left\{HITS_A\left(V_i\right), HITS_H\left(V_i\right)\right\}$$

4. rank equals to the maximum between two scores

$$H\left(V_i\right) = \max\left\{HITS_A\left(V_i\right), HITS_H\left(V_i\right)\right\}$$

[1] The term frequency (TF) is the number of times the word appears in a document divided by the number of total words in the document.

[2] In our experiment the threshold is set to 0.05

[3] There are many variants for calculating sentence location score (Nobata et al., 2001). In this paper, we calculate it as an reciprocal of the sentence location in text: $Score\left(S_i\right) = \frac{1}{i}$

[4] There are many different formulas used to calculate *tfidf*. We use the next formula: $\frac{tf}{tf+1}\log_2\frac{|D|}{df}$, where tf - term frequency (as defined above), $|D|$ - total number of documents in the corpus, *df* - number of documents where the term appears.

average merit	rank	feature
0.192 +- 0.005	1	Frequent words distribution
0.029 +- 0	2	In Degree
0.029 +- 0	3	Out Degree
0.025 +- 0	4	Frequency
0.025 +- 0	5	Degree
0.017 +- 0	6	Headline Score
0.015 +- 0	7	Location Score
0.015 +- 0.001	8	Tfidf Score

Table 1: Feature selection results according to GainRatio value

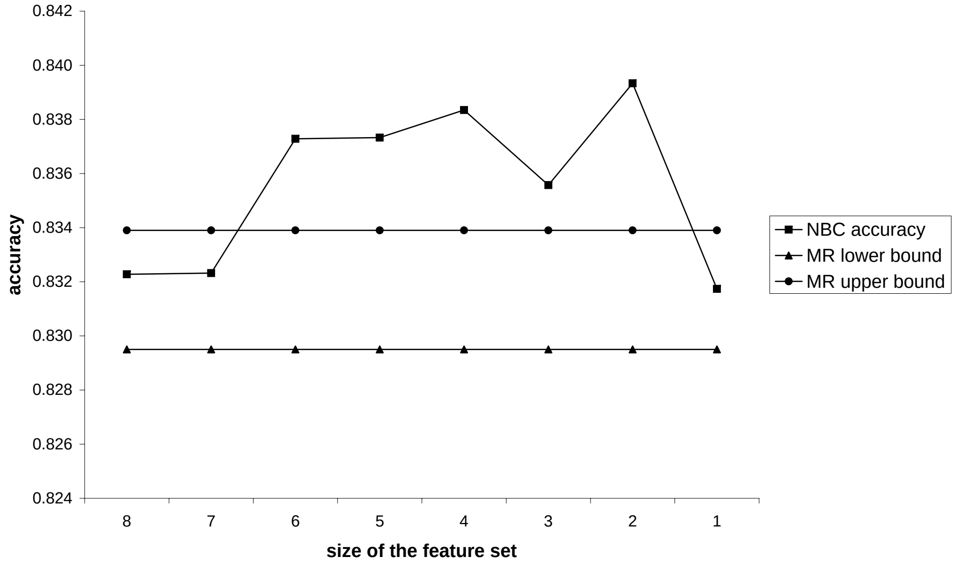


Figure 1: Accuracy for NaïveBayes classifier (NBC) and Majority Rule (MR)

4 Experimental results

All experiments have been performed on the collection of summarized news articles provided by the Document Understanding Conference 2002 (DUC, 2002). This collection contains 566 English texts along with 2-3 summaries per document on average. The size[5] of syntactic graphs extracted from these texts is 196 on average, varying from 62 to 876.

4.1 Supervised approach

We utilized several classification algorithms implemented in Weka's software (Witten and Frank, 2005) : J48 (known as C4.5), SMO (Support Vector Machine) and NaïveBayes for building binary classification models (a word belongs to summary / does not belong to the summary). For the training we built dataset with two classes: YES for nodes belonging to at least one summary of the document, and NO for those that do not belong to any summary. The accuracy of the default (majority) rule over all nodes is equal to the percentage of non-salient nodes (83.17%). For better classification results we examined the importance of each one of the features, described in Section 3.1 using automated feature selection. Table 1 presents the average GainRatio[6] values (”merits”) and the average rank of the features calculated from the DUC 2002 document collection, based on 10-fold cross validation.

As expected, the results of J48 and SMO (these algorithms perform feature selection while building the model) did not vary on different feature sets, while NaïveBayes gave the best accuracy on the reduced set. Figure 1 demonstrates the accuracy variations of NaïveBayes classifier on the different feature sets relative to the confidence inter-

[5] We define the size of a graph as the number of its vertices.

[6] $Gain\_Ratio(A) = \frac{Information\_Gain(A)}{Intrinsic\_Info(A)}$, where $Intrinsic\_Info(A) = -\sum_x \frac{N_x}{N} \log\left[\frac{N_x}{N}\right]$

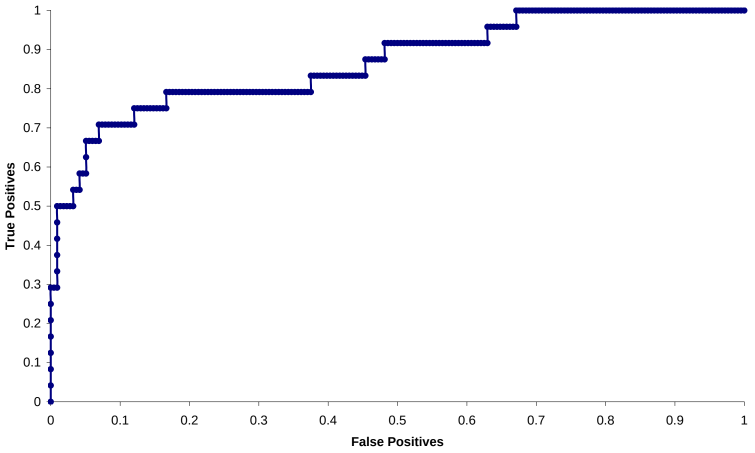


Figure 2: Sample ROC curve for one of the DUC'02 documents

Ranking function	Degree vectors	Converged vectors
Authority	0.625	0.600
Hub	0.620	0.601
Avg(Authority, Hub)	0.651	0.622
Max(Authority, Hub)	0.651	0.624

Table 2: Average AUC for each rank calculating function

val for the majority rule accuracy according to the normal approximation of the binomial distribution with $\alpha = 0.05$. Table 3 presents classification results for supervised algorithms (for NaïveBayes the results shown on the top 2 features) based on 10-fold cross validation as well as results of unsupervised learning.

4.2 Unsupervised approach

We have studied the following research questions:

1. Is it possible to induce some classification model based on HITS scores?

2. Is it necessary to run HITS until convergence?

In order to answer these questions we performed the following two experiments:

1. In the first one, we run HITS only one iteration. Note, that the ranks resulted from the first iteration are just in-degree and out-degree scores for each node in graph, and may be easily computed without even starting HITS[7].

2. In the second experiment we run HITS until convergence[8] (different number of steps for different graphs) and compare the results with the results of the first experiment.

After each experiment we sorted the nodes of each graph by rank for each function (see the rank calculating functions described in Section 3.2). After the sorting we built an ROC (Receiver Operating Characteristic) curve for each one of the graphs. Figure 2 demonstrates a sample ROC curve for one of the documents from DUC 2002 collection.

In order to compare between ranking functions (see Section 3.2) we calculated the average of AUC (Area Under Curve) for the 566 ROC curves for each function. Table 2 presents the average AUC results for the four functions. According to these results, functions that take into account both scores (average and maximum between two scores) are optimal. We use the *average* function for comparing and reporting the following results. Also, we can see that degree vectors give better AUC results

[7] Initially, both authority and hub vectors (a and h respectively) are set to $u = (1, 1, \ldots, 1)$. At each iteration HITS sets an authority vector to $a = A^T h$, and the hub vector to $h = Aa$, where A is an adjacency matrix of a graph. So, after the first iteration, $a = A^T u$ and $h = Au$, that are the vectors containing in-degree and out-degree scores for nodes in a graph respectively.

[8] There are many techniques to evaluate the convergence achievement. We say that convergence is achieved when for any vertex i in the graph the difference between the scores computed at two successive iterations falls below a given threshold: $\frac{|x_i^{k+1} - x_i^k|}{x_i^k} < 10^{-3}$ (Kamvar, 2003; Mihalcea and Tarau, 2004)

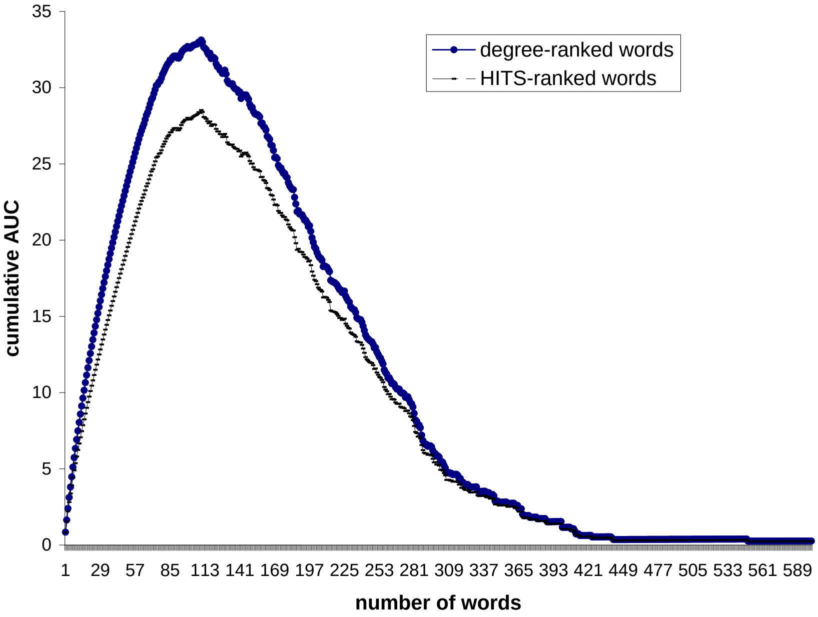


Figure 3: Cumulative AUC curves for degree and converged vectors

Method		Accuracy	TP	FP	Precision	Recall	F-Measure
Classification	J48	**0.847**	0.203	0.022	0.648	0.203	0.309
	NaïveBayes	0.839	0.099	0.011	0.648	0.099	0.172
	SMO	0.839	0.053	**0.002**	**0.867**	0.053	0.100
Degree-based Ranking	$N = 10$	0.813	0.186	0.031	0.602	0.186	0.282
	$N = 20$	0.799	0.296	0.080	0.480	0.296	0.362
	$N = 30$	0.772	0.377	0.138	0.409	0.377	0.388
	$N = 40$	0.739	**0.440**	0.200	0.360	**0.440**	**0.392**

Table 3: Results for each supervised and unsupervised method

than converged ones.

In order to compare between the degree-based vectors and the converged ones we calculated the precision curves[9] for each graph in both experiments. Then for each ranking method the curve representing an average cumulative AUC over the 566 precision curves was calculated. Figure 3 demonstrates the difference between resulting curves. As we can conclude from this chart, the degree-based vectors have a slight advantage over the converged ones. The "optimum" point where the average AUC is maximum for both methods is 111 words with the average AUC of 28.4 for degree-based words and 33 for HITS-ranked words. That does not have much significance because each document has a different "optimum" point.

Finally, we compared the results of unsupervised method against the supervised one. For this purpose, we consider unsupervised model based on extracting top N ranked words for four different values of N: 10, 20, 30 and 40. Table 3 represents the values for such commonly used metrics as: Accuracy, True Positive Rate, False Positive Rate, Precision, Recall and F-Measure respectively for each one of the tested methods. The optimal values are signed in bold.

Despite the relatively poor accuracy performance of both approaches, the precision and recall results for the unsupervised methods show that the classification model, where we choose the top most ranked words, definitely succeeds compared to the similar keyword extraction methods. (Leskovec et al., 2004) that is about "logical triples" extraction rather than single keyword extraction, presents results on DUC 2002 data, which are similar to ours in terms of the F-measure (40%

[9] For each number of top ranked words the percentage of positive words (belonging to summary) is shown.

against 39%) though our method requires much less linguistic pre-processing and uses a much smaller feature set (466 features against 8). (Mihalcea and Tarau, 2004) includes a more similar task to ours (single keyword extraction) though the definition of a keyword is different (”keywords manually assigned by the indexers” against the ”summary keywords”) and a different dataset (Inspec) was used for results presentation.

5 Conclusions

In this paper we have proposed and evaluated two graph-based approaches: supervised and unsupervised, for the cross-lingual keyword extraction to be used in extractive summarization of text documents. The empirical results suggest the following. When a large labeled training set of summarized documents is available, the supervised classification is the most accurate option for identifying the salient keywords in a document graph. When there is no high-quality training set of significant size, it is recommended to use the unsupervised method based on the node degree ranking, which also provides a higher F-measure than the supervised approach. The intuition behind this conclusion is very simple: most words that are highly ”interconnected” with other words in text (except stop-words) should contribute to the summary. According to our experimental results, we can extract up to 15 words with an average precision above 50%. Running HITS to its convergence is redundant, since it does not improve the initial results of the degree ranking.

6 Future work

The next stage of our extractive summarization methodology is generation of larger units from the selected keywords. At each step, we are going to reduce document graphs to contain larger units (subgraphs) as nodes and apply some ranking algorithms to the reduced graphs. This algorithm is iterative, where graph reduction steps are repeated until maximal subgraph size is exceeded or another constraint is met. Also, we plan to work on the supervised classification of sub-graphs, where many graph-based features will be extracted and evaluated.

In the future, we also intend to evaluate our method on additional graph representations of documents, especially on the concept-based representation where the graphs are built from the concepts fused from the texts. Once completed, the graph-based summarization methodology will be compared to previously developed state-of-the-art summarization methods and tools. All experiments will include collections of English and non-English documents to demonstrate the cross-linguality of our approach.

References

S. Brin and L. Page. 1998. *The anatomy of a large-scale hypertextual Web search engine*. *Computer Networks and ISDN Systems*, 30:1–7.

Document Understanding Documents 2002 [http://www-nlpir.nist.gov/projects/duc/index.html]

Sepandar D. Kamvar, Taher H. Haveliwala, and Gene H. Golub. *Adaptive methods for the computation of pagerank*. Technical report, Stanford University.

Kleinberg, J.M. 1999. *Authoritative sources in a hyperlinked environment*. *Journal of the ACM*, 46(5):604-632.

Last, M. and Markov A. 2005. *Identification of terrorist web sites with cross-lingual classiffication tools*. In Last, M. and Kandel, A. (Editors), Fighting Terror in Cyberspace. *World Scientific, Series in Machine Perception and Artificial Intelligence*, 65:117–143.

Leskovec, J., Grobelnik, M. and Milic-Frayling, N. 2004. *Learning Semantic Graph Mapping for Document Summarization*. In Proceedings of ECML/PKDD-2004 Workshop on Knowledge Discovery and Ontologies.

Mani, I. and Maybury, M.T. 1999. *Advances in Automatic Text Summarization*. MIT Press, Cambridge, MA.

Markov A., Last, M. and Kandel, A. 2007. *Fast Categorization of Web Documents Represented by Graphs*. Advances in Web Mining and Web Usage Analysis - 8th International Workshop on Knowledge Discovery on the Web, WEBKDD 2006, Revised Papers, O. Nasraoui, et al. (Eds). *Springer Lecture Notes in Computer Science* 4811:56–71.

Mihalcea R. 2004. *Graph-based ranking algorithms for sentence extraction, applied to text summarization*. In Proceedings of the 42nd Annual Meeting of the Association for Computational Lingusitics, Barcelona, Spain.

Mihalcea and P. Tarau. 2004. *TextRank - bringing order into texts*. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain.

Martin F. Porter. 1980. *An algorithm for suffix stripping*. *Program*, 14(3):130137, July.

Nobata, C., Sekine, S., Murata, M., Uchimoto, K., Utiyama, M. and Isahara, H. 2001. *Sentence extraction system assembling multiple evidence*. In Proceedings of the Second NTCIR Workshop Meeting, 5–213–218.

Salton, G., Wong, A. and Yang, C. S. 1975. *A Vector Space Model for Automatic Indexing Communications of the ACM*, 18(11):613-620.

Schenker, A., Bunke, H., Last, M., Kandel, A. 2005. *Graph-Theoretic Techniques for Web Content Mining*, volume 62. World Scientific, Series in Machine Perception and Artificial Intelligence.

Peter D. Turney. 2000. *Learning Algorithms for Keyphrase Extraction*. *Information Retrieval*, 2(4):303–336.

Ian H. Witten and Eibe Frank 2005. *Data Mining: Practical machine learning tools and techniques*, 2nd Edition, Morgan Kaufmann, San Francisco.

MultiSum
Query-Based Multi-Document Summarisation

Michael Rosner
Dept. Artificial Intelligence
University of Malta
mike.rosner@um.edu.mt

Carl Camilleri
Dept. Artificial Intelligence
University of Malta
ccam0002@um.edu.mt

Abstract

This paper describes a generic, open-domain multi-document summarisation system which combines new and existing techniques in a novel way. The system is capable of automatically identifying query-related online documents and compiling a report from the most useful sources, whilst presenting the result in such a way as to make it easy for the researcher to look up the information in its original context.

1 Introduction

Although electronic resources have several inherent advantages over traditional research media, they also introduce several drawbacks, such as *Information Overload* (Edmunds and Morris, 2000),which has become synonymous with the information retrieval phase of any research-related task. Another problem which is directly related to the one just described is that of *Source Identification* (Eppler and Mengis, 2004). This refers to the problem of having relevant results intermingled with results that are less relevant, or actually irrelevant.

Lastly, the researcher usually has to also manually traverse the relevant sources of information in order to form an answer to the research query.

These problems have led to the study of various areas in computing, all of which aim to try and minimise the manual effort of information retrieval and extraction, one of which is Multi-Document Summarisation (MDS).

The core aim of any MDS system is that of processing multiple sources of information and outputting a relatively brief but broad report or summary. Uses of MDS systems vary widely, from summarisation of closed-domain documents, such as news documents (Evans et al., 2004), to aggregation of information from several sources in an open domain.

2 Aims and Objectives

MDS techniques can be used in various tools that may help addressing the problems described in Section 1. On the other hand, a brief study of the relevant literature indicates that the majority of the work done in this area concerns closed-domains such as news summarisation, which is perhaps the reason why such tools have not yet become more popular. The objectives of this study are thus twofold.

- The primary objective is that of designing, implementing and evaluating an open-domain, query-based MDS system which is capable of compiling an acceptably-coherent report from the most relevant online sources of information, whilst making it easy for the reader to access the full source of information in its original context.
- A secondary objective of this study is Search Engine Optimisation (SEO): We require the system to produce summaries which, if published on the Internet, would be deemed relevant to the original query by search engine ranking algorithms. This is measured by keyoword density in the summary. Success on this objective addresses the problem of Source Identification since the summary would at the very least serve as a gateway to the other relevant sources from which it was formed.

Unsurprisingly, one of the problems that has to be overcome in the field of summarisation and particularly in an open-domain system such as ours is

Coling 2008: Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization, pages 25–32
Manchester, August 2008

the quality of output, as measured by a number of different linguistic and non-linguistic criteria (see Section 5). We have adopted a number of novel techniques to address this such as

- Multi-Layered Architecture
- Sentence Ordering Model
- Heuristic Sentence Filtering
- Paragraph Clustering

3 Background

3.1 Search Engine Ranking Criteria

Search engine ranking algorithms vary, and are continuously being optimised in order to provide better and more accurate results. However, some guidelines that outline factors which web masters need to take into account have been established (cf. Google (2007), Vaughn (2007)).

When ranking documents for a particular search query, ranking algorithms take into account both *on-page* and *off-page* factors. Off-page factors comprise mainly the number and quality of inbound links to a particular page, whilst on-page factors comprise various criteria, most important of which is the relevance of the content to the search query.

3.2 Multi-Document Summarisation

Several different approaches and processes have been developed in automatic MDS systems. These vary according to the problem domain, which usually defines particular formats for both input and output. However, five basic sub-systems of any MDS system can be identified (Mani, 2001).

1. **Unit Identification** During this first phase, input documents are parsed and tokenised into “units”, which can vary from single words to whole documents, according to the application problem.

2. **Unit Matching (Clustering)** The second stage involves grouping similar units together. In the context of MDS, similar units usually mean either identical or informationally-equivalent strings (Clarke, 2004), with the purpose of discovering the main themes in the different units and identify the most salient ones.

3. **Unit Filtering** The filtering stage eliminates units residing in clusters which are deemed to be non-salient.

4. **Compacting** During this phase, it is often assumed that different clusters contain similar units. Thus, a sample of units from different clusters is chosen.

5. **Presentation/Summary Generation** The last phase of the MDS process involves using the output from the Compacting stage, and generating a summary. Usually, naïve string concatenation does not produce coherent summaries and thus, techniques such as named entity normalisation and sentence ordering criteria are used at this stage.

3.3 Clustering Techniques

As outlined in Section 3.2, MDS often makes use of clustering techniques in order to group together similar units. Clustering can be defined as a process which performs “unsupervised classification of patterns into groups based on their similarity” (Clarke, 2004).

A particular clustering technique typically consists of three main components:

1. Pattern Representation

2. Similarity Measure

3. Clustering Algorithm

The very generic nature of our problem domain requires a clustering technique which is both suitable and without scenario-dependant parameters. Fung’s algorithm (Fung et al., 2003), comprising a pre-processing stage and a further three-phase core process, uses the following concepts, and is briefly described in Figure 1.

ItemSet A set of words occurring together within a document. An ItemSet composed of k words is called a *k-ItemSet*.

Global Support The Global Support of a word item is the number of documents from the document collection it appears in (cf. document frequency).

Cluster Support The Cluster Support of a word item is the number of documents within a cluster it appears in.

1. Pre-Processing - stem, remove stop words and convert to TFxIDF representation
2. Discover Global Frequent ItemSets
3. For each Global Frequent ItemSet (GFI) create a corresponding cluster, containing all documents that contain all items found within the GFI associated with each cluster. This GFI will act as a "label" to the cluster.
4. Make Clusters Disjoint

Figure 1: Hierarchical Document Clustering Using Frequent Itemsets

Frequent ItemSet An ItemSet occurring in a pre-determined minimum portion of a document collection. The pre-defined minimum is referred to as the *Minimum Support*, and is usually determined empirically according to the application.

Global Frequent ItemSet An ItemSet which is frequent within the whole document collection. The words within a Global Frequent ItemSet are referred to as *Global Frequent Items*, whilst the minimum support is referred to as the *Minimum Global Support*.

Cluster Frequent ItemSet An ItemSet which is frequent within a cluster. In this context, the minimum support is referred to as the *Minimum Cluster Support*.

With these definitions, it is now possible to describe into more detail the core non-trivial phases of the algorithm.

3.3.1 Discovering Global Frequent ItemSets

From the definition of an ItemSet, it can be concluded that the set of ItemSets is the power set of all features[1] within the document collection. Given even a small document collection, enumerating all the possible ItemSets and checking which of them are Global Frequent would be intractable. In order to discover Global Frequent ItemSets, the authors recommend the use of the Apriori Candidate Generation algorithm, a data mining algorithm proposed by Agrawal and Srikant (1994). This algorithm defines a way to reduce the number of candidate frequent ItemSets generated. The generation algorithm basically operates on the principle that, given a set of frequent *k-1*-ItemSets, a set of candidate frequent *k*-ItemSets can be generated such that each candidate is composed of frequent *k-1*-ItemSets.

Agrawal and Srikant (1994) also mention a similar algorithm proposed by Mannila et al. (1994). As illustrated in Figure 2, this algorithm consists of first generating candidates, and then pruning the result based on a principle similar to that mentioned.

3.3.2 Making Clusters Disjoint

The purpose of the last phase of the algorithm is converting a fuzzy cluster result to its crisp equivalent. In order to identify the best cluster for a document contained in multiple clusters, the authors define the scoring function illustrated in the equation of Figure 3, where x is a global and cluster-frequent item in doc_j, x' a global frequent but **not** cluster frequent item in doc_j, and $n(x)$ a weighted frequency (TF.IDF) of feature x in doc_j.

Using this function, the best cluster for a particular document is that which maximises the score. In case of a tie, the most specific cluster (having the largest number of labels) is chosen.

4 Procedure

The system was designed in two parts, namely a simple web-based user interface and a server process responsible for iterating sequentially over user queries and performing the content retrieval and summarisation tasks. The following sections describe the various sub-systems that compose the server process.

4.1 Content Retrieval

The Content Retrieval sub-system is responsible for retrieving web documents related to a user's query. This is done simply by querying a search engine and retrieving the top ranked documents[2]. Although throughout the course of this study the system was configured to use only Google as its document source, the number of search engines that can be queried is arbitrary, and the system can

[1] Features here constitute distinct, single words found in the whole document collection. In practice, stemming is applied before feature extraction.

[2] It was empirically determined that retrieving the top 30 ranked documents achieved the best results. Considering less documents meant that, in most scenarios, main relevant sources were missed, whilst considering more documents caused the infiltration of irrelevant information

1. **Join**
   $C_k = \{X \cup X' \mid X, X' \in L_{k-1},\ |X \cap X'| = k-2\}$
2. **Prune**
   $C_k = \{X \in C'_k \mid X \text{ contains } k \text{ members of } L_{k-1}\}$

Figure 2: Candidate Generation Algorithm by Mannila et al. (1994)

$$Score(C_i \leftarrow doc_j) = \sum_{x} n(x) \times cluster\_support(x) - \sum_{x'} n(x') \times global\_support(x')$$

Figure 3: Definition of Scoring Function

be given a set of parameters to query a particular search engine.

4.2 Content Extraction

The Content Extraction module is responsible for transforming the retrieved HTML documents into raw text. However, a simple de-tagging process is not sufficient. This module was designed so as to be able to identify the main content of a web document, and leave out other clutter such as navigation menus and headings. Finn et al. (2001) introduce a generic method to achieve this, by translating the content extraction problem to an optimisation problem. The authors observe that, essentially, an HTML document consists of two types of elements, that is, actual text and HTML tags. Thus, such a document can easily be encoded as a binary string B, where 0 represents a natural word, whilst 1 represents and HTML tag. Figure 4 shows a typical graphical representation obtained when cumulative HTML tag tokens are graphed against the cumulative number of tokens in a typical HTML document.

Finn et al. (2001) suggest that, typically, the plateau that can be discerned in such a graph contains the actual document content. Therefore, in order to extract the content, the start and end point of the plateau (marked with black boxes in Figure , and referred to hereafter as `i` and `j` respectively) must be identified.

The optimisation problem now becomes maximisation of the number of HTML tags below `i` and above `j`, in parallel with maximisation of the number of natural language words between `i` and `j`. The maximisation formula proposed by the authors is given by Equation 1.

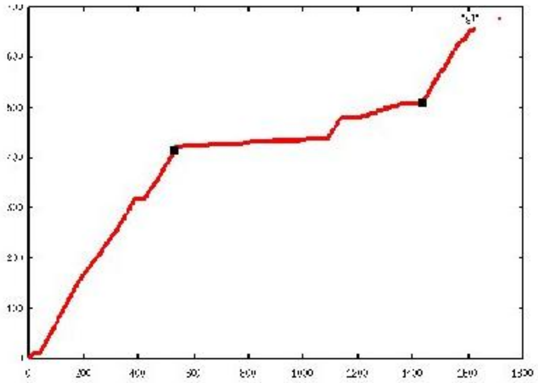


Figure 4: Total HTML Tokens VS Total Tokens (Finn et al., 2001)

$$T_{i,j} = \sum_{n=0}^{i-1} B_n + \sum_{n=i}^{j} (1-B_n) + \sum_{n=j+1}^{N-1} (1-B_n) \quad (1)$$

Our Content Extraction module is further decomposed into three sub-modules. The first is a pre-processing module, which parses out the body of the HTML document, and removes superfluous content such as scripts and styling sections. The second and core sub-module consists namely of an implementation of the content extraction method introduced by Finn et al. (2001), which is primarily responsible for identifying the main content section of the input document. The last post-processing module then ensures that the output from the previous sub-modules is converted to raw text, by performing an HTML detagging processing and also inserting paragraph marks in places where they are explicit cf. HTML `<p>` tag) or where an element from a predefined set of HTML text break delimiters occurs.

4.3 Summarisation

The overall design of the core summarisation module is loosely based upon the two-tiered MDS architecture introduced by Torralbo et al. (2005) The following sections map our system to a similar two-tiered architecture, and explain how each module operates.

Document Identification

Document Identification is trivial, since documents are explicitly defined by the content retrieval module, the output of which is basically a set of query-related text documents.

Document Filtering

The job of Document Filtering is partially done at the very beginning by the search engine. However, our system further refines the document collection by pre-processing each document, applying a noise[3] removal procedure, stemming and stop word and rare word removal. Each document is then converted to a bag of words, or the Vector Space Model, where each word is associated with its corresponding TF•IDF measure. Any document which, after pre-processing, ends up with an empty bag of words, is filtered out from the document collection. Furthermore, in order to ensure the robustness of the system especially in subsequent intensive processing, documents which are longer than 5 times the average document length are truncated.

Paragraph Identification

As outlined in Section 4.2, the Content Extraction sub-system inserts paragraph indicators in the text wherever appropriate. Thus, the paragraph identification phase is trivial, and entails only splitting the content of a document at the indicated positions.

Paragraph Clustering and Filtering

In contrast to the technique of Torralbo et al. (2005), a paragraph filtering module was introduced in order to select only the most informative, query-related paragraphs. To achieve this, we implemented the clustering technique outlined in Section 3.3 in order to obtain clusters of thematically-similar paragraphs, using the Global Frequent ItemSet generation technique from Mannila et al. (1994) and setting the Minimum Global Support and Minimum Cluster Support parameters to 35 and 50 respectively.

```
1. For each paragraph $p_k$
    (a) Initialise the target summary $Sum_k$ as an empty text
    (b) Let $p = p_k$
    (c) Remove the first sentence $s$ from $p$, and add it at the end of $Sum_k$.
    (d) Calculate the similarity between $s$ and the first sentence of all the paragraphs, using the size of the intersection of the two vectors of words as a similarity metric.
    (e) Let $p$ be the paragraph whose first sentence maximises the similarity, and go back to step (c) with that paragraph. If the best similarity is 0, stop.
2. Choose the longest one of the $k$ different summaries.
```

Figure 5: Summary Generation Algorithm (Torralbo et al., 2005)

The filtering technique then consists of simply choosing the largest cluster. This is based on the intuition that most of the paragraphs having the central theme as their main theme will get clustered together. Therefore, choosing the largest cluster of paragraphs would filter out irrelevant paragraphs. This paragraph filtering method may filter out paragraphs which are actually relevant, however, we rely on the redundancy of information usually found in information obtained from the web. Thus, the paragraph filtering gives more importance to filtering out all the irrelevant paragraphs.

Summary Generation

The role of the summary generation module is to generate a report from a cluster of paragraphs. We based our summary generation method on that used by Torralbo et al. (2005), which is illustrated in Figure 5. However, in order to make it more applicable to our problem domain and increase the output quality, we introduced some improvements.

Sentence Ordering Model We introduced a probabilistic sentence ordering model which enables the algorithm to choose the sentence that

[3] "Noise" refers to any character which is not in the English alphabet.

maximises the probability given the previous sentence. The sentence ordering model, based on a method of probabilistic text structuring introduced by Lapata (2003), is trained upon the whole document collection. We used Minipar (Lin, 1993), a dependency-based parser, in order to identify verbs, nouns, verb dependencies and noun dependencies. Using counts of these features and Simple Good-Turing smoothing (Gale and Sampson, 1995), we were able to construct a probabilistic sentence ordering model such that, during summary generation, given the previous sentence, we are able to identify the sentence which is the most likely to occur from the pool of sentences appearing at the beginning of the remaining paragraphs.

Sentence Filtering We also introduced at this stage a method to filter out sentences that decrease the coherency and fluency of the resultant summary. This is based on two criteria:

1. **Very low probability of occurrence**
   If the most likely next-occurring sentence that is chosen and removed from a paragraph still has a very low probabilistic score, it is not added to the output summary.

2. **Heuristics**
   We also introduce a set of heuristics to filter out sentences having a wrong construction or sentences which would not make sense in a given context. These heuristics include:

   (a) Sentences with no verbs

   (b) Sentences starting with an adverb and occurring at a paragraph transition within the summary

   (c) Sentences occurring at a context switch[4] within the summary and starting with a word matched with a select list of words that usually occur as anaphora

   (d) Malformed sentences (including sentences not starting with a capital letter and very short sentences)

5 Evaluation

5.1 Automatic Evaluation

5.1.1 Coherence Evaluation

In order to evaluate the local coherence of the reports generated by the system, we employed an automatic coherence evaluation method introduced by Barzilay and Lapata (2005)[5]. The main objective of this part of the evaluation phase was to determine the effect on output quality when parameters are varied, namely the minimum cluster support parameter for the clustering algorithm, and the key phrase popularity.

From this evaluation, we empirically determined that the optimum minimum cluster support threshold for this application is 50, whilst the quality of the output is directly proportional to the keyword popularity.

5.1.2 Keyword Density Evaluation

Here we focused on determining whether the secondary objective was achieved (cf. section 2).

We measured the frequency of occurrence of the keyword phrase within the output, or more specifically, the keyword density. The average key phrase density achieved by the system was 1.32%, when taking into account (i) the original keyword phrase and its constituent keywords, and (ii) secondary keyword phrases and their constituents.

5.2 Manual Quality Evaluation

In order to measure the quality of the output and determine whether the objectives of the study was achieved, three users were introduced to the system and asked to grade the system, on a scale of 1-5, on several criteria. Table 1 illustrates the results obtained from this evaluation.

6 Conclusions

6.1 Interpretation of Results

In this section we will identify some conclusions elicited from the results obtained from the evaluation phase and illustrated in Section 5.

Automatic Coherence Evaluation The automatic coherence evaluation tests, although, in this application, the level of "coherence" indicated did not match that of manual evaluation, provided nonetheless a standard by which different outputs from the system using different parameters and application scenarios could be compared. From the results, we could empirically determine that the optimal value for the cluster support parameter was around 50%. Furthermore, unsurprisingly, the system tends to produce output of a higher quality

[4] *Context Switch* refers to scenarios where a candidate sentence comes from a different document than that of the last sentence in the summary.

[5] Data required to set up the automatic coherence evaluation model was available from the author's website http://people.csail.mit.edu/regina/coherence/.

	Grammaticality	Non-Redundancy	Referential Clarity	Focus	Structure	Naturalness	Usefulness
Average	3.62	2.21	4.03	4.28	3.27	2.76	4.78

Table 1: Results of Manual Evaluation

in scenarios where the keyword phrase is popular, and thus more data is available.

SEO Evaluation From an SEO perspective, it was predictable that the system would produce query-related text, since its data source is obtained from query-related search engine results. However, the resulting average keyword density achieved is significant, and is at a level which is totally acceptable by most search engine ranking algorithms[6].

Manual Quality Evaluation Due to limited resources, the results of the manual evaluation procedure were not statistically significant since only three users were involved in evaluating six summaries. However, allowing for a factor of subjectivity, some conclusions could still be elicited, namely:

1. The system did not perform well enough to have its output rated as high as a manual summarisation procedure. This can be concluded from the low rating on the output *Naturalness* criterion, as well as from the presence of repeated and irrelevant content in some of the output summaries.

2. The system performed acceptably well in generating reports that were adequately coherent and high-level enough to give an overview of concepts represented by users' queries. This can be concluded from the average scores achieved in the *Focus* and *Referential Clarity* criteria.

3. The evaluators were also asked to give a grade indicating whether this system and similar tools would actually be useful. A positive grade was obtained on this criterion, indicating that the system achieved the MDS objective, enabling users to get a brief overview of the topic as well as facilitating document identification.

When comparing these results to those achieved by Torralbo et al. (2005), we can elicit two main conclusions:

1. Although our system achieved lower rankings on the *Non-Redundancy, Structure* and *Grammaticality* criteria, these rankings were not unacceptable. We could attributed this to the more generic domain in which our system operates, where it is not possible to introduce fixed heuristics such as those used by Torralbo et al. (2005) for avoiding repeated information by replacing a term definition by its corresponding acronym. Such heuristics tend to be relevant in the context of such a term definition system.

2. Our system achieved higher grades on the *Referential Clarity* and *Focus* criteria. Given the fact that the system of Torralbo et al. (2005) retrieves results from search engines in a similar way used by our system, the improvement Focus might be attributed to the fact that our paragraph filtering methodology tends to perform well in selecting only the most relevant parts of the document base. Furthermore, the improved grade achieved in the Referential Clarity criterion might arise from the more advanced sentence ordering methodology used, as well as to the different heuristic-based sentence filtering techniques employed by our summary generation module.

6.2 Limitations

The main limitation is that the quality of the output is very susceptible to the quality and amount of resources available. However, we also noticed a severe fall in quality where results were largely composed of business-oriented portals, which tend to lack textual information. Furthermore, the output summary is largely dictated by the results of search engines. Therefore, the queries submitted to the system must be formulated similarly to those submitted to search engines, since the system would fail to generate a focused summary for queries which, when submitted to traditional search engines, return irrelevant results.

The system performance is also limited by the quality and number of external components being referenced, which are not state of the art and which

[6] Very high keyword density (more than a threshold of 2% - 5%) is usually considered as a spammy technique known as *keyword stuffing*.

introduce performance bottlenecks by imposing a batch-processing regime.

6.3 Final Conclusions

Our system combines several existing techniques in a novel way. New techniques, such as our Heuristic-Based Sentence Filtering algorithm, are also introduced.

The primary objective of creating an MDS was achieved albeit with limited "coherency". However, our system was considered a useful research tool - supporting the hypothesis that a partially coherent but understandable report with minimum effort is arguably better than a perfectly coherent one, if the latter is unrealistically laborious to produce.

The secondary SEO objective was also achieved, to the extent that the system generated query-related content that has a natural level of key phrase density. Such content has the potential of being considered query-related also by search engine ranking algorithms, if published within the right context.

7 Future Work

There remains much is to be done. We propose:

- To increase the output quality and naturalness by focusing on an a sub-system for anaphora identification and resolution which would complement our probabilistic sentence ordering model.
- To widen the scope by applying the system to sources of information other than web documents.
- To convert our batch-processing system to an interactive one by incorporating all the required tools within the same environment.

References

Agrawal, R. and Ramakrishnan Srikant. 1994. Fast algorithms for mining association rules in large databases. In *VLDB'94, Proc. of 20th International Conference on Very Large Data Bases, Sept. 1994, Santiago de Chile, Chile*, pages 487–499. Morgan Kaufmann.

Barzilay, R. and M. Lapata. 2005. Modeling local coherence: an entity-based approach. In *ACL '05: Proc. 43rd Annual Meeting of the ACL*, pages 141–148, Morristown, NJ, USA. ACL.

Clarke, J. 2004. Clustering techniques for multi-document summarisation. Master's thesis, University of Sheffield.

Edmunds, A. and A. Morris. 2000. The problem of information overload in business organisations: a review of the literature. *Int. Journal of Information Management*, 20(1):17–28.

Eppler, M.J. and J. Mengis. 2004. The Concept of Information Overload: A Review of Literature from Organization Science, Accounting, Marketing, MIS, and Related Disciplines. *The Information Society*, 20(5):325–344.

Evans, D.K., J.L. Klavans, and K.R. McKeown. 2004. Columbia Newsblaster: Multilingual News Summarization on the Web. *Proc. HLT Conference and the NAACL Annual Meeting*.

Finn, A., N. Kushmerick, and B. Smyth. 2001. Fact or fiction: Content classification for digital libraries. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*.

Fung, B.C.M., K. Wang, and M. Ester. 2003. Hierarchical Document Clustering Using Frequent Itemsets. *Proc. of the SIAM International Conference on Data Mining*, 30.

Gale, W.A. and G. Sampson. 1995. Good-Turing Frequency Estimation Without Tears. *Journal of Quantitative Linguistics*, 2(3):217–237.

Google. 2007. *Google Webmaster Guidelines*. http://www.google.com/support/webmasters /bin/answer.py?answer=35769.

Lapata, M. 2003. Probabilistic text structuring: Experiments with sentence ordering. *Proc. of the 41st Meeting of the ACL*, pages 545–552.

Lin, Dekang. 1993. Principle-based parsing without overgeneration. In *Meeting of the ACL*, pages 112–120.

Mani, I. 2001. Automatic Summarization. *Computational Linguistics*, 28(2).

Mannila, Heikki, Hannu Toivonen, and A. Inkeri Verkamo. 1994. Efficient algorithms for discovering association rules. In Fayyad, Usama M. and Ramasamy Uthurusamy, editors, *AAAI Workshop on Knowledge Discovery in Databases (KDD-94)*, pages 181–192, Seattle, Washington. AAAI Press.

Torralbo, R., E. Alfonseca, A. Moreno-Sandoval, and J.M. Guirao. 2005. Automatic generation of term definitions using multidocument summarisation from the Web. In *Proc. Workshop on Crossing Barriers in Text Summarisation Research, RANLP Borovets*.

Vaughn. 2007. *Google Ranking Factors - SEO Checklist*. http://www.vaughns-1-pagers.com/internet/google-ranking-factors.htm.

Mixed-Source Multi-Document Speech-to-Text Summarization

Ricardo Ribeiro
INESC ID Lisboa/ISCTE/IST
Spoken Language Systems Lab
Rua Alves Redol, 9
1000-029 Lisboa, Portugal
rdmr@l2f.inesc-id.pt

David Martins de Matos
INESC ID Lisboa/IST
Spoken Language Systems Lab
Rua Alves Redol, 9
1000-029 Lisboa, Portugal
david@l2f.inesc-id.pt

Abstract

Speech-to-text summarization systems usually take as input the output of an automatic speech recognition (ASR) system that is affected by issues like speech recognition errors, disfluencies, or difficulties in the accurate identification of sentence boundaries. We propose the inclusion of related, solid background information to cope with the difficulties of summarizing spoken language and the use of multi-document summarization techniques in single document speech-to-text summarization. In this work, we explore the possibilities offered by phonetic information to select the background information and conduct a perceptual evaluation to better assess the relevance of the inclusion of that information. Results show that summaries generated using this approach are considerably better than those produced by an up-to-date latent semantic analysis (LSA) summarization method and suggest that humans prefer summaries restricted to the information conveyed in the input source.

1 Introduction

News have been the subject of summarization for a long time, demonstrating the importance of both the subject and the process. Systems like NewsInEssence (Radev et al., 2005), Newsblaster (McKeown et al., 2002), or even Google News substantiate this relevance that is also supported by the spoken language scenario, where most speech summarization systems concentrate on broadcast news (McKeown et al., 2005). Nevertheless, although the pioneering efforts on summarization go back to the work of Luhn (1958) and Edmundson (1969), it is only after the renaissance of summarization as a research area of great activity—following up on the Dagstuhl Seminar (Endres-Niggemeyer et al., 1995)—that the first multi-document news summarization system, SUMMONS (McKeown and Radev, 1995), makes its breakthrough (Radev et al., 2005; Spärck Jones, 2007). In what concerns speech summarization, the state of affairs is more problematic: news summarization systems appeared later and still focus only on single document summarization (McKeown et al., 2005). In fact, while text summarization has attained some degree of success (Hovy, 2003; McKeown et al., 2005; Spärck Jones, 2007) due to the considerable body of work, speech summarization still requires further research, both in speech and text analysis, in order to overcome the specific challenges of the task (McKeown et al., 2005; Furui, 2007). Issues like speech recognition errors, disfluencies, and difficulties in accurately identifying sentence boundaries must be taken into account when summarizing spoken language. However, if on the one hand, recognition errors seem not to have a considerable impact on the summarization task (Murray et al., 2006; Murray et al., 2005), on the other hand, spoken language summarization systems often explore ways of minimizing that impact (Zechner and Waibel, 2000; Hori et al., 2003; Kikuchi et al., 2003).

We argue that by including related solid background information from a different source less prone to this kind of errors (e.g., a textual source) in the summarization process, we are able to reduce the influence of recognition errors on the resulting summary. To support this argument, we developed a new approach to speech-to-text summarization that combines information from multiple information sources to produce a summary driven by the spoken language document to be summarized. The idea mimics the natural human behavior, in which information acquired from different sources is used to build a better understanding of a given topic (Wan et al., 2007). Furthermore, we build on the conjecture that this background information is often used by humans to overcome perception difficulties. In that sense, one of our goals is also to understand what is expected in a summary: a comprehensive, shorter, text that addresses the same subject of the input source to be summarized (possibly introducing new information); or a text restricted to the information conveyed in the input source.

This work explores the use of phonetic domain information to overcome speech recognition errors and disfluencies. Instead of using the traditional output of the ASR module, we use the phonetic transliteration of the output and compare it to the phonetic transliteration of solid background information. This enables the use of text, related to the input source, free from the common speech recognition issues, in further processing.

We use broadcast news as a case study and news stories from online newspapers provide the background information. Media monitoring systems, used to transcribe and disseminate news, provide an adequate framework to test the proposed method.

This document is organized as follows: section 2 briefly introduces the related work; section 3 presents a characterization of the speech-to-text summarization problem and how we propose to address it; section 4 explicits our use of phonetic domain information, given the previously defined context; the next section describes the case study, including the experimental set up and results; conclusions close the document.

2 Related Work

McKeown et al. (2005) depict spoken language summarization as a much harder task than text summarization. In fact, the previously enumerated problems that make speech summarization such a difficult task constrain the applicability of text summarization techniques to speech summarization (although in the presence of planned speech, as it partly happens in the broadcast news domain, that portability is more feasible (Christensen et al., 2003)). On the other hand, speech offers possibilities like the use of prosody and speaker identification to ascertain relevant content.

Furui (2007) identifies three main approaches to speech summarization: sentence extraction-based methods, sentence compaction-based methods, and combinations of both.

Sentence extractive methods comprehend, essentially, methods like LSA (Gong and Liu, 2001), Maximal Marginal Relevance (Carbonell and Goldstein, 1998), and feature-based methods (Edmundson, 1969). Feature-based methods combine several types of features: current work uses lexical, acoustic/prosodic, structural, and discourse features to summarize documents from domains like broadcast news or meetings (Maskey and Hirschberg, 2005; Murray et al., 2006; Ribeiro and de Matos, 2007). Even so, spoken language summarization is still quite distant from text summarization in what concerns the use of discourse features, and shallow approaches is what can be found in state-of-the-art work such as the one presented by Maskey and Hirschberg (2005) or Murray et al. (2006). Sentence compaction methods are based on word removal from the transcription, with recognition confidence scores playing a major role (Hori et al., 2003). A combination of these two types of methods was developed by Kikuchi et al. (2003), where summarization is performed in two steps: first, sentence extraction is done through feature combination; second, compaction is done by scoring the words in each sentence and then a dynamic programming technique is applied to select the words that will remain in the sentence to be included in the summary.

3 Problem Characterization

Summarization can be seen as a reductive transformation ϕ that, given an input source I, produces a summary S:

$$S = \phi(I),$$

where $len(S) < len(I)$ and $inf(S)$ is as close as possible of $inf(I)$; $len()$ is the length of the given input and $inf()$ is the information conveyed by its argument.

The problem is that in order to compute S, we are not using I, but $\tilde{I}$, a noisy representation of I.

Thus, we are computing $\tilde{S}$, which is a summary affected by the noise present in $\tilde{I}$:

$$\tilde{S} = \phi(\tilde{I}).$$

This means that

$$inf(\tilde{S}) \subset inf(S) \subset inf(I), \text{ whereas}$$
$$len(\tilde{S}) \approx len(S) < len(I).$$

Our argument is that using a similar reductive transformation ψ, where solid background information B is also given as input, it is possible to compute a summary $\hat{S}$:

$$\hat{S} = \psi(\tilde{I}, B), \text{ such that}$$
$$inf(\tilde{S}) \subset (inf(\hat{S}) \cap inf(S)) \subset inf(I), \text{ with}$$
$$len(\hat{S}) \approx len(\tilde{S}) \approx len(S) < len(I).$$

As seen in section 2, the most common method to perform these transformations is by selecting sentences (or extracts) from the corresponding input sources.

Thus, let the input source representation $\tilde{I}$ be composed by a sequence of extracts e_i,

$$\tilde{I} = e_1, e_2, \ldots, e_n$$

and the background information be defined as a sequence of sentences

$$B = s_1, s_2, \ldots, s_m.$$

The proposed method consists of selecting sentences s_i form the background information B such that

$$sim(s_i, e_j) < \varepsilon \wedge 0 \leq i \leq m \wedge 0 \leq j \leq n,$$

with $sim()$ being a similarity function and ε an adequate threshold. The difficulty lies in defining the function and the threshold.

4 Working in the phonetic domain

The approach we introduce minimizes the effects of recognition errors through the selection, from previously determined background knowledge, of sentence-like units close to the ones of the news story transcription. In order to select sentence-like units, while diminishing recognition problems, we compute the similarity between them at the phonetic level. The estimation of the threshold is based on the distance, measured in the phonetic domain, between the output of the ASR and its hand-corrected version.

Feature	Values
Type	vowel, consonant
Vowel length	short, long, diphthong, schwa
Vowel height	high, mid, low
Vowel frontness	front mid back
Lip rounding	yes, no
Consonant type	stop, fricative, affricative, nasal, liquid
Place of articulation	labial, alveolar, palatal, labio-dental, dental, velar
Consonant voicing	yes, no

Table 1: Phone features.

The selection of sentences from the background information is based on the alignment cost of the phonetic transcriptions of sentences from the input source and sentence from the background information. Sentences from the background information with alignment costs below the estimated threshold are selected to be used in summary generation.

4.1 Similarity Between Segments

There are several ways to compute phonetic similarity. Kessler (2005) states that phonetic distance can be seen as, among other things, differences between acoustic properties of the speech stream, differences in the articulatory positions during production, or as the perceptual distance between isolated sounds. Choosing a way to calculate phonetic distance is a complex process.

The phone similarity function used in this process is based on a model of phone production, where the phone features correspond to the articulatory positions during production: the greater the matching between phone features, the smaller the distance between phones. The phone features used are described in table 1.

The computation of the similarity between sentence-like units is based on the alignment of the phonetic transcriptions of the given segments. The generation of the possible alignments and the selection of the best alignment is done through the use of Weighted Finite-State Transducers (WFSTs) (Mohri, 1997; Paulo and Oliveira, 2002).

4.2 Threshold Estimation Process

To estimate the threshold to be used in the sentence selection process, we use the algorithm presented in figure 1. The procedure consists of comparing automatic transcriptions and their hand-corrected versions: the output is the average difference between the submitted inputs.

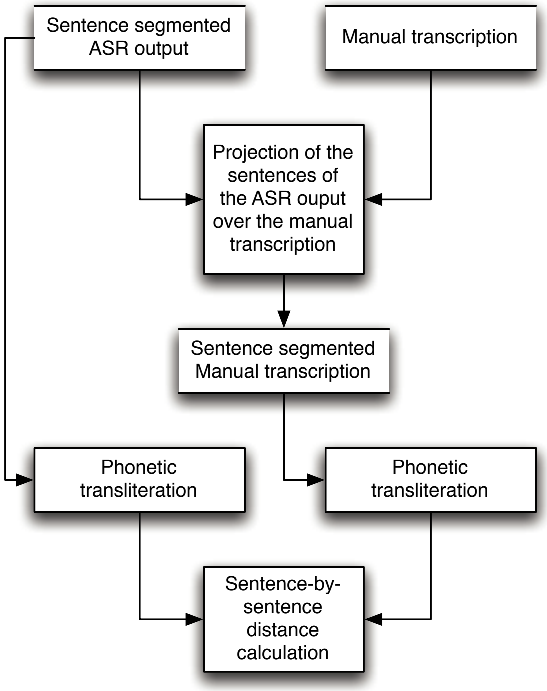

Sentence segmented ASR output
Manual transcription
Projection of the sentences of the ASR ouput over the manual transcription
Sentence segmented Manual transcription
Phonetic transliteration
Phonetic transliteration
Sentence-by-sentence distance calculation

Figure 1: Threshold estimation process.

The idea is that the phonetic distance between the automatic transcription and its hand-corrected version would be similar to the phonetic distance between the automatic transcription and the background information. Even though this heuristic may appear naif, we believe it is adequate as a rough approach, considering the target material (broadcast news).

5 A Case Study Using Broadcast News

5.1 Media Monitoring System

SSNT (Amaral et al., 2007) is a system for selective dissemination of multimedia contents, working primarily with Portuguese broadcast news services. The system is based on an ASR module, that generates the transcriptions used by the topic segmentation, topic indexing, and title&summarization modules. User profiles enable the system to deliver e-mails containing relevant news stories. These messages contain the name of the news service, a generated title, a summary, a link to the corresponding video segment, and a classification according to a thesaurus used by the broadcasting company.

Preceding the speech recognition module, an audio preprocessing module, based on Multi-layer Perceptrons, classifies the audio in accordance to several criteria: speech/non-speech, speaker segmentation and clustering, gender, and background conditions.

The ASR module, based on a hybrid speech recognition system that combines Hidden Markov Models with Multi-layer Perceptrons, with an average word error rate of 24% (Amaral et al., 2007), greatly influences the performance of the subsequent modules.

The topic segmentation and topic indexing modules were developed by Amaral and Trancoso (2004). Topic segmentation is based on clustering and groups transcribed segments into stories. The algorithm relies on a heuristic derived from the structure of the news services: each story starts with a segment spoken by the anchor. This module achieved an $F\text{-}measure$ of 68% (Amaral et al., 2007). The main problem identified by the authors was boundary deletion: a problem which impacts the summarization task. Topic indexing is based on a hierarchically organized thematic thesaurus provided by the broadcasting company. The hierarchy has 22 thematic areas on the first level, for which the module achieved a correctness of 91.4% (Amaral et al., 2006; Amaral et al., 2007).

Batista et al. (2007) inserted a module for recovering punctuation marks, based on maximum entropy models, after the ASR module. The punctuation marks addressed were the “full stop” and “comma”, which provide the sentence units necessary for use in the title&summarization module. This module achieved an $F\text{-}measure$ of 56% and SER (Slot Error Rate, the measure commonly used to evaluate this kind of task) of 0.74.

Currently, the title&summarization module produces a summary composed by the first n sentences, as detected by the previous module, of each news story and a title (the first sentence).

5.2 Corpora

Two corpora were used in this experiment: a broadcast news corpus, the subject of our summarization efforts; and a written newspaper corpus, used to select the background information.

Corpus	Stories	SUs	Tokens	Duration
train	184	2661	57063	5h
test	26	627	7360	1h

Table 2: Broadcast news corpus composition.

The broadcast news corpus is composed by 6 Portuguese news programs, and exists in two versions: an automatically processed one, and a hand-corrected one. Its composition (number of stories, number of sentence-like units (SUs), number of tokens, and duration) is detailed in table 2. To estimate the threshold used for the selection of the background information, 5 news programs were used. The last one was used for evaluation.

The written newspaper corpus consists of the online version a Portuguese newspaper, downloaded daily from the Internet. In this experiment, three editions of the newspaper were used, corresponding to the day and the two previous days of the news program to be summarized. The corpus is composed by 135 articles, 1418 sentence-like units, and 43102 tokens.

5.3 The Summarization Process

The summarization process we implemented is characterized by the use of LSA to compute the relevance of the extracts (sentence-like units) of the given input source.

LSA is based on the singular vector decomposition (SVD) of the term-sentence frequency $m \times n$ matrix, M. U is an $m \times n$ matrix of left singular vectors; Σ is the $n \times n$ diagonal matrix of singular values; and, V is the $n \times n$ matrix of right singular vectors (only possible if $m \geq n$):

$$M = U\Sigma V^T$$

The idea behind the method is that the decomposition captures the underlying topics of the document by means of co-occurrence of terms (the latent semantic analysis), and identifies the best representative sentence-like units of each topic. Summary creation can be done by picking the best representatives of the most relevant topics according to a defined strategy.

For this summarization process, we implemented a module following the original ideas of Gong and Liu (2001) and the ones of Murray, Renals, and Carletta (2005) for solving dimensionality problems, and using, for matrix operations, the GNU Scientific Library[1].

5.4 Experimental Results

Our main objective was to understand if it is possible to select relevant information from background information that could improve the quality of speech-to-text summaries. To assess the validity of this hypothesis, five different processes of generating a summary were considered. To better analyze the influence of the background information, all automatic summarization methods are based on the up-to-date LSA method previously described: one taking as input only the news story to be summarized (*Simple*) and used as baseline; other taking as input only the selected background information (*Background only*); and, the last one, using both the news story and the background information (*Background* + *News*). The other two processes were human: extractive (using only the news story) and abstractive (understanding the news story and condensing it by means of paraphrase). Since the abstractive summaries had already been created, summary size was determined by their size (which means creating summaries using a compression rate of around 10% of the original size).

As mentioned before, the whole summarization process begins with the selection of the background information. Using the threshold estimated as described in section 4.2 and the method described in section 4.1 to compute similarity between sentence-like units, no background information was selected for 11 of the 26 news stories of the test corpus. For the remaining 15 news stories, summaries were generated using the three automatic summarization strategies described before.

In what concerns the evaluation process, although ROUGE (Lin, 2004) is the most common evaluation metric for the automatic evaluation of summarization, since our approach might introduce in the summary information that it is not present in the original input source, we found that a human evaluation was more adequate to assess the relevance of that additional information. A perceptual evaluation is also adequate to assess the perceive quality of the summaries and a better indicator of the what is expected to be in a summary.

We asked an heterogeneous group of sixteen people to evaluate the summaries created for the 15 news stories for which background information

[1] `http://www.gnu.org/software/gsl/`

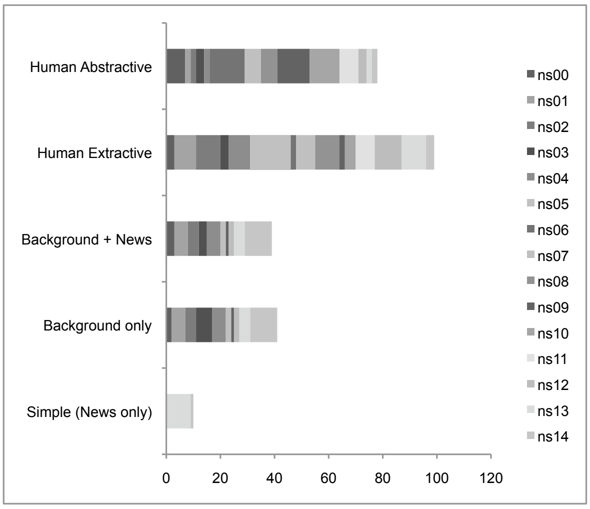


Figure 2: Overall results for each summary creation method (nsnn identifies a news story).

was selected. Each evaluator was given, for each story, the news story itself (without background information) and five summaries, corresponding to the five different methods presented before. The evaluation procedure consisted in identifying the best summary and in the classification of each summary (1–5, 5 is better) according to its content and readability (which covers issues like grammaticality, existence of redundant information, or entity references (Nenkova, 2006)).

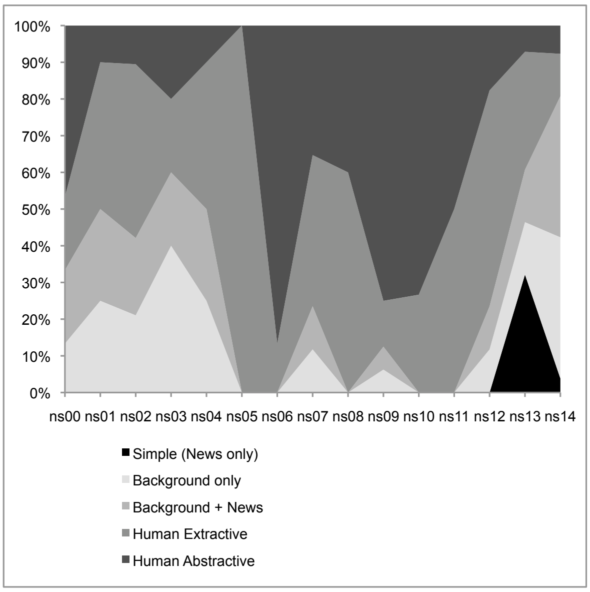


Figure 3: Relative results for each news story (nsnn identifies a news story; stack order is inverse of legend order).

Surprisingly enough (see figures 2 and 3), in general, the extractive human summaries were preferred over the abstractive ones. Moreover, the summaries generated automatically using background information (exclusively or not) were also selected as best summary (over the human created ones) a non-negligible number of times. The poorest performance was attained, as expected, by the simple LSA summarizer, only preferred on two news stories for which all summaries were very similar. The results of the two approaches using background information were very close, a result that can be explained by the fact the summaries generated by these two approaches were equal for 11 of the 15 news stories (in the remaining 4, the average distribution was 31.25% from the news story versus 68.75% from the background information).

Figure 4 further discriminates the results in terms of content and readability.

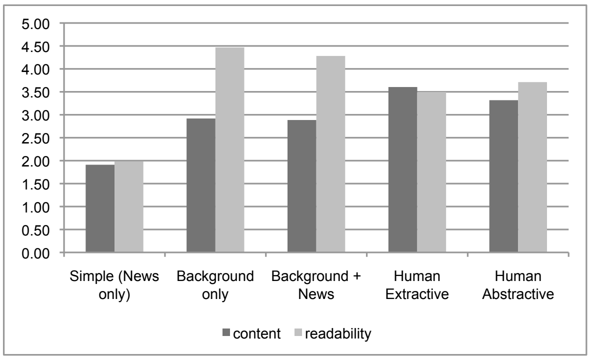


Figure 4: Average of the content and readability scores for each summary creation method.

Regarding content, the results suggest that the choice of the best summary is highly correlated with its content, as the average content scores mimic the overall ones of figure 2. In what concerns readability, the summaries generated using background information achieved the best results. The reasons underlying these results are that the newspaper writing is naturally better planned than speech and that speech transcriptions are affected by the several problems described before (and the original motivation for the work), hence the idea of using them as background information. However, what is odd is that the result obtained by the human abstractive summary creation method is worse than the ones obtained by automatic generation using background information, which could suffer from coherence and cohesion problems. One possible explanation is that the human abstractive summaries tend to mix both informative and indicative styles of summary.

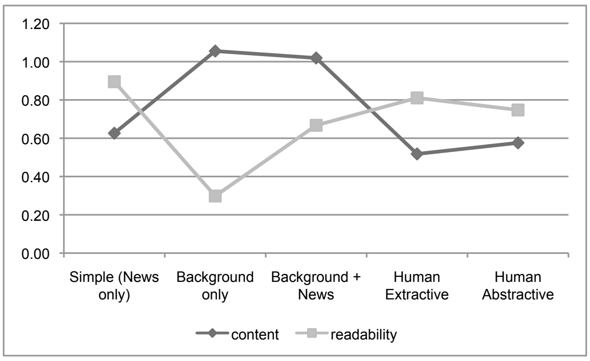


Figure 5: Standard deviation of the content and readability scores.

Figure 5 presents the standard deviation for content and readability scores: concerning content, automatically generated summaries using background information achieved the highest standard deviation scores (see also figure 6 for a sample story). That is in part supported by some commentaries made by the human evaluators on whether a summary should contain information that is not present in the input source. This aspect and the obtained results, suggest that this issue should be further analyzed, possibly using an extrinsic evaluation setup. On the other hand, readability standard deviation scores show that there is a considerable agreement in what concerns this criterion.

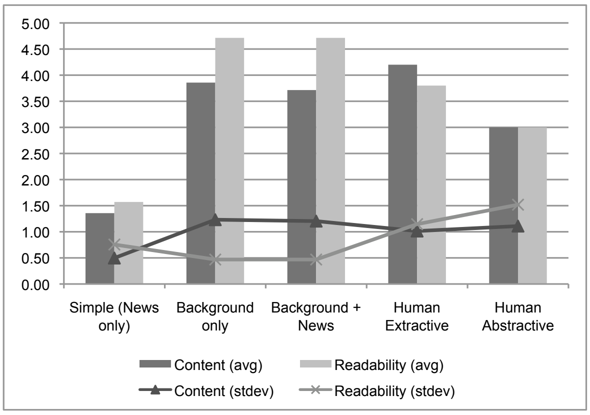


Figure 6: Average and standard deviation of the content and readability scores for one news story.

6 Conclusions

We present a new approach to speech summarization that goes in the direction of the integration of text and speech analysis, as suggested by McKeown et al. (2005). The main idea is the inclusion of related, solid background information to cope with the difficulties of summarizing spoken language and the use of multi-document summarization techniques in single document speech-to-text summarization. In this work, we explore the possibilities offered by phonetic information to select the background information and conducted a perceptual evaluation to assess the relevance of the inclusion of that information.

The results obtained show that the human evaluators preferred human extractive summaries over human abstractive summaries. Moreover, simple LSA summaries attained the poorest results both in terms of content and readability, while human extractive summaries achieved the best performance in what concerns content, and a considerably better performance than simple LSA in what concerns readability. This suggests that it is sill relevant to pursue new methods for relevance estimation. On the other hand, automatically generated summaries using background information were significantly better than simple LSA. This indicates that background information is a viable way to increase the quality of automatic summarization systems.

References

Amaral, R. and I. Trancoso. 2004. Improving the Topic Indexation and Segmentation Modules of a Media Watch System. In *Proceedings of INTERSPEECH 2004 - ICSLP*, pages 1609–1612. ISCA.

Amaral, R., H. Meinedo, D. Caseiro, I. Trancoso, and J. P. Neto. 2006. Automatic vs. Manual Topic Segmentation and Indexation in Broadcast News. In *Proc. of the IV Jornadas en Tecnologia del Habla*.

Amaral, R., H. Meinedo, D. Caseiro, I. Trancoso, and J. P. Neto. 2007. A Prototype System for Selective Dissemination of Broadcast News in European Portuguese. *EURASIP Journal on Advances in Signal Processing*, 2007.

Batista, F., D. Caseiro, N. J. Mamede, and I. Trancoso. 2007. Recovering Punctuation Marks for Automatic Speech Recognition. In *Proceedings of INTERSPEECH 2007*, pages 2153–2156. ISCA.

Carbonell, J. and J. Goldstein. 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *SIGIR 1998: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–336. ACM.

Christensen, H., Y. Gotoh, B. Kolluru, and S. Renals. 2003. Are Extractive Text Summarisation Techniques Portable To Broadcast News? In *Proceedings* *of the IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 489–494. IEEE.

Edmundson, H. P. 1969. New methods in automatic abstracting. *Journal of the Association for Computing Machinery*, 16(2):264–285.

Endres-Niggemeyer, B., J. R. Hobbs, and K. Spärck Jones, editors. 1995. *Summarizing Text for Intelligent Communication—Dagstuhl-Seminar-Report 79*. IBFI.

Furui, S. 2007. Recent Advances in Automatic Speech Summarization. In *Proceedings of the 8^{th} Conference on Recherche d'Information Assistée par Ordinateur (RIAO)*. Centre des Hautes Études Internationales d'Informatique Documentaire.

Gong, Y. and X. Liu. 2001. Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. In *SIGIR 2001: Proceedings of the 24^{st} Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 19–25. ACM.

Hori, T., C. Hori, and Y. Minami. 2003. Speech Summarization using Weighted Finite-State Transducers. In *Proceedings of the 8^{th} EUROSPEECH - INTERSPEECH 2003*, pages 2817–2820. ISCA.

Hovy, E., 2003. *The Oxford Handbook of Computational Linguistics*, chapter Text Summarization, pages 583–598. Oxford University Press.

Kessler, B. 2005. Phonetic comparison algorithms. *Transactions of the Philological Society*, 103(2):243–260.

Kikuchi, T., S. Furui, and C. Hori. 2003. Two-stage Automatic Speech Summarization by Sentence Extraction and Compaction. In *Proceedings of the ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition (SSPR-2003)*, pages 207–210. ISCA.

Lin, C. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81. ACL.

Luhn, H. P. 1958. The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, 2(2):159–165.

Maskey, S. and J. Hirschberg. 2005. Comparing Lexical, Acoustic/Prosodic, Strucural and Discourse Features for Speech Summarization. In *Proceedings of the 9^{th} EUROSPEECH - INTERSPEECH 2005*, pages 621–624. ISCA.

McKeown, K. R. and D. Radev. 1995. Generating Summaries of Multiple News Articles. In *SIGIR 1995: Proceedings of the 18^{th} Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 74–82. ACM.

McKeown, K. R., R. Barzilay, D. Evans, V. Hatzivassiloglou, J. L. Klavans, A. Nenkova, C. Sable, B. Schiffman, and S. Sigelman. 2002. Tracking and Summarizing News on a Daily Basis with Columbia's Newsblaster. In *Proc. of the 2^{nd} International Conference on Human Language Technology Research*, pages 280–285. Morgan Kaufmann.

McKeown, K. R., J. Hirschberg, M. Galley, and S. Maskey. 2005. From Text to Speech Summarization. In *2005 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings*, volume V, pages 997–1000. IEEE.

Mohri, M. 1997. Finite-State Transducers in Language and Speech Processing. *Computational Linguistics*, 23(2):269–311.

Murray, G., S. Renals, and J. Carletta. 2005. Extractive Summarization of Meeting Records. In *Proceedings of the 9^{th} EUROSPEECH - INTERSPEECH 2005*, pages 593–596. ISCA.

Murray, G., S. Renals, J. Carletta, and J. Moore. 2006. Incorporating Speaker and Discourse Features into Speech Summarization. In *Proceedings of the HLT/NAACL*, pages 367–374. ACL.

Nenkova, A. 2006. Summarization Evaluation for Text and Speech: Issues and Approaches. In *Proceedings of INTERSPEECH 2006 - ICSLP*, pages 1527–1530. ISCA.

Paulo, S. and L. C. Oliveira. 2002. Multilevel Annotation Of Speech Signals Using Weighted Finite State Transducers. In *Proc. of the 2002 IEEE Workshop on Speech Synthesis*, pages 111–114. IEEE.

Radev, D., J. Otterbacher, A. Winkel, and S. Blair-Goldensohn. 2005. NewsInEssence: Summarizing Online News Topics. *Communications of the ACM*, 48(10):95–98.

Ribeiro, R. and D. M. de Matos. 2007. Extractive Summarization of Broadcast News: Comparing Strategies for European Portuguese. In *Text, Speech and Dialogue – 10^{th} International Conference. Proceedings*, volume 4629 of *Lecture Notes in Computer Science (Subseries LNAI)*, pages 115–122. Springer.

Spärck Jones, K. 2007. Automatic summarising: The state of the art. *Information Processing and Management*, 43:1449–1481.

Wan, X., J. Yang, and J. Xiao. 2007. CollabSum: Exploiting Multiple Document Clustering for Collaborative Single Document Summarizations. In *SIGIR 2007: Proc. of the 30^{th} Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 143–150. ACM.

Zechner, K. and A. Waibel. 2000. Minimizing Word Error Rate in Textual Summaries of Spoken Language. In *Proceedings of the 1^{st} conference of the North American chapter of the ACL*, pages 186–193. Morgan Kaufmann.

Evaluating automatically generated user-focused multi-document summaries for geo-referenced images

Ahmet Aker
Department of Computer Science
University of Sheffield
Sheffield, S1 4DP, UK
A.Aker@dcs.shef.ac.uk

Robert Gaizauskas
Department of Computer Science
University of Sheffield
Sheffield, S1 4DP, UK
R.Gaizauskas@dcs.shef.ac.uk

Abstract

This paper reports an initial study that aims to assess the viability of a state-of-the-art multi-document summarizer for automatic captioning of geo-referenced images. The automatic captioning procedure requires summarizing multiple web documents that contain information related to images' location. We use SUMMA (Saggion and Gaizauskas, 2005) to generate generic and query-based multi-document summaries and evaluate them using ROUGE evaluation metrics (Lin, 2004) relative to human generated summaries. Results show that, even though query-based summaries perform better than generic ones, they are still not selecting the information that human participants do. In particular, the areas of interest that human summaries display (history, travel information, etc.) are not contained in the query-based summaries. For our future work in automatic image captioning this result suggests that developing the query-based summarizer further and biasing it to account for user-specific requirements will prove worthwhile.

1 Introduction

Retrieving textual information related to a location shown in an image has many potential applications. It could help users gain quick access to the information they seek about a place of interest just by taking its picture. Such textual information could also, for instance, be used by a journalist who is planning to write an article about a building, or by a tourist who seeks further interesting places to visit nearby. In this paper we aim to generate such textual information automatically by utilizing multi-document summarization techniques, where documents to be summarized are web documents that contain information related to the image content. We focus on geo-referenced images, i.e. images tagged with coordinates (latitude and longitude) and compass information, that show things with fixed locations (e.g. buildings, mountains, etc.).

Attempts towards automatic generation of image-related textual information or captions have been previously reported. Deschacht and Moens (2007) and Mori et al. (2000) generate image captions automatically by analyzing image-related text from the immediate context of the image, i.e. existing image captions, surrounding text in HTML documents, text contained in the image, etc. The authors identify named entities and other noun phrases in the image-related text and assign these to the image as captions. Other approaches create image captions by taking into consideration image features as well as image-related text (Westerveld, 2000; Barnard et al., 2003; Pan et al., 2004). These approaches can address all kinds of images, but focus mostly on images of people. They analyze only the immediate textual context of the image on the web and are concerned with describing *what* is in the image only. Consequently, background information about the objects in the image is not provided. Our aim, however, is to have captions that inform users' specific interests about a location, which clearly includes more than just image content description. Multi-document summarization techniques offer the possibility to include image-related information from multiple documents, however, the challenge lies in being able to summarize unrestricted web documents.

Various multi-document summarization tools have been developed: SUMMA (Saggion and Gaizauskas, 2005), MEAD (Radev et al., 2004), CLASSY (Conroy et al., 2005), CATS (Farzinder et al., 2005) and the system of Boros et al. (2001), to name just a few. These systems generate either generic or query-based summaries or both. Generic summaries address a broad readership whereas query-based summaries are preferred by specific groups of people aiming for quick knowledge gain about specific topics (Mani, 2001). SUMMA and MEAD generate both generic and query-based multi-document summaries. Boros et al. (2001) create only generic summaries, while CLASSY and CATS create only query-based summaries from multiple documents. The performance of these tools has been reported for DUC tasks[1]. As Sekine and Nobata (2003) note, although DUC tasks provide a common evaluation standard, they are restricted in topic and are somewhat idealized. For our purposes the summarizer needs to create summaries from unrestricted web input, for which there are no previous performance reports.

For this reason we evaluate the performance of both a generic and a query-based summarizer and use SUMMA which provides both summarization modes. We hypothesize that a query-based summarizer will better address the problem of creating summaries tailored to users' needs. This is because the query itself may contain important hints as to what the user is interested in. A generic summarizer generates summaries based on the topics it observes from the documents and cannot take user specific input into consideration. Using SUMMA, we generate both generic and query-based multi-document summaries of image-related documents obtained from the web. In an online data collection procedure we presented a set of images with related web documents to human subjects and asked them to select from these documents the information that best describes the image. Based on this user information we created model summaries against which we evaluated the automatically generated ones.

Section 2 in this paper describes how image-related documents were collected from the web. In section 3 SUMMA is described in detail. In section 4 we explain how the human image descriptions were collected. Section 5 discusses the results, and section 6 concludes the paper and outlines directions for future work and improvements.

2 Web Document Collection

For web document collection we used geo-referenced images of locations in London such as *Westminster Abbey, London Eye, etc.* The images were taken with a digital SLR camera with a Geotagger plugged-in to its flash slot. The Geotagger helped us to identify the location by means of coordinates of the position where the photographer stands, as well as the direction the camera is pointing (compass information). Based on the coordinates and compass information for each image, we carried out the following steps to collect related documents from the web:

- identify a set of toponyms (terms that denote locations or associate names with locations, e.g. *Westminster Abbey*) that can be passed to a search engine as query terms for document search;
- use a search engine to retrieve HTML documents to be summarized;
- extract the pure text out of the HTML documents.

2.1 Toponym Collection

In order to create the web queries a set of toponyms were collected semi-automatically. We implemented an application (cf. Figure 1) that suggests a list of toponyms close to the photographer's location. The application uses Microsoft's MapPoint[2] service which allows users to query location-related information. For example, a user can query for tourist attractions (interesting buildings, museums, art galleries etc.) close to a location that is identified by its address or its coordinates.

Based on the coordinates (latitude and longitude), important toponyms for a particular image can be queried from the MapPoint database. In order to facilitate this, MapPoint returns a metric that measures the importance of each toponym. A value close to zero means that the returned toponym is closer to the specified coordinates than a toponym with a higher value. For instance for

[1] http://www-nlpir.nist.gov/projects/duc/index.html

[2] http://www.microsoft.com/mappoint/

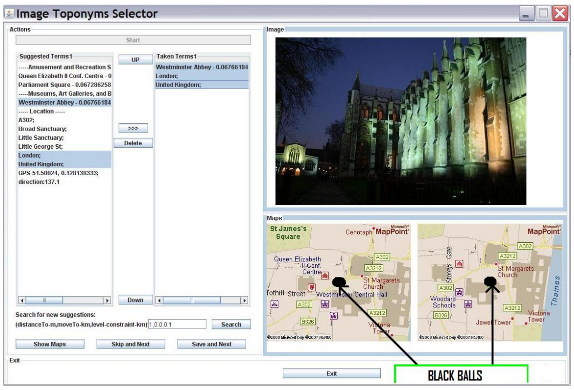


Image Toponyms Selector

Suggested Terms1

-----Amusement and Recreation S
Queen Elizabeth II Conf. Centre - 0
Parliament Square - 0.067286258
-----Museums, Art Galleries, and B
Westminster Abbey - 0.06766184
----- Location -----
A302;
Broad Sanctuary;
Little Sanctuary;
Little George St;
London;
United Kingdom;
GPS-51.50024,-0.128138333;
direction:137.1

Taken Terms1

Westminster Abbey - 0.06766184
London;
United Kingdom;

Search for new suggestions:
(distanceTo-m,moveTo-km,level-constraint-km): 1,0,0,0.1

Figure 1: Image Toponym Collector: Westminster Abbey, Lat: 51.50024 Lon: -0.128138333: Direction: 137.1

the image of *Westminster Abbey* shown in the *Image* box of Figure 1 the following toponyms are collected:

```
Queen Elizabeth II Conf. Centre: 0.059
Parliament Square: 0.067
Westminster Abbey: 0.067
```

The photographer's location is shown with a black dot on the first map in the *Maps* box of Figure 1. The application suggests the toponyms shown in the *Suggested Terms* list.

Knowing the direction the photographer was facing helps us to select the correct toponyms from the list of suggested toponyms. The current MapPoint implementation does not allow an arrow to be drawn on the map which would be the best indication of the direction the photographer is facing. To overcome this problem we create a second map (cf. *Maps* box of Figure 1) that shows another dot moved 50 meters in the compass direction. By following the dot from the first map to the second map we can determine the direction the photographer is facing. When the direction is known, it is certain that the image shows *Westminster Abbey* and not the *Queen Elizabeth II Conf. Centre* or *Parliament Square*. The *Queen Elizabeth II Conf. Centre* is behind the photographer and *Parliament Square* is on the left hand side.

Consequently in this example the toponym *Westminster Abbey* is selected manually for the web search. In order to avoid ambiguities, the city name and the country name (also generated by MapPoint) are added manually to the selected toponyms. Hence, for *Westminster Abbey*, *London* and *United Kingdom* are added to the toponym list. Finally the terms in the toponym list are simply separated by a boolean *AND* operator to form the web query. Then, the query is passed to the search engine as described in the next section.

2.2 Document Query and Text Extraction

The web queries were passed to the Google Search engine and the 20 best search results were retrieved, from which only 11 were taken for the summarization process. We ensure that these 20 search results are healthy hyperlinks, i.e. that the content of the hyperlink is accessible. In addition to this, multiple hyperlinks belonging to the same domain are ignored as it is assumed that the content obtained from the same domain would be similar. Each remaining search result is crawled to obtain its content.

The web-crawler downloads only the content of the document residing under the hyperlink, which was previously found as a search result, and does not follow any other hyperlinks within the document. The content obtained by the web-crawler encapsulates an HTML structured document. We further process this using an HTML parser[3] to select the *pure* text, i.e. text consisting of sentences.

The HTML parser removes advertisements, menu items, tables, java scripts etc. from the HTML documents and keeps sentences which contain at least 4 words. This number was chosen after several experiments. The resulting data is passed on to the multi-document summarizer which is described in the next section.

3 SUMMA

SUMMA[4] is a set of language and processing resources to create and evaluate summarization systems (single document, multi-document, multilingual). The components can be used within GATE[5] to produce ready summarization applications. SUMMA has been used in this work to create an extractive multi-document summarizer: both generic and query-based.

In the case of generic summarization SUMMA uses a single cluster approach to summarize *n* related documents which are given as input. Using GATE, SUMMA first applies sentence detection and sentence tokenisation to the given documents. Then each sentence in the documents is represented as a vector in a vector space model (Salton, 1988), where each vector position contains a term

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

[4] http://www.dcs.shef.ac.uk/ saggion/summa/default.htm

[5] http://gate.ac.uk

(word) and a value which is a product of the *term frequency* in the document and the *inverse document frequency (IDF)*, a measurement of the term's distribution over the set of documents (Salton and Buckley, 1988). Furthermore, SUMMA enhances the sentence vector representation with further features such as the sentence position in its document and the sentence similarity to the lead-part in its document. In addition to computing the vector representation for all sentences in the document collection the centroid of this sentence representation is also computed.

In the sentence selection process, each sentence in the collection is ranked individually, and the top sentences are chosen to build up the final summary. The ranking of a sentence depends on its distance to the centroid, its absolute position in its document and its similarity to the lead-part of its document. For calculating vector similarities, the cosine similarity measure is used (Salton and Lesk, 1968).

In the case of the query-based approach, SUMMA adds an additional feature to the sentence vector representation as computed for generic summarization. For each sentence, cosine similarity to the given query is computed and added to the sentence vector representation. Finally, the sentences are scored by summing all features in the vector space model according to the following formula:

$$Sentence_{score} = \sum_{i=1}^{n} feature_i * weight_i$$

After the scoring process, SUMMA starts selecting sentences for summary generation. In both generic and query-based summarization, the summary is constructed by first selecting the sentence that has the highest score, followed by the next sentence with the second highest score until the compression rate is reached. However, before a sentence is selected a similarity metric for redundancy detection is applied to each sentence which decides whether a sentence is distinct enough from already selected sentences to be included in the summary or not. SUMMA uses the following formula to compute the similarity between two sentences:

$$NGramSim(S_1, S_2, n) = \sum_{j=1}^{n} w_j * \frac{grams(S_1, j) \bigcap grams(S_2, j)}{grams(S_1, j) \bigcup grams(S_2, j)}$$

where n specifies maximum size of the n-grams to be considered, grams(S_X, j) is the set of j-grams in sentence X and w_j is the weight associated with j-gram similarity. Two sentences are similar if *NGramSim*(S_1, S_2, *n*) $> \alpha$. In this work n is set to 4 and α to 0.1. For j-gram similarity weights $w_1 = 0.1$, $w_2 = 0.2$, $w_3 = 0.3$ and $w_4 = 0.4$ are selected. These values are coded in SUMMA as defaults.

Using SUMMA, generic and query-based summaries are generated for the image-related documents obtained from the web. Each summary contains a maximum of 200 words. The queries used in the query-based mode are toponyms collected as described in section 2.1.

4 Creating Model Summaries

For evaluating automatically generated summaries as image captions, information that people associate with images is collected. For this purpose, an online data collection procedure was set up. Participants were provided with a set of 24 images. Each image had a detailed map showing the location where it was taken, along with URLs to 11 related documents which were used for the automated summarization. Figure 2 shows an example of an image and Table 2 contains the corresponding related information.

Each participant was asked to familiarize him- or herself with the location of the image by analyzing the map and going through all 11 URLs. Then each participant decided on up to 5 different pieces of information he/she would like to know if he/she sees the image or information about something he/she relates with the image. The information we collected in this way is similar to 'information nuggets' (Voorhees, 2003). Information nuggets are facts which help us assess automatic summaries by checking whether the summary contains the fact or not. In addition to this, each participant was asked to collect the information only from the given documents, ignoring any other links in these documents.

Eleven students participated in this survey, simulating the scenario in which tourists look for information about an image of a popular sight. The number of images annotated by each participant is shown in Table 1.

The participants selected the information from original HTML documents on the web and not from the documents which were preprocessed for the multi-document summarization task. We found

Table 1: Number of images annotated by each particant

User1	User2	User3	User4	User5	User6	User7	User8	User9	User10	User11
24	7	24	24	18	24	8	4	16	12	24

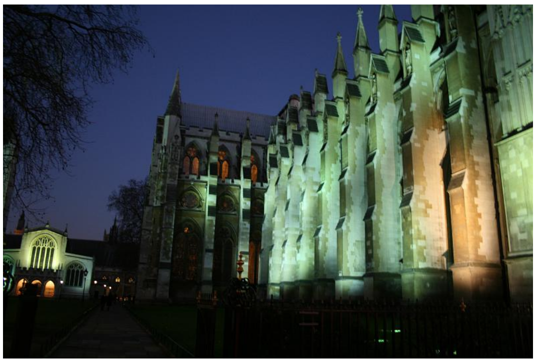


Figure 2: Example image

Table 2: Information related to Figure 2

1. Westminster Abbey is the place of the coronation, marriage and burial of British monarchs, except Edward V and Edward VIII since 1066

2. the parish church of the Royal Family

3. the centrepiece to the City of Westminster

4. first church on the site is believed to have been constructed around the year 700

5. The history and the monuments, crypts and memorials are not to be missed.

out that in some cases the participants selected information that did not occur in the preprocessed documents. To ensure that the information selected by the participants also occurs in the preprocessed documents, we retained only the information selected by the participants that could also be found in these documents, i.e. that was available to the summarizer. Out of 807 nuggets selected by participants 21 (2.6%) were not found in the documents available to the summarizer and were removed.

Furthermore, as the example above shows (cf. Table 2), not all the items of information selected by the participants were in form of full sentences. They vary from phrases to whole sentences. The participants were free to select any text unit from the documents that they related to the image content. However, SUMMA works extractively and its summaries contain only sentences selected from the given input documents. The user selected information was normalized to sentences in order to have comparable summaries for evaluation. This was achieved by selecting the sentence(s) from the documents in which the participant-selected information was found and replacing the participant-selected phrases or clauses with the full sentence(s). In this way model summaries were obtained.

5 Results

The model summaries were compared against 24 summaries generated automatically using SUMMA by calculating ROUGE-1 to ROUGE-4, ROUGE-L and ROUGE-W-1.2 recall metrics (Lin, 2004). For all these metrics ROUGE compares each automatically generated summary s pairwise to every model summary m_i from the set of M model summaries and takes the maximum $ROUGE_{Score}$ value among all pairwise comparisons as the best $ROUGE_{Score}$ score:

$$ROUGE_{Score} = argmax_i ROUGE_{Score}(m_i, s)$$

ROUGE repeats this comparison M times. In each iteration it applies the Jackknife method and takes one model summary from the M model summaries away and compares the automatically generated summary s against the $M - 1$ model summaries. In each iteration one best $ROUGE_{Score}$ is calculated. The final $ROUGE_{Score}$ is then the average of all best scores calculated in M iterations.

In this way each generic and query-based summary was compared with the corresponding model summaries. The results are given in the first two columns of Table 3. We also collected the common information all participants selected for a particular image and compared this to the corresponding query-based summary. The common information is the intersection set of the sets of information each of the participants selected for a particular image. The results for this comparison are shown in column *QueryToCPOfModel* of Table 3.

The model summaries were also compared against each other in order to assess the agreement between the participants. To achieve this, the image information selected by each participant was compared against the rest. The corresponding results are shown in column *UserToUser* of Table 4. We applied the same pairwise comparison we used for our model summaries to the model summaries of task 5 in DUC 2004 in order to mea-

Table 3: Comparison: Automatically generated summaries against model summaries. The column GenericToModel for example shows ROUGE results for generic summaries relative to model summaries. CP stands for common part, i.e. common information selected by all participants.

Recall	GenericToModel	QueryToModel	QueryToCPOfModel	QueryToModelInDUC
R-1	0.38293	0.39655	0.22084	0.3341
R-2	0.14760	0.17266	0.09894	0.0723
R-3	0.09286	0.11196	0.06222	0.0279
R-4	0.07450	0.09219	0.04971	0.0131
R-L	0.34437	0.35837	0.20913	0.3320
R-W-1.2	0.11821	0.12606	0.06350	0.1130

Table 4: Comparison: Model summaries against each other

Recall	UserToUser	UserToUserInDUC
R-1	0.42765	0.45407
R-2	0.30091	0.13820
R-3	0.26338	0.05870
R-4	0.24964	0.02950
R-L	0.40403	0.41594
R-W-1.2	0.15846	0.13973

sure the agreements between the participants on this standard task. This gives us a benchmark relative to which we can assess how well users agree on what information should be related to images. The results for this comparison are shown in column *UserToUserInDUC* of Table 4.

All ROUGE metrics except R-1 and R-L indicate higher agreement in human image-related summaries than in DUC document summaries. The ROUGE metrics most indicative of agreement between human summaries are those that best capture words occurring in longer sequences of words immediately following each other (R-2, R-3, R-4 and R-W). If long word sequences are identical in two summaries it is more likely that they belong to the same sentence than if only single words are common, as captured by R-1, or sequences of words that do not immediately follow each other, as captured by R-L. In R-L gaps in word sequences are ignored so that for instance *A B C D G* and *A E B F C K D* have the common sequence *A B C D* according to R-L. R-W considers the gaps in words sequences so that this sequence would not be recognized as common. Therefore the agreement on our image-related human summaries is substantially higher than agreement on DUC document human summaries.

The results in Table 3 support our hypothesis that query-based summaries will perform better than generic ones on image-related summaries. All ROUGE results of the query-based summaries are greater than the generic summary scores. This reinforces our decision to focus on query-based summaries in order to create image-related summaries which also satisfy the users' needs. However, even though the query-based summaries are more appropriate for our purposes, they are not completely satisfactory. The query-based summaries cover only 39% of the unigrams (ROUGE 1) in the model summaries and only 17% of the bigrams (ROUGE 2), while the model summaries have 42% agreement in unigrams and 30% agreement in bigrams (cf. column *UserToUser* in Table 4). The agreement between the query-based and model summaries gets lower for ROUGE-3 and ROUGE-4 indicating that the query-based summaries contain very little information in common with the participants' results. This indication is supported by the ROUGE-L (35%) and the low ROUGE-W (12%) agreement which are substantially lower compared to the *UserToUser* ROUGE-L (40%) and ROUGE-W (15%) and the low ROUGE scores in column *QueryToCPOfModel*. For comparison with automated summaries in a different domain, we include ROUGE scores of query based SUMMA used in DUC 2004 (Saggion and Gaizauskas, 2005) as shown in the last column of Table 3. All scores are lower than our *QueryToModel* results which might be due to low agreement between human generated summaries for the DUC task (cf. *UserToUserInDUC* column in Table 4) or maybe because image captioning is an easier task. The possibility that our summarization task is easier than DUC due to the summarizer having fewer documents to summarize or due to the documents being shorter than those in the DUC task can be excluded. In the DUC task the multi-document clusters contain 10 documents on average while our summarizer works with 11 documents. The mean length in documents in DUC

Table 5: Query-based summary for Westminster Abbey and information selected by participants

Query-based summary	Information selected by participants
The City of London has St Pauls, but Westminster Abbey is the centrepiece to the City of Westminster. Westminster Abbey should be at the top of any London traveler's list. Westminster Abbey, however, lacks the clear lines of a Rayonnant church,... I loved Westminster Abbey on my trip to London. **Westminster Abbey was rebuilt after 1245 by Henry III's order, and in 1258 the remodeling of the east end of St. Paul's Cathedral began.** He was interred in Westminster Abbey. From 1674 to 1678 he tuned the organ at Westminster Abbey and was employed there in 1675-76 to copy organ parts of anthems. The architectural carving found at Westminster Abbey (mainly of the 1250s) has much of the daintiness of contemporary French work, although the drapery is still more like that of the early Chartres or Wells sculpture than that of the Joseph Master. Nevertheless, Westminster Abbey is something to see if you have not seen it before. I happened upon the Westminster Abbey on an outing to Parliament and Big Ben.	**1.***(3)* Westminster Abbey is the place of the coronation, marriage and burial of British monarchs, except Edward V and Edward VIII since 1066. **2.***(1)* **What is unknown, however is just how old it is. The first church on the site is believed to have been constructed around the year 700**. **3.***(2)* Standing as it does between Westminster Abbey and the Houses of Parliament, and commonly called "the parish church of the House of Commons", St Margaret's has witnessed many important events in the life of this country. **4.***(1)* In addition, the Abbey is the parish church of the Royal Family, when in residence at Buckingham Palace. **5.***(1)* The history and the monuments, crypts and memorials are not to be missed. **6.***(1)* For almost one thousand years, Westminister Abbey has been the setting for much of London's ceremonies such as Royal Weddings, Coronations, and Funeral Services. **7.***(1)* It is also where many visitors pay pilgrimage to The Tomb of the Unknown Soldier. **8.***(1)* The City of London has St Pauls, but Westminster Abbey is the centrepiece to the City of Westminster.

is 23 sentences while our documents have 44 sentences on average.

Table 5 shows an example query-based summary for the image of *Westminster Abbey* and the information participants selected for this particular image. Jointly the participants have selected 8 different pieces of information as indicated by the bold numbers in the table. The numbers in parentheses show the number of times that a particular information unit was selected. By comparing the two sides it can be seen that the query-based summary does not cover most of the information from the list with the exception of item 2. The item 2 is semantically related to the sentence in bold on the summary side as it addresses the year the abbey was built, but the information contained in the two descriptions is different.

Our results have confirmed our hypothesis that query-based summaries will better address the aim of this research, which is to get summaries tailored to users' needs. A generic summary does not take the user query into consideration and generates summaries based on the topics it observes. For a set of documents containing mainly historical and little location-related information, a generic summary will probably contain a higher number of history-related than location-related sentences. This might satisfy a group of people seeking historical information, however, it might not be interesting for a group who want to look for location-related information. Therefore using a query-based multi-document summarizer is more appropriate for image-related summaries than a generic one. However, the results of the query-based summaries show that even so they only cover a small part of the information the users select. One reason for this is that the query-based summarizer takes relevant sentences according to the query given to it and does not take into more general consideration the information likely to be relevant to the user. However, we can assume that users will have shared interests in some of the information they would like to get about a particular type of object in an image (e.g. a bridge, church etc.). This assumption is supported by the high agreement between participants' performances in our online survey (cf. column *UserToUser* of Table 4).

Therefore, one way to improve the performance of the query-based summarizer is to give the summarizer the information that users typically associate with a particular object type as input and bias the multi-document summarizer towards this information. To do this we plan to build models of user preferences for different object types from the large number of existing image captions from web resources, which we believe will improve the quality of automatically generated captions.

6 Conclusion

In this work we showed that query-based summarizers perform slightly better than generic summarizers on an image captioning task. However, their output is not completely satisfactory when compared to what human participants indicated as important in our data collection study. Our future work will concentrate on extending the query-based summarizer to improve its performance in generating captions that match user expectations regarding specific image types. This will include collecting a large number of existing captions from web sources and applying machine learning techniques for building models of the kinds of information that people use for captioning. Further work also needs to be carried out on improving the readability of the extractive caption summaries.

7 Acknowledgement

This work is supported by the EU-funded TRIPOD project [6]. We would like to thank Horacio Saggion for his support with SUMMA. We are also grateful to Emina Kurtic, Mark Sanderson, Mesude Bicak and Dilan Paranavithana for comments on the previous versions of this paper.

References

Barnard, Kobus and Duygulu, Pinar and Forsyth, David and de Freitas, Nando and Blei M, David and Jordan I, Michael. 2003. *Matching words and pictures. The Journal of Machine Learning Research, MIT Press Cambridge, MA, USA*, 3: 1107–1135.

Boros, Endre and Kantor B, Paul and Neu j, David. 2001. *A Clustering Based Approach to Creating Multi-Document Summaries. Proc. of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.*

Conroy M, John and Schlesinger D, Judith and Stewart G, Jade 2005. *CLASSY query-based multi-document summarization. Proc. of the 2005 Document Understanding Workshop, Boston.*

Deschacht, Koen and Moens F, Marie. 2007. *Text Analysis for Automatic Image Annotation. Proc. of the 45th Annual Meeting of the Association for Computational Linguistics, Prague.*

Farzindar, Atefeh and Rozon, Frederik and Lapalme, Guy. 2005. *CATS a topic-oriented multi-document summarization system at DUC 2005. Proc. of the 2005 Document Understanding Workshop (DUC2005).*

Lin, Chin-Yew 2004. *ROUGE: A Package for Automatic Evaluation of Summaries. Proc. of the Workshop on Text Summarization Branches Out (WAS 2004).*

Mani, Inderjeet. 2001. *Automatic Summarization. John Benjamins Publishing Company.*

Mori, Yasuhide and Takahashi, Hironobu and Oka, Ryuichi. 2000. *Automatic word assignment to images based on image division and vector quantization. Proc. of RIAO 2000: Content-Based Multimedia Information Access.*

Pan, Jia-Yu. and Yang, Hyung-Jeong and Duygulu, Pinar and Faloutsos, Christos. 2004. *Automatic image captioning. Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on.*

Radev R, Dragomir. and Jing, Hongyan and Styś, Malgorzata and Tam, Daniel. 2004. *Centroid-based summarization of multiple documents. Information Processing and Management*,40(6): 919–938.

Saggion, Horacio and Gaizauskas, Robert 2004. *Multi-document summarization by cluster/profile relevance and redundancy removal. Document Understanding Conference (DUC04).*

Salton, Gerhard 1988. *Automatic text processing. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.*

Salton, Gerhard and Buckley, Chris 1988. *Term-weighting approaches in automatic text retrieval. Pergamon Press, Inc. Tarrytown, NY, USA.*

Salton, Gerhard and Lesk E., Michael 1968. *Computer Evaluation of Indexing and Text Processing. Journal of the ACM*,15(1):8–36.

Sekine, Satoshi and Nobata, Chikashi. 2003. *A Survey for Multi-Document Summarization. Association for Computational Linguistics Morristown, NJ, USA, Proc. of the HLT-NAACL 03 on Text summarization workshop-Volume 5.*

Voorhees M, Ellen. 2003. *Overview of the TREC 2003 Question Answering Track. Proc. of the Twelfth Text REtrieval Conference (TREC 2003).*

Westerveld, Thijs. 2000. *Image retrieval: Content versus context. Content-Based Multimedia Information Access, RIAO 2000 Conference.*

[6] http://tripod.shef.ac.uk/

Story tracking: linking similar news over time and across languages

Bruno Pouliquen & Ralf Steinberger
European Commission
Joint Research Centre
Via E. Fermi 2749, 21027 Ispra, Italy
Firstname.Lastname@jrc.it

Olivier Deguernel
Temis S.A.
Tour Gamma B, 193-197 rue de Bercy
75582 Paris Cedex, France
Olivier.Deguernel@temis.com

Abstract

The *Europe Media Monitor* system (EMM) gathers and aggregates an average of 50,000 newspaper articles per day in over 40 languages. To manage the *information overflow*, it was decided to group similar articles per day and per language into clusters and to link daily clusters over time into *stories*. A story automatically comes into existence when related groups of articles occur within a 7-day window. While cross-lingual links across 19 languages for individual news clusters have been displayed since 2004 as part of a freely accessible online application (http://press.jrc.it/NewsExplorer), the newest development is work on linking entire stories across languages. The evaluation of the monolingual aggregation of historical clusters into stories and of the linking of stories across languages yielded mostly satisfying results.

1 Introduction

Large amounts of information are published daily on news web portals around the world. Presenting the most important news on simple, newspaper-like pages is enough when the user wants to be informed about the latest news. However, such websites do not provide a long-term view on how any given story or event developed over time. Our objective is to provide users with a fully automatic tool that groups individual news articles every day into *clusters* of related news and to aggregate the daily clusters into stories, by linking them to the related ones identified in the previous weeks and months. In our jargon, *stories* are thus groups of articles talking about a similar event or theme *over time*. We work with the daily clusters computed by the NewsExplorer application (Pouliquen et al. 2004). For each daily cluster in currently nineteen languages, the similarity to all clusters produced during the previous seven days is computed and a link is established if the similarity is above a certain threshold. It is on the basis of these individual links that stories are built, i.e. longer chains of news clusters related over time. The current NewsExplorer application additionally identifies for all news clusters, whether there are related clusters in the other languages. These daily cross-lingual links are used to link the longer-lasting stories across languages.

After a review of related work (Section 12), we will present the *Europe Media Monitor* (EMM) system and its NewsExplorer application (section 3). We will then provide details on the process to build the multi-monolingual stories (Section 4) and on the more recent work on linking stories across languages (Section 5). Section 6 presents evaluation results both for the monolingual story compilation and for the establishment of cross-lingual links. Section 7 concludes and points to future work.

2 Related work

The presented work falls into the two fields of *Topic Detection and Tracking* and cross-lingual document similarity calculation.

2.1 Topic detection and tracking (TDT)

TDT was promoted and meticulously defined by the US-American DARPA programme (see Wayne 2000). An example explaining the TDT concept was that of the Oklahoma City bombing in 1995, where not only the bombing, but also the related memorial services, investigations, prosecution etc. were supposed to be captured.

Human evaluators will often differ in their opinion whether a given document belongs to a topic or not, especially as 'topic' can be defined broadly (e.g. the Iraq war and the following period of insurgence) or more specifically. For instance, the capture and prosecution of Saddam Hussein, individual roadside bombings and air strikes, or the killing of Al Qaeda leader Abu Musab al-Zarqawi could either be seen as individual topics or as part of the Iraq war. This fuzziness regarding what is a 'topic' makes a formal evaluation rather difficult. Our system is more inclusive and will thus include all the mentioned sub-events into one topic (story). A separate clustering system was developed as part of the *EMM-NewsBrief* (http://press.jrc.it/NewsBrief/), which produces more short-lived and thus more specific historical cluster links.

2.2 Cross-lingual linking of documents

Since 2000, the TDT task was part of the TIDES programme (Translingual Information Detection, Extraction and Summarisation), which focused on cross-lingual information access. The goal of TIDES was to enable English-speaking users to access, correlate and interpret multilingual sources of real-time information and to share the essence of this information with collaborators. The purpose of our own work includes the topic detection and tracking as well as the cross-lingual aspect. Main differences between our own work and TIDES are that we need to monitor more languages, that we are interested in all cross-lingual links (as opposed to targeting only English), and that we use different methods to establish cross-lingual links (see Section 5).

All TDT and TIDES participants used either Machine Translation (MT; e.g. Leek et al. 1999) or bilingual dictionaries (e.g. Wactlar 1999) for the cross-lingual tasks. Performance was always lower for cross-lingual topic tracking (Wayne 2000). An interesting insight was formulated in the "native language hypothesis" by Larkey et al (2004), which states that topic tracking works better in the original language than in (machine-)translated collections. Various participants stated that the usage of named entities helped (Wayne 2000). Taking these insights into account, we always work in the source language and make intensive use of named entities.

Outside TDT, an additional two approaches for linking related documents across languages have been proposed, both of which use bilingual vector space models: Landauer & Littman (1991) used bilingual *Lexical Semantic Analysis* and Vinokourov et al. (2002) used *Kernel Canonical Correlation Analysis*. These and the approaches using MT or bilingual dictionaries have in common that they require bilingual resources and are thus not easily scalable for many language pairs. For N languages, there are `N*(N-1)/2` language pairs (e.g. for 20 languages, there are 190 language pairs and 380 language pair directions). Due to the multilinguality requirement in the European Union (EU) context (there are 23 official EU languages as of 2007), Steinberger et al. (2004) proposed to produce an interlingual document (or document cluster) representation based on named entities (persons, organisations, disambiguated locations), units of measurement, multilingual specialist taxonomies (e.g. medicine), thesauri and other similar resources that may help produce a language-independent document representation. Similarly to Steinberger et al. (2004), the work described in the following sections equally goes beyond the language pair-specific approach, but it does not make use of the whole range of information types.

In Pouliquen et al. (2004), we showed how NewsExplorer links individual news clusters over time and across languages, but without aggregating the clusters into the more compact and high-level representations (which we call *stories*). This new level of abstraction was achieved by exploiting the monolingual and cross-lingual cluster links and by adding additional filtering heuristics to eliminate wrong story candidate clusters. As a result, long-term developments can now be visualised in timelines and users can explore the development of events over long time periods (see Section 4.2). Additionally, meta-information for each story can be compiled automatically, including article and cluster statistics as well as lists of named entities associated to a given story.

2.3 Commercial applications

Compared to commercial or other publicly accessible news analysis and navigation applications, the one presented here is unique in that it is the only one offering automatic linking of news items related either historically or across languages. The news aggregators *Google News* (http://news.google.com) and *Yahoo! News* (http://news.yahoo.com/), for instance, deliver daily news in multiple languages, but do not link the found articles over time or across languages. The monolingual English language applications *DayLife* (http://www.daylife.com/), *SiloBreaker* (http://www.silobreaker.com/), and *NewsVine*

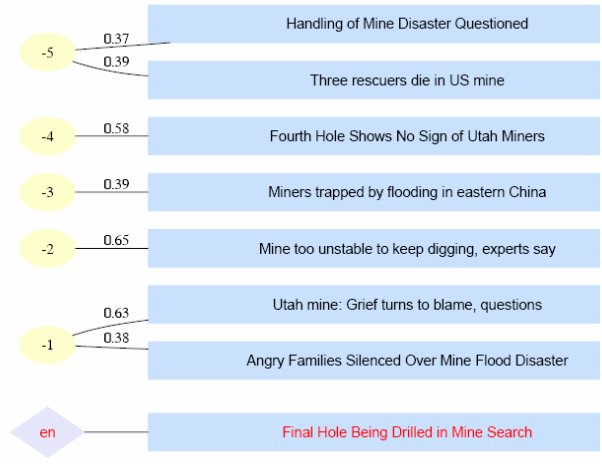

-5
0.37
0.39
Handling of Mine Disaster Questioned
Three rescuers die in US mine
-4
0.58
Fourth Hole Shows No Sign of Utah Miners
-3
0.39
Miners trapped by flooding in eastern China
-2
0.65
Mine too unstable to keep digging, experts say
0.63
0.38
-1
Utah mine: Grief turns to blame, questions
Angry Families Silenced Over Mine Flood Disaster
en
Final Hole Being Drilled in Mine Search

Figure 1. Example of historical links between clusters: The graph shows the cosine similarity between today's English language cluster (*Final hole being drilled …*) and seven clusters identified during five previous days. Only clusters with a similarity above 0.5 will be retained.

(http://www.newsvine.com/) do not link related news over time either. *NewsTin* (http://www.newstin.com) is the only one to offer more languages (ten) and to categorise news into a number of broad categories, but they, again, do not link related news over time or across languages.

3 Europe Media Monitor (EMM) & NewsExplorer

EMM has been gathering multilingual news articles from many different web portals since 2002. It's *NewsBrief* application has since displayed the world's most recent news items on its public web servers (http://emm.jrc.it/overview.html). Every day, and for each of 19 languages separately, EMM's *NewsExplorer* application groups related articles into *clusters*. Clusters are computed using a group average agglomerative bottom-up clustering algorithm (similar to Schultz & Liberman 1999). Each article is represented as a vector of keywords with the keywords being the words of the text (except stop words) and their weight being the log-likelihood value computed using word frequency lists based on several years of news. We additionally enrich the vector space representation of each cluster with country information (see Pouliquen et al., 2004), based on log-likelihood-weighted, automatically recognised and disambiguated location and country names (see Pouliquen et al. 2006).

Each computed daily cluster consists of its keywords (i.e. the average log-likelihood weight for each word) and the title of the cluster's medoid (i.e. the article closest to the centroid of the cluster). In addition we enrich the cluster with features that will be used in further processes. These include the cluster size, lists of persons, organisations, geo-locations and subject domain codes (see Section 5).

When comparing two clusters in the same language, the keywords offer a good representation (especially when the keywords are enriched with the country information). Section 5 will show that the additional ingredients are useful to compare two clusters in different languages.

4 Building stories enriched with meta-information

For each language separately and for each individual cluster of the day, we compute the *cosine* similarity with all clusters of the past 7 days (see Figure 1). Similarity is based on the keywords associated with each cluster. If the similarity between the keyword vectors of two clusters is above the empirically derived threshold of 0.5, clusters are linked. This optimised threshold was established by evaluating cluster linking in several languages (see Pouliquen et al. 2004). A cluster can be linked to several previous clusters, and it can even be linked to two different clusters of the same day.

4.1 Building stories by linking clusters over time

Stories are composed of several clusters. If a new cluster is similar to clusters that are part of a story, it is likely that this new cluster is a continuation of the existing story. For the purpose of building stories, individual and yet unlinked clusters of the previous seven days are treated like (single cluster) stories. If clusters have not been linked to within seven days, they remain individual clusters that are not part of a story. Building stories out of clusters is done using the following incremental algorithm (for a given day):

```
for each cluster c
 for each story s
  score[s]=0;
  for each cluster cp (linked to c)
   if (s: story containing cp) then
    score[s] += (1-score[s])*sim(cp,s);
   endif
  endfor
 endfor
 if (s: story having the maximum score)
 then
  add c to story s (with sim score[s])
 else // not similar to any story
  create new story containing only c
 endif
endfor
```

Lang	Biggest title	Keywords
En	US Airways won't pursue Delta forever	*United states* / **Doug Parker, Delta Airlines** / airways, offer, emerge, grinstein, bid, regulatory, creditors, bankruptcy, atlanta, increased
It	Stop al massacro di balene. Il mondo contro il Giappone	*Australia, N. Zealand, Japan*/ **Greenpeace International, John Howard**/ caccia, megattere, balene, sydney, acqua, mesi, antartico, salti
Es	Mayor operación contra la pornografía infantil en Internet en la historia de España	**Guardia Civil, Fernando Herrero Tejedor** / pornografía, imputados, mayor, cinco, delito, internet, registros, siete, informática, sci
De	Australian Open: "Tommynator" mit Gala-Vorstellung	*Russia, Australia, United states* / **Australian Open, Mischa Zverev** / satz, tennis, deutschen, bozoljac, erstrunden, melbourne, kohlschreiber, Donnerstag
Fr	Il faut aider l'Afrique à se mondialiser, dit Jacques Chirac	**Jacques Chirac, African Union** / afrique, sommet, continent, président, cannes, darfour, état, pays, conférence, chefs, omar

Table 1. Examples of stories, their biggest titles and their corresponding keywords. Countries are displayed in italic, person and organisation names in boldface.

with *sim(cp,s)* being the similarity of the cluster to the story (the first cluster of a story gets a *sim* of 1, the following depend on the *score* computed by the algorithm).

When deciding whether a new cluster should be part of an existing story, the challenge is to combine the similarities of the new cluster with each of the clusters in the story. As stories change over time and the purpose is to link the newest events to existing stories, the new cluster is only compared to the story's clusters of the last 7 days. A seven-day window is intuitive and automatically takes care of fluctuations regarding the number of articles during the week (weekends are quieter). In the algorithm to determine whether the new cluster is linked to the story, the similarity score is computed incrementally: The score is the similarity of the new cluster with the latest cluster of the story (typically yesterday's) plus the similarity of the new cluster with the story's cluster of the day before multiplied with a reducing factor (1-score$_{i-1}$), plus the similarity of the new cluster with the story's cluster of yet another day before multiplied with a reducing factor (1-score$_{i-2}$), etc. The reducing factor helps to keep the similarity score between the theoretical values 0 (unrelated) and 1 (highly related):

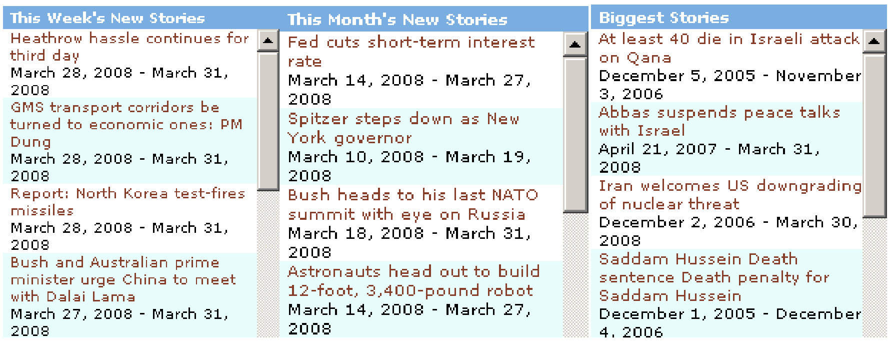
$$score_i = \begin{cases} 0 & (i=0) \\ (1-score_{i-1})\cdot sim(c_i,s) & (0<i<7) \end{cases}$$

If the final score is above the threshold of 0.5, the cluster gets linked to the existing story. Otherwise it remains unlinked. The story building algorithm is language-independent and could thus be applied to all of the 19 NewsExplorer languages. Currently, it is run every day (in sequential order) in the following nine languages: Dutch, English, French, German, Italian, Portuguese, Slovene, Spanish and Swedish.

Out of the daily average of 970 new clusters (average computed for all nine languages over a period of one month), only 281 get linked to an existing story (29%) and 90 contribute to a new story (9%). The remaining 599 clusters (62%) remain unlinked singleton clusters. A small number of stories are very big and go on over a long time. This reflects big media issues such as the Iraq insurgence, the Iran-nuclear negotiations and the Israel-Palestine conflict. The latter is the currently longest story ever (see http://press.jrc.it/NewsExplorer/storyedition/en/RTERadio-5f47a76fe35215964cbab22dcbc88d7b.html).

4.2 Aggregating and displaying information about each story

For each story, daily updated information gets stored in the NewsExplorer knowledge base. This includes (a) the title of the first cluster of the story (i.e. the title of the medoid article of that first cluster); (b) the title of the biggest cluster of the story (i.e. the cluster with most articles); (c) the most frequently mentioned person names in the story (*related people*); (d) the person names most highly associated to the story (*associated people*, see below); (e) the most frequently mentioned other names in the story (mostly organisations, but also events such as *Olympics*, *World War II*, etc.); (f) the countries most frequently referred to in the story (either directly with the country name or indirectly, e.g. by referring to a city in that country); (g) a list of keywords describing the story (see below). This meta-information is exported every day into XML files for display on NewsExplorer. The public web pages display up to 13 keywords, including up to three country names and up to two person or organisation names (see Table 1). To



This Week's New Stories

Heathrow hassle continues for third day
March 28, 2008 - March 31, 2008

GMS transport corridors be turned to economic ones: PM Dung
March 28, 2008 - March 31, 2008

Report: North Korea test-fires missiles
March 28, 2008 - March 31, 2008

Bush and Australian prime minister urge China to meet with Dalai Lama
March 27, 2008 - March 31, 2008

This Month's New Stories

Fed cuts short-term interest rate
March 14, 2008 - March 27, 2008

Spitzer steps down as New York governor
March 10, 2008 - March 19, 2008

Bush heads to his last NATO summit with eye on Russia
March 18, 2008 - March 31, 2008

Astronauts head out to build 12-foot, 3,400-pound robot
March 14, 2008 - March 27, 2008

Biggest Stories

At least 40 die in Israeli attack on Qana
December 5, 2005 - November 3, 2006

Abbas suspends peace talks with Israel
April 21, 2007 - March 31, 2008

Iran welcomes US downgrading of nuclear threat
December 2, 2006 - March 30, 2008

Saddam Hussein Death sentence Death penalty for Saddam Hussein
December 1, 2005 - December 4, 2006

Figure 2. Examples of English language stories, as on the NewsExplorer main page (2.04. 2008).

see examples of all meta-information types for each story, see the NewsExplorer pages.

Stories are currently accessible through three different indexes (see Figure 2): the stories of the week, the stories of the month and the biggest stories (all displayed on the main page of NewsExplorer). The biggest stories are ordered by the number of clusters they contain without any consideration of the beginning date or the end date. The stories of the month present stories that started within the last 30 days, stories of the week those that started within the last seven days.

For each story, a time line graph (a flash application taking an XML export as input) is produced automatically, allowing users to see trends and to navigate and explore the story (Figure 3). While a story can have more than one cluster on a given day, the graph only displays the largest cluster for that day.

The story's keyword signature is computed using the keywords appearing in most of the constituent clusters. If any of the keywords represents a country, it will be displayed first. A filtering function eliminates keywords that are part of one of the selected entities. For instance, if a selected entity is *George W. Bush* and a selected country is *Iraq*, the keywords *Bush*, *George*, *Iraqi*, etc. will not be displayed.

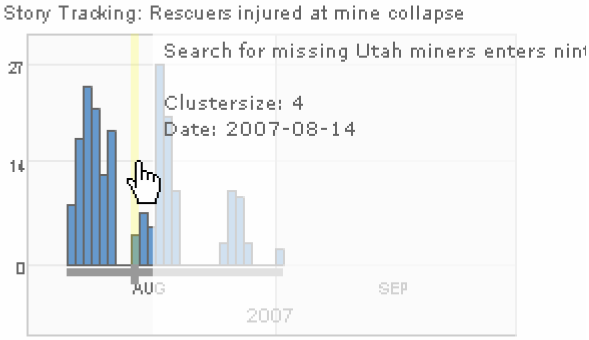


Figure 3. Sample of a short story timeline. When mousing over the graph, title, date and cluster size for that day are displayed. A simple click allows to jump to the relevant cluster, enabling users to explore the story. Available on page http://press.jrc.it/NewsExplorer/storyedition/en/guardian-ee9f870100be631c0147646d29222de9.html.

As mentioned in the previous paragraph, a story's *related entities* are those that have been mentioned most frequently. This typically includes many media VIPs. *Associated entities* are names that appear in this particular story, but are *not* so frequently mentioned in news clusters outside this story, according to the following, TF.IDF-like formula:

$$related(S,e) = \sum_{c_i \in S} fr(c_i,e)$$

$$ass(S,e) = \frac{\sum_{c_i \in S} fr(c_i,e)}{\min(\log(fr(e)),1)} \cdot (1+\log(C(S,e))$$

with *fr(e)* being the number of clusters the entity appears in (in a collection of three years of news) and *C(S,e)* being the number of clusters *in the story S* mentioning the entity. Inversely, the NewsExplorer person and organisation pages also display, for each entity, the biggest stories they are involved in.

5 Cross-lingual cluster and story linking

For each daily cluster in nine NewsExplorer languages, the similarity to clusters in the other 18 languages is computed. To achieve this, we produce three different language-independent vector representations for each cluster (for details, see Pouliquen et al. 2004): a weighted list of Eurovoc subject domain descriptors (*eurov*, available only for EU languages), a frequency list of person and organisation names (*ent*), and a weighted list of direct or indirect references to countries (*geo*). As a fourth ingredient, we also make use of language-dependent keyword lists because even monolingual keywords sometimes match across languages due to cognate words (*cog*), etc. (e.g. *tsunami*, *airlines*, *Tibet* etc.). The overall similarity *clsim* for two clusters *c'* and *c''* in different languages is calculated using a linear combination of the four cosine similarities, using the values for α, β, γ & λ as 0.4, 0.3, 0.2 and 0.1, respectively (see Figure 4):

$$clsim(c',c'') = \alpha \cdot eurov(c',c'') + \beta.geo(c',c'')$$
$$+ \gamma.ent(c',c'') + \lambda.cog(c',c'')$$

5.1 Filtering and refining cross-lingual cluster links

The process described in the previous paragraphs produces some unwanted cross-lingual links. We also observed that not all cross-lingual links are transitive although they should be. We thus developed an additional filtering and link weighting algorithm to improve matters, whose basic idea is the following: When clusters are linked in more than two languages, our assumption is: If cluster A is linked to cluster B and cluster C, then cluster B should also be linked to cluster C. We furthermore assume that if cluster B is not linked to cluster C, then cluster B is less likely to be linked to cluster A. The new algorithm thus checks these 'inter-links' and calculates a new similarity value which combines the standard similarity (described in 5.0) with the number of inter-links. The formula punishes links to an isolated cluster (i.e. links to a target language cluster which itself is not linked to other linked languages) and raises the score for inter-linked clusters (i.e. links to a target language cluster which itself is linked to other linked languages). The new similarity score uses the formula:

$$clsim'(C',C'') = clsim(C',C'').\frac{Cl(C')}{\sqrt{El(C')}}$$

with *Cl*(*C*) being the number of computed cross-lingual links and *El*(C) being the number of expected cross-links (i.e. all cross-language links observed when looking at all languages). For instance, if a cluster is linked to three languages and these are linked to a further three, then *Cl(C')*=3 and *El(C')*=6.

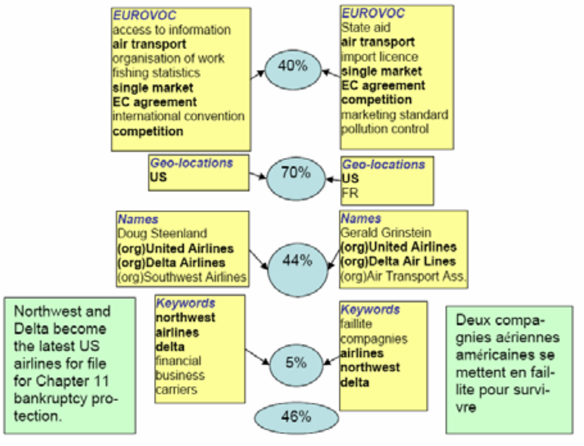


Figure 4. Example of the similarity calculation for an English and a French cluster. The overall similarity for these two clusters, based on the linear combination of four different vectors, is 0.46.

5.2 Linking whole stories across languages

The stories contain clusters which are themselves linked to clusters in other languages (see 5.1). This information can be used to compute the similarity between two whole stories in different languages. The formula is quite simple:

$$Sclsim(S',S'') = \sum_{c_i' \in S', c_j'' \in S''} clsim'(c_i',c_j'')$$

with *S'* and *S''* being two stories in different languages, and *c'* and *c''* being constituent clusters. Cross-lingual cluster similarity values are only added if they are above the threshold of 0.15. Table 2 shows an English story and its links in seven languages.

As the evaluation results in Section 6 show, this formula produces reasonable results, but it has some limitations. Firstly, it relies exclusively on

Lang.	Biggest title	Nb. of clusters	Nb. of articles	Common clusters	Similarity
En	Rescuers injured at mine collapse	17	200	---	---
Pt	EUA: mineiros presos numa mina continuam incontactáveis	12	63	7	2.1363
Es	Colapsa mina en EE.UU.	5	24	3	0.9138
De	USA: Sechs Bergleute eingeschlossen	3	28	2	0.7672
Nl	Mijnwerkers vast na aardbeving in Utah	2	7	2	0.6082
Fr	Le sauvetage de mineurs dans l'Utah tourne au drame	3	16	2	0.5541
Nl	Reddingswerkers omgekomen in mijn Utah	2	12	2	0.4644
Sv	Mystisk "ubåt" undersöks i New York	4	16	2	0.3681

Table 2. Example of cross-lingual links between the English language *US mine collapse* story and stories in seven other languages. The Swedish story, which has the lowest similarity score, is actually unrelated.

daily cross-lingual links, whereas stories are not necessarily reported on the same day across languages. Secondly, we might be able to produce better results by making use of the available meta-information *at story level* described in Section 4.2. We are thus planning to refine this formula in future work.

6 Evaluation

Evaluating such a system is not straightforward as there is a lot of room for interpretation regarding the relatedness of clusters and stories. Cluster consistency evaluation and the monolingual and cross-lingual linking of individual clusters using a very similar approach has already been evaluated in Pouliquen et al. (2004).

In order to evaluate the precision for the story building in four languages, we have evaluated the relatedness of the individual components (the clusters) with the story itself. We compiled a list of 330 randomly selected stories (in the 4 languages English, German, Italian and Spanish) and asked an expert to judge if each of the clusters is linked to the main story. For each story, we thus have a ratio of 'correctly linked' clusters (see Table 3). The average ratio corresponds to the precision of the story tracking system. There clearly is room for improvement, but we found the results good enough to display the automatically identified stories as part of the live application.

We did make an attempt at evaluating also the recall for story building, but soon found out that the results would not make sense. The idea was to carry out a usage-oriented evaluation for the situation in which users are looking for any story of their choice using their own search words (e.g. *Oscar* and *nomination*, *Pavarotti* and *death*, etc.). It was found that relevant stories did indeed exist for almost every query. However, the results would entirely depend on the type of story the evaluator is looking for and on the evaluator's capacity to identify significant search words. We can thus not present results for the recall evaluation of the story tracking system.

Language	Number of stories	Correct components	All components	Precision
German	93	249	265	0.94
English	113	490	570	0.86
Spanish	33	78	91	0.86
Italian	91	239	299	0.80
All	330	1056	1225	0.86

Table 3. Evaluation of the monolingual linking of clusters into stories for four languages.

Type of story	Number of stories	Nb of correct cross-lingual links	Number of cross-lingual links	Precision
All stories	112	275	465	0.59
Stories containing at least 5 clusters	39	145	232	0.62
Stories containing at least 10 clusters	11	75	100	0.75
10 top stories in 4 languages	40	235	270	0.87

Table 4. Evaluation of cross-lingual story linking.

The purpose of a second test was to evaluate the accuracy of the cross-lingual story linking. For that purpose, we evaluated those 112 multilingual stories out of the 330 stories in the previous experiment that had cross-lingual links to any of the languages Dutch, English, French, German, Italian, Portuguese, Spanish or Swedish. Table 4 shows that only 59% of the automatically established cross-lingual story links were accurate, but that the situation improves when looking at stories consisting of more clusters, i.e. 5 or 10. This trend was confirmed by a separate study evaluating only the cross-lingual links for the 10 largest stories in the same four languages, into the same eight other languages: 87% of the cross-lingual links were correct. Note that – for these large stories – the cross-lingual links were 96.5% complete (270 out of 280 possible links were present). Further insights from this evaluation are that there are only two out of the 40 top stories that should be merged (there are two English top stories on Israel) and that there is one cluster in each of the four languages which should be split (all China-related news merges into one story). It is clear that more experiments are needed to improve the cross-lingual links for smaller stories. We have not evaluated the recall of the cross-lingual story linking as recall evaluation is very time-consuming and we first want to optimise the algorithm.

7 Conclusion and Future Work

The story tracking system has been running for two years. There is definitely space for improvement as unrelated clusters are sometimes part of a story, but informal positive user feedback makes us believe that users already find the current results useful. An analysis of the web logs shows that more than 400 separate visitors per day look at story-related information, split quite evenly across the different languages (Table 5).

The story tracking algorithm is rather sensitive to the starting date for the process: Different starting dates may result in different stories and certain starting dates may result in having two separate parallel stories talking about very closely related subjects. Another issue is the seven-day window: We may want to extend the window as it happens occasionally that a story ‘dies’ because no related articles are published on the subject for a week, and that another story talking about the same subject starts 8 days later. Finally, our algorithm should try to cope with the fact that stories can split or merge (an issue not currently dealt with), but this is a non-trivial issue.

Regarding the cross-lingual linking, the current results are encouraging, but not sufficient. The accuracy needs to be improved before the results can go online. The most promising idea here is to make use of each story’s meta-information (lists of related persons, organisations, countries and keywords at story level) and to allow a time delay in the publication of stories across languages. However, the application has high potential, as it will provide users with (graphically visualisable) information on how the media report events across languages and countries.

In a separate effort, a ‘live’ news clustering system has been developed within EMM, which groups the news as they come in during the day (see http://press.jrc.it/NewsBrief/). This process needs to be integrated with the daily and more long-term story tracking process so that users can explore the history and the background for current events.

Acknowledgements

We thank the *Web Mining and Intelligence* team and our team leader Erik van der Goot for the valuable news data and the robust web sites. A special thanks to Jenya Belyaeva for her evaluation.

			Hits/		Visits	
Lang	Hits	Pct	day	Visits	/day	Pct
De	59993	14%	2143	1611	58	13%
En	164557	38%	5877	2273	81	19%
Es	49360	11%	1763	1431	51	12%
Fr	56023	13%	2001	1514	54	12%
It	29445	7%	1052	1425	51	12%
Nl	25175	6%	899	1242	44	10%
Pt	42933	10%	1533	2170	78	18%
Sv	7284	2%	260	575	21	5%
Total:	434770		15527	12241	437	

Table 5. Number of connections to *story-related* NewsExplorer web pages only, and distribution per language (period 1-28/06/2008). Only visits from different IP addresses were counted.

References

Landauer Thomas & Michael Littman (1991). A Statistical Method for Language-Independent Representation of the Topical Content of Text Segments. Proceedings of the 11th International Conference ‘Expert Systems and Their Applications’, vol. 8: pp. 77-85.

Larkey Leah, Fangfang Feng, Margaret Connell, Victor Lavrenko (2004). *Language-specific Models in Multilingual Topic Tracking*. Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 402-409.

Leek Tim, Hubert Jin, Sreenivasa Sista & Richard Schwartz (1999). The BBN Crosslingual Topic Detection and Tracking System. In 1999 TDT Evaluation System Summary Papers.

Pouliquen Bruno, Ralf Steinberger, Camelia Ignat, Emilia Käsper & Irina Temnikova (2004). *Multilingual and cross-lingual news topic tracking*. In: Proceedings of the 20th International Conference on Computational Linguistics, Vol. II, pp. 959-965.

Pouliquen Bruno, Marco Kimler, Ralf Steinberger, Camelia Ignat, Tamara Oellinger, Ken Blackler, Flavio Fuart, Wajdi Zaghouani, Anna Widiger, Ann-Charlotte Forslund & Clive Best (2006). Geocoding multilingual texts: Recognition, Disambiguation and Visualisation. Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006), pp. 53-58.

Schultz J. Michael & Mark Liberman (1999). Topic detection and Tracking using idf-weighted Cosine Coefficient. DARPA Broadcast News Workshop Proceedings.

Steinberger Ralf, Bruno Pouliquen & Camelia Ignat (2004). Providing cross-lingual information access with knowledge-poor methods. In: Andrej Brodnik, Matjaž Gams & Ian Munro (eds.): Informatica. An international Journal of Computing and Informatics. Vol. 28-4, pp. 415-423. Special Issue 'Information Society in 2004'.

Vinokourov Alexei, John Shawe-Taylor, Nello Cristianini (2002). *Inferring a semantic representation of text via cross-language correlation analysis*. Advances of Neural Information Processing Systems 15.

Wactlar Howard (1999). *New Directions in Video Information Extraction and Summarization*. Proceedings of the 10th DELOS Workshop.

Wayne Charles (2000). *Multilingual topic detection and tracking: Successful research enabled by corpora and evaluation*. Proceedings of 2nd International Conference on Language Resources and Evaluation.

Automatic Annotation of Bibliographical References with Target Language

Harald Hammarström
Dept. of Comp. Sci.
Chalmers University
S-412 96 Gothenburg
SWEDEN
harald2@chalmers.se

Abstract

In a large-scale project to list bibliographical references to all of the ca 7 000 languages of the world, the need arises to automatically annotated the bibliographical entries with ISO-639-3 language identifiers. The task can be seen as a special case of a more general Information Extraction problem: to classify short text snippets in various languages into a large number of classes. We will explore supervised and unsupervised approaches motivated by distributional characterists of the specific domain and availability of data sets. In all cases, we make use of a database with language names and identifiers. The suggested methods are rigorously evaluated on a fresh representative data set.

1 Introduction

There are about 7 000 languages in the world (Hammarström, 2008) and there is a quite accurate database of which they are (Gordon, 2005). Language description, i.e., producing a phonological description, grammatical description, wordlist, dictionary, text collection or the like, of these 7 000 languages has been on-going on a larger scale since about 200 years. This process is fully decentralized, and at present there is no database over which languages of the world have been described, which have not, and which have partial descriptions already produced (Hammarström, 2007b). We are conducting a large-scale project of listing all published descriptive work on the languages of the world, especially lesser-known languages. In this project, the following problem naturally arises:

Given: A database of the world's languages (consisting minimally of <unique-id, language-name>-pairs)

Input: A bibliographical reference to a work with descriptive language data of (at least one of) the language in the database

Desired output: The identification of which language(s) is described in the bibliographical reference

We would like to achieve this with as little human labour as possible. In particular, this means that thresholds that are to be set by humans are to be avoided. However, we will allow (and do make use of – see below) supervision in the form of databases of language references annotated with target language as long as they are *freely available*.

As an example, say that we are given a bibliographical reference to a descriptive work as follows:

> Dammann, Ernst 1957 *Studien zum Kwangali: Grammatik, Texte, Glossar*, Hamburg: Cram, de Gruyter & Co. [Abhandlungen aus dem Gebiet der Auslandskunde / Reihe B, Völkerkunde, Kulturgeschichte und Sprachen 35]

This reference happens to describe a Namibian-Angolan language called Kwangali [kwn]. The task is to automatically infer this, for an arbitrary bibliographical entry in an arbitrary language, using the database of the world's languages and/or databases of annotated entries, but without humanly tuned thresholds. (We will assume that the bibliographical comes segmented into fields, at least as to the title, though this does not matter much.)

Unfortunately, the problem is not simply that of a clean database lookup. As shall be seen, the distributional characteristics of the world language database and input data give rise to a special case of a more general Information Extraction (IE) problem. To be more precise, an abstract IE problem may be defined as follows:

- There is a set of natural language objects O
- There is a fixed set of categories C
- Each object in O belong to zero or more categories, i.e., there is a function $C : O \rightarrow Powerset(C)$
- The task is to find classification function f that mimics C.

The special case we are considering here is such that:

- Each object in O contains a small amount of text, on the order of 100 words
- The language of objects in O varies across objects, i.e., not all objects are written in the same language
- $|C|$ is large, i.e., there are many classes (about 7 000 in our case)
- $|C(o)|$ is small for most objects $o \in O$, i.e., most objects belong to very few categories (typically exactly one category)
- Most objects $o \in O$ contain a few tokens that near-uniquely identifies $C(o)$, i.e., there are some words that are very informative as to category, while the majority of tokens are very little informative. (This characteristic excludes the logical possibility that each token is fairly informative, and that the tokens *together*, on an equal footing, serve to pinpoint category.)

We will explore and compare ways to exploit these skewed distributional properties for more informed database lookups, applied and evaluated on the outlined reference-annotation problem.

2 Data and Specifics

The exact nature of the data at hand is felt to be quite important for design choices in our proposed algorithm, and is assumed to be unfamiliar to most readers, wherefore we go through it in some detail here.

2.1 World Language Database

The Ethnologue (Gordon, 2005) is a database that aims to catalogue all the known living languages of the world.[1] As far as language inventory goes, the database is near perfect and language/dialect divisions are generally accurate, though this issue is thornier (Hammarström, 2005).

Each language is given a unique three-letter identifier, a canonical name and a set of variant and/or dialect names.[2] The three-letter codes are draft ISO-639-3 standard. This database is freely downloadable[3]. For example, the entry for Kwangali [kwn] contains the following information:

> Canonical name: Kwangali
> ISO 639-3: kwn
> Alternative names[4]: {Kwangali, Shisambyu, Cuangar, Sambio, Kwangari, Kwangare, Sambyu, Sikwangali, Sambiu, Kwangali, Rukwangali}.

The database contains 7 299 languages (thus 7 299 unique id:s) and a total of 42 768 name tokens. Below are some important characteristics of these collections:

- Neither the canonical names nor the alternative names are guaranteed to be unique (to one language). There are 39 419 unique name strings (but 42 768 name tokens in the database!). Thus the average number of different languages (= unique id:s) a name denotes is 1.08, the median is 1 and the maximum is 14 (for Miao).

[1] It also contains some sign languages and some extinct attested languages, but it does not aim or claim to be complete for extinct and signed languages.

[2] Further information is also given, such as number of speakers and existence of a bible translation is also given, but is of no concern for the present purposes.

[3] From `http://www.sil.org/iso639-3/download.asp` accessed 20 Oct 2007.

[4] The database actually makes a difference between dialect names and other variant names. In this case Sikwangali, Rukwangali, Kwangari, Kwangare are altername names denoting Kwangali, while Sambyu is the name of a specific dialect and Shisambyu, Sambiu, Sambio are variants of Sambyu. We will not make use of the distinction between a dialect name and some other alternative name.

- The average number of names (including the canonical name) of a language is 5.86, the median is 4, and the maximum is 77 (for Armenian [hye]).

- It is not yet well-understood how complete database of alternative names is. In the preparation of the test set (see Section 2.4) an attempt to estimate this was made, yielding the following results. 100 randomly chosen bibliographical entries contained 104 language names in the title. 43 of these names (41.3%) existed in the database as written. 66 (63.5%) existed in the database allowing for variation in spelling (cf. Section 1). A more interesting test, which could not be carried out for practical reasons, would be to look at a language and gather *all* publications relating to that language, and collect the names occurring in titles of these. (To collect the full range of names denoting languages used in the bodies of such publications is probably not a well-defined task.) The Ethnologue itself does not systematically contain bibliographical references, so it is not possible to deduce from where/how the database of alternative names was constructed.

- A rough indication of the ratio between spelling variants versus alternative roots among alternative names is as follows. For each of the 7299 sets of alternative names, we conflate the names which have an edit distance[5] of $\leq i$ for $i = 0, \ldots, 4$. The mean, median and max number of names after conflating is shown below. What this means is that languages in the database have about 3 names on average and another 3 spelling variants on average.

i	Mean	Median	Max
0	5.86	4	77 'hye'
1	4.80	3	65 'hye'
2	4.07	3	56 'eng'
3	3.41	2	54 'eng'
4	2.70	2	47 'eng'

2.2 Bibliographical Data

Descriptive data on the languages of the world are found in books, PhD/MA theses, journal articles, conference articles, articles in collections and manuscripts. If only a small number of languages is covered in one publication, the title usually carries sufficient information for an experienced human to deduce which language(s) is covered. On the other hand, if a larger number of languages is targeted, the title usually only contains approximate information as to the covered languages, e.g., *Talen en dialecten van Nederlands Nieuw-Guinea* or *West African Language Data Sheets*. The (meta-)language [as opposed to target language] of descriptive works varies (cf. Section 2.4).

2.3 Free Annotated Databases

Training of a classifier ('language annotator') in a supervised framework, requires a set of annotated entries with a distribution similar to the set of entries to be annotated. We know of only two such databases which can be freely accessed[6]; WALS and the library catalogue of MPI/EVA in Leipzig.

WALS: The bibliography for the *World Atlas of Language Structures* book can now be accessed online (`http://www.wals.info/`). This database contains 5633 entries annotated to 2053 different languages.

MPI/EVA: The library catalogue for the library of the Max Planck Institute for Evolution Anthropology (`http://biblio.eva.mpg.de/`) is queryable online. In May 2006 it contained 7266 entries annotated to 2246 different languages.

Neither database is free from errors, imprecisions and inconsistencies (impressionistically 5% of the entries contain such errors). Nevertheless, for training and development, we used both databases put together. The two databases put together, duplicates removed, contains 8584 entries annotated to 2799 different languages.

2.4 Test Data

In a large-scale on-going project, we are trying to collect all references to descriptive work for lesser-known languages. This is done by tediously

[5] Penalty weights set to 1 for deletion, insertion and substitution alike.

[6] For example, the very wide coverage database worldcat (`http://www.worldcat.org/`) does not index individual articles and has insufficient language annotation; sometimes no annotation or useless categories such as 'other' or 'Papuan'. The SIL Bibliography (`http://www.ethnologue.com/bibliography.asp`) is well-annotated but contains only work produced by the SIL. (SIL has, however, worked on very many languages, but not all publications of the de-centralized SIL organization are listed in the so-called SIL Bibliography.)

going through handbooks, overviews and bibliographical for all parts of the world alike. In this bibliography, the (meta-)language of descriptive data is be English, German, French, Spanish, Portuguese, Russian, Dutch, Italian, Chinese, Indonesian, Thai, Turkish, Persian, Arabic, Urdu, Nepali, Hindi, Georgian, Japanese, Swedish, Norwegian, Danish, Finnish and Bulgarian (in decreasing order of incidence)[7]. Currently it contains 11788 entries. It is this database that needs to be annotated as to target language. The overlap with the joint WALS-MPI/EVA database is 3984 entries.[8] Thus $11788 - 3984 = 7804$ entries remain to be annotated. From these 7 804 entries, 100 were randomly selected and humanly annotated to form a test set. This test set was not used in the development at all, and was kept totally fresh for the final tests.

3 Experiments

We conducted experiments with three different methods, plus the enhancement of spelling variation on top of each one.

Naive Lookup: Each word in the title is looked up as a possible language name in the world language database and the output is the union of all answers to the look-ups.

Term Weight Lookup: Each word is given a weight according to the number of unique-id:s it is associated with in the training data. Based on these weights, the words of the title are split into two groups; informative and non-informative words. The output is the union of the look-up:s of the informative words in the world language database.

Term Weight Lookup with Group Disambiguation: As above, except that names of genealogical (sub-)groups and country names that occur in the title are used for narrowing down the result.

Following a subsection on terminology and definitions, these will be presented in increasing order of sophistication.

3.1 Terminology and Definitions

- C: The set of 7 299 unique three-letter language id:s

- N: The set of 39 419 language name strings in the Ethnologue (as above)

- $C(c)$: The set of names $\subseteq N$ associated with the code $c \in C$ in the Ethnologue database (as above)

- $LN(w) = \{id | w \in C(id), id \in C\}$: The set of id:s $\subseteq C$ that have w as one of its names

- $C_S(c) = \cup_{winC(c)} Spellings(w)$: The set of variant spellings of the set of names $\subseteq N$ associated with the code $c \in C$ in the Ethnologye database. For reference, the $Spelling(w)$-function is defined in detail in Table 1.

- $LN_S(w) = \{id | w \in C_S(id), id \in C\}$: The set of id:s $\subseteq C$ that have w as a possible spelling of one of its names

- WE: The set of entries in the joint WALS-MPI/EVA database (as above). Each entry e has a title e_t and a set e_c of language id:s $\subseteq C$

- $Words(e_t)$: The set of words, everything lowercased and interpunctation removed, in the title e_t

- $LWEN(w) = \{id | e \in WE, w \in e_t, id \in e_c\}$: The set of codes associated with the entries whose titles contain the word w

- $TD(w) = LN(w) \cup LWEN(w)$: The set of codes tied to the word w either as a language name or as a word that occurs in a title of an code-tagged entry (in fact, an Ethnologue entry can be seen as a special kind of bibliographical entry, with a title consisting of alternative names annotated with exactly one category)

- $TD_S = LN_S(w) \cup LWEN(w)$: The set of codes tied to the word w either as a (variant spelling of a) language name or as a word that occurs in a title of an code-tagged entry

[7] Those entries which are natively written with a different alphabet always also have a transliteration or translation (or both) into ascii characters.

[8] This overlap at first appears surprisingly low. Part of the discrepancy is due to the fact that many references in the WALS database are in fact to secondary sources, which are not intended to be covered at all in the on-going project of listing. Another reason for the discrepancy is due to a deprioritization of better-known languages as well as dictionaries (as opposed to grammars) in the on-going project. Eventually, all unique references will of course be merged.

- $WC(w) = |TD(w)|$: The number of different codes associated with the word w

- $WI(w) = |\{e_t|w \in Words(e_t), e_t \in WE\}|$: The number of different bibliographical entries for which the word w occurs in the title

- A: The set of entries in the test set (as above). Each entry e has a title e_t and a set e_c of language id:s $\subseteq C$

- $PA_A(X) = \frac{|\{e|X(e)==e_c, e \in A\}|}{|A|}$: The perfect accuracy of a classifier function X on test set A is the number of entries in A which are classified correctly (the sets of categories have to be fully equal)

- $SA_A(X) = \sum_{e \in A} \frac{|\{X(e) \cap e_c\}|}{|e_c \cup X(e)|}$: The sum accuracy of a classifier function X on a test set A is the sum of the (possibly imperfect) accuracy of the entries of A (individual entries match with score between 0 and 1)

3.2 Naive Union Lookup

As a baseline to beat, we define a naive lookup classifier. Given an entry e, we define naive union lookup (NUL) as:

$$NUL(e) = \cup_{w \in Words(e_t)} LN(w)$$

For example, consider the following entry e:

> Anne Gwenaïélle Fabre 2002 *Étude du Samba Leko, parler d'Allani (Cameroun du Nord, Famille Adamawa)*, PhD Thesis, Université de Paris III – Sorbonne Nouvelle

The steps in its NUL-classification is as follows are given in Table 2.

Finally, $NUL(e) = \{ndi, lse, smx, dux, lec, ccg\}$, but, simply enough, $e_c = \{ndi\}$.

The resulting accuracies are $PA_{NUL}(A) \approx 0.15$ and $SA_{NUL}(A) \approx 0.21$. NUL performs even worse with spelling variants enabled. Not surprisingly, NUL overclassifies a lot, i.e., it consistently guesses more languages than is the case. This is because guessing that a title word indicates a target language just because there is one language with such a name, is not a sound practice. In fact, common words like *du* [dux], *in* [irr], *the* [thx], *to* [toz], and *la* [wbm, lic, tdd] happen to be names of languages (!).

3.3 Term Weight Lookup

We learn from the Naive Union Lookup experiment that we cannot guess blindly which word(s) in the title indicate the target language. Something has to be done to individate the informativeness of each word. Domain knowledge tells us two relevant things. Firstly, a title of a publication in language description typically contains one or few words with very precise information on the target language(s), namely the name of the language(s), and in addition a number of words which recur throughout many titles, such as 'a', 'grammar', etc. Secondly, most of the language of the world are poorly described, there are only a few, if any, publications with original descriptive data. Inspired by the *tf-idf* measure in Information Retrieval (Baeza-Yates and Ribeiro-Neto, 1997), we claim that informativeness of a word w, given annotated training data, can be assessed as $WC(w)$, i.e., the number of distinct codes associated with w in the training data or Ethnologue database. The idea is that a uniqitous word like 'the' will be associated with many codes, while a fairly unique language name will be associated with only one or a few codes. For example, consider the following entry:

> W. M. Rule 1977 *A Comparative Study of the Foe, Huli and Pole Languages of Papua New Guinea*, University of Sydney, Australia [Oceania Linguistic Monographs 20]

Table 3 shows the title words and their associated number of codes associated (sorted in ascending order).

So far so good, we now have an informativeness value for each word, but at which point (above which value?) do the scores mean that word is a near-unique language name rather than a relatively ubiquitous non-informative word? Luckily, we are assuming that there are only those two kinds of words, and that at least one near-unique language will appear. This means that if we cluster the values into two clusters, the two categories are likely to emerge nicely. The simplest kind of clustering of scalar values into two clusters is to sort the values and put the border where the relative increase is the highest. Typically, in titles where there is exactly one near-unique language name, the border will almost always isolate that name. In the example above, where we actually have three near-

#	Substition Reg. Exp.	Replacement	Comment
1.	`\’\‘\^\~\"`	`’’`	diacritics truncated
2.	`[qk](?=[ei])`	`qu`	k-sound before soft vowel to qu
3.	`k(?=[aou]|$)|q(?=[ao])`	`c`	k-sound before hard vowel to c
4.	`oo|ou|oe`	`u`	oo, ou, oe to u
5.	`[hgo]?u(?=[aouei]|$)`	`w`	hu-sound before hard vowel to w
6.	`((?:[^aouei]*[aouei][^aouei]*)+?)(?:an$\|ana$|ano$\|o$)`	`\1a`	an? to a
7.	`eca$`	`ec`	eca to ec
8.	`tsch|tx|tj`	`ch`	tsch, tx to ch
9.	`dsch|dj`	`j`	dsch, dj to j
10.	`x(?=i)`	`sh`	x before i to sh
11.	`i(?=[aouei])`	`y`	i before a vowel to y
12.	`ern$\|i?sche?$`	`’’`	final sche, ern removed
13.	`([a-z])\1`	`\1`	remove doublets
14.	`[bdgv]`	`b/p,d/t,g/k,v/f`	devoice b, d, g, v
15.	`[oe]`	`o/u,e/i`	lower vowels

Table 1: Given a language name w, its normalized spelling variants are enumerate according to the following (ordered) list of substitution rules. The set of spelling variants $Spelling(w)$ should be understood as the strings $\{w/action_{1-i} | i \leq 15\}$, where $w/action_{1-i}$ is the string with substitutions 1 thru i carried out. This normalization scheme is based on extensive experience with language name searching by the present author.

$Words(e_t)$	$LN(Words(e_t))$	$Words(e_t)$	$LN(Words(e_t))$
etude	$\{\}$	cameroun	$\{\}$
du	$\{dux\}$	du	$\{dux\}$
samba	$\{ndi, ccg, smx\}$	nord	$\{\}$
leko	$\{ndi, lse, lec\}$	famille	$\{\}$
parler	$\{\}$	adamawa	$\{\}$
d’allani	$\{\}$		

Table 2: The calculation of NUL for an example entry

unique identifiers, this procedure correctly puts the border so that Foe, Pole and Huli are near-unique and the rest are non-informative.

Now, that we have a method to isolate the group of most informative words in a title e_t (denoted $SIG_{WC}(e_t)$), we can restrict lookup only to them. TWL is thus defined as follows:

$$TWL(e) = \cup_{w \in SIG_{WC}(e_t)} LN(w)$$

In the example above, $TWL(e_t)$ is $\{fli, kjy, foi, hui\}$ which is almost correct, containing only a spurious [fli] because Huli is also an alternative name for Fali in Cameroon, nowhere near Papua New Guinea. This is a complication that we will return to in the next section.

The resulting accuracies jump up to $PA_{TWL}(A) \approx 0.57$ and $SA_{TWL}(A) \approx 0.73$.

Given that we “know” which words in the title are the supposed near-unique language names, we can afford, i.e., not risk too much overgeneration, to allow for spelling variants. Define TWL_S (“with spelling variants”) as:

$$TWL_S(e) = \cup_{w \in SIG_{WC}(e_t)} LN_S(w)$$

We get slight improvements in accuracy $PA_{TWL_S}(A) \approx 0.61$ and $SA_{TWL_S}(A) \approx 0.74$.

The $WC(w)$-counts make use of the annotated entries in the training data. An intriguing modification is to estimate $WC(w)$ without this annotation. It turns out that $WC(w)$ can be sharply estimated with $WI(w)$, i.e., the raw number of entries in the training set in which w occurs in the

foe	pole	huli	papua	guinea	comparative	new	study	languages	and	a	the	of
1	2	3	57	106	110	145	176	418	1001	1101	1169	1482
1.0	2.0	1.5	19.0	1.86	1.04	1.32	1.21	2.38	2.39	1.10	1.06	1.27

Table 3: The values of $WC(w)$ for w taken from an example entry (mid row). The bottom row shows the *relative increase* of the sequence of values in the mid-row, i.e., each value divided by the previous value (with the first set to 1.0).

title. This identity breaks down to the extent that a word w occurs in many entries, all of them pointing to one and the same language id. From domain knowledge, we know that this is unlikely if w is a near-unique language name, because most languages do not have many descriptive works about them. The TWL-classifier is now unsupervised in the sense that it does not have to have annotated training entries, but it still needs raw entries which have a realistic distribution. (The test set, or the set of entries to be annotated, can of course itself serve as such a set.)

Modeling Term Weight Lookup with WI in place of WC, call it TWI, yields slight accuracy drops $PA_{TWI}(A) \approx 0.55$ and $SA_{TWI}(A) \approx 0.70$, and with spelling variants $PA_{TWI_S}(A) \approx 0.59$ and $SA_{TWI_S}(A) \approx 0.71$. Since, we do in fact have access to annotated data, we will use the supervised classifier in the future, but it is important to know that the unsupervised variant is nearly as strong.

4 Term Weight Lookup with Group Disambiguation

Again, from our domain knowledge, we know that a large number of entries contain a "group name", i.e., the name of a country, region of genealogical (sub-)group in addition to a near-unique language name. Since group names will naturally tend to be associated with many codes, they will sorted into the non-informative camp with the TWL-method, and thus ignored. This is unfortunate, because such group names can serve to disambiguate inherent small ambivalences among near-unique language names, as in the case of Huli above. Group names are not like language names. They are much fewer, they are typically longer (often multi-word), and they exhibit less spelling variation.

Fortunately, the Ethnologue database also contains information on language classification and the country (or countries) where each language is spoken. Therefore, it was a simple task to build a database of group names with genealogical groups and sub-groups as well as countries.

	PA	SA
NUL	0.15	0.21
TWL	0.57	0.73
TWL_S	0.61	0.74
TWI	0.55	0.70
TWI_S	0.59	0.71
TWG	0.59	0.74
TWG_S	0.64	0.77

Table 4: Summary of methods and corresponding accuracy scores.

All group names are unique[9] as group names (but some group names of small genetic groups are the same as that of a prominent language in that group). In total, this database contained 3 202 groups. This database is relatively complete for English names of (sub-)families and countries, but should be enlarged with the corresponding names in other languages.

We can add group-based disambiguation to TWL as follows. The non-significant words of a title is searched for matching group names. The set of languages denoted by a group name is denoted $L(g)$ with $L(g) = C$ if g is not a group name found in the database.

$$TWG(e) = (\cup_{w \in SIG_{WC}(e_t)} LN(w)) \cap_{g \in (Words(e_t) \setminus SIG_{WC}(e_t))} L(g)$$

We get slight improvements in accuracy $PA_{TWG}(A) \approx 0.59$ and $SA_{TWG}(A) \approx 0.74$. The corresponding accuracies with spelling variation enabled are $PA_{TWG}(A) \approx 0.64$ and $SA_{TWG}(A) \approx 0.77$.

5 Discussion

A summary of accuracy scores are given in Table 4.

All scores conform to expected intuitions and motivations. The key step beyond naive lookup

[9] In a few cases they were forced unique, e.g., when two families X, Y were listed as having subgroups called Eastern (or the like), the corresponding group names were forced to Eastern-X and Eastern-Y respectively.

is the usage of term weighting (and the fact the we were able to do this without a threshold or the like).

In the future, it appears fruitful to look more closely at automatic extraction of groups from annotated data. Initial experiments along this line were unsucessful, because data with evidence for groups is sparse. It also seems worthwhile to take multiword language names seriously (which is more implementational than conceptual work). Given that near-unique language names and group names can be reliably identified, it is easy to generate frames for typical titles of publications with language description data, in many languages. Such frames can be combed over large amounts of raw data to speed up the collection of further relevant references, in the typical manner of contemporary Information Extraction.

6 Related Work

As far as we are aware, the same problem or an isomorphic problem has not previously been discussed in the literature. It seems likely that isomorphic problems exist, perhaps in Information Extraction in the bioinformatics and/or medical domains, but so far we have not found such work.

The problem of language identification, i.e., identify the language of a (written) document given a set of candidate languages and training data for them, is a very different problem – requiring very different techniques (see Hammarström (2007a) for a survey and references).

We have made important use of ideas from Information Retrieval and Data Clustering.

7 Conclusion

We have presented (what is believed to be) the first algorithms for the specific problem of annotating language references with their target language(s). The methods used are tailored closely to the domain and our knowledge of it, but it is likely that there are isomorphic domains with the same problem(s). We have made a proper evaluation and the accuracy achieved is definetely useful.

8 Acknowledgements

We wish to thank the responsible entities for posting the Ethnologue, WALS, and the MPI/EVA library catalogue online. Without these resources, this study would have been impossible.

References

Baeza-Yates, Ricardo and Berthier Ribeiro-Neto. 1997. *Modern Information Retrieval*. Addison-Wesley.

Gordon, Jr., Raymond G., editor. 2005. *Ethnologue: Languages of the World*. SIL International, Dallas, 15 edition.

Hammarström, Harald. 2005. Review of the Ethnologue, 15th ed., Raymond G. Gordon, Jr. (ed.), SIL international, Dallas, 2005. *LINGUIST LIST*, 16(2637), September.

Hammarström, Harald. 2007a. A fine-grained model for language identification. In *Proceedings of iNEWS-07 Workshop at SIGIR 2007, 23-27 July 2007, Amsterdam*, pages 14–20. ACM.

Hammarström, Harald. 2007b. *Handbook of Descriptive Language Knowledge: A Full-Scale Reference Guide for Typologists*, volume 22 of *LINCOM Handbooks in Linguistics*. Lincom GmbH.

Hammarström, Harald. 2008. On the ethnologue and the number of languages in the world. Submitted Manuscript.

Author Index