A Platform for Multilingual News Summarization

David Kirk Evans and Judith L. Klavans
devans@cs.columbia.edu klavans@cs.columbia.edu
Computer Science Columbia University

April 20, 2003

Paper ID: Multilingual-Summarization-CU

Keywords: multilingual, summarization

Contact Author: devans@cs.columbia.edu

Under consideration for other conferences (specify)? No

Abstract

We have developed a multilingual version of Columbia Newsblaster as a testbed for multilingual multi-document summarization. The system collects, clusters, and summarizes news documents from sources all over the world daily. It crawls news sites in many different countries, written in different languages, extracts the news text from the HTML pages, uses a variety of methods to translate the documents for clustering and summarization, and produces an English summary for each cluster. The system is robust, running daily over real-world data. The multilingual version of Columbia Newsblaster provides a platform for testing different strategies for multilingual document clustering, and approaches for multilingual multi-document summarization.

A Platform for Multilingual News Summarization

Abstract

We have developed a multilingual version of Columbia Newsblaster as a testbed for multilingual multi-document summarization. The system collects, clusters, and summarizes news documents from sources all over the world daily. It crawls news sites in many different countries, written in different languages, extracts the news text from the HTML pages, uses a variety of methods to translate the documents for clustering and summarization, and produces an English summary for each cluster. The system is robust, running daily over real-world data. The multilingual version of Columbia Newsblaster provides a platform for testing different strategies for multilingual document clustering, and approaches for multilingual multi-document summarization.

1 Introduction

The Columbia Newsblaster[1] system has been online and providing summaries of topically clustered news daily since late 2001 (McKeown et al., 2002). The goal of the system is to aid daily news browsing by providing an automatic, user-friendly access to important news topics, along with summaries and links to the original article for further information. The system has six major phases: **crawling**, **article extraction**, **clustering**, **summarization**, **classification**, and **web page generation**.

The focus of this paper is to present the entire multilingual Columbia Newsblaster system as a platform for multilingual multi-document summarization experiments. The phases in the multilingual version of Columbia Newsblaster have been modified to take language and character encoding into account, and a new phase, **translation**, has been added. Figure 1 depicts the multilingual Columbia Newsblaster architecture. We will describe changes made to the system, and in particular will describe:

1. a method using machine learning to extract article text from web pages that is applicable to different languages, with an evaluation of the technique over English, Russian, and Japanese.

2. an approach using fast document glossing

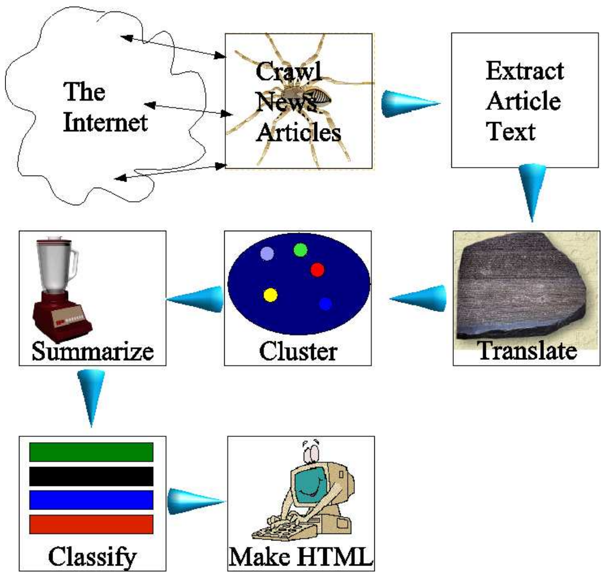

Figure 1: Architecture of the multilingual Columbia Newsblaster system.

[1]http://newsblaster.cs.columbia.edu/

techniques for multilingual document clustering with a monolingual document clustering system.

3. a baseline approach to multilingual multi-document summarization.

1.1 Related Research

There has been previous work in multilingual document summarization, such as the SUMMARIST system (Hovy and Lin, 1999) which extracts sentences from documents in a variety of languages, and translates the resulting summary. This system has been applied to Information Retrieval in the MuST System (Lin, 1999) which uses query translation to allow a user to search for documents in a variety of languages, summarize the documents using SUMMARIST, and translate the summary. The Keizei system (Ogden et al., 1999) uses query translation to allow users to search Japanese and Korean documents in English, and displays query-specific summaries focusing on passages containing query terms. Our work differs in the document clustering component – we cluster news to provide emergent topic structure from the data, instead of using an information retrieval model. This is useful in analysis, monitoring, and browsing settings, where a user does not have an a priori topic in mind. Our summarization strategy also differs from the approach taken by MuST in that we invest in a sophisticated summarization system, but only target a single language, shifting the majority of the multilingual knowledge burden to specialized machine translation systems. The Keizei system has the advantage of being able to generate query-specific summaries, which our system lacks. Future work does, however, include user-directed clustering and subsequent on-the-fly summarization.

Chen and Lin (Chen and Lin, 2000) describe a system that combines multiple monolingual news clustering components, a multilingual news clustering component, and a news summarization component. Their system clusters news in each language into topics, then the multilingual clustering component relates the clusters that are similar across languages. A summary is generated by linking "sentences" that are similar from the two languages. The system has been implemented for Chinese and English, and an evaluation over six topics is presented. Our clustering strategy differs here, as we translate documents before clustering, and cluster documents from all languages at the same time. This makes it easy to add support for additional languages by incorporating a new translation system for the language; no other changes need to be made. Our summarization model also provides summaries for documents from each language, allowing comparisons between them.

2 Extracting article text

To move Columbia Newsblaster into a multilingual capable environment, we must be able to extract the "article text" from web pages in multiple languages. The article text is the portion of a web page that contains the actual news content of the page, as opposed to site navigation links, ads, layout information, etc. For example, a recent web page from the New York Times consisted of a total of 70,671 bytes, but the actual article text of the web page was only 6,887 bytes. The remaining 60k was extraneous.

Our previous approach to extracting article text in Columbia Newsblaster used regular expressions that were hand-tailored to specific web sites. Adapting this approach to new web sites was difficult, since a human has to build regular expressions for each new site. Additionally, if the site layout changed, regular expressions would often need to be re-written. This approach is also difficult to adapt to foreign languages sites; in addition to requiring a human to write regular expressions, technical problems often arise when dealing with regular expressions in different character encodings, as discussed in Section 6.

We solved this problem by incorporating a new article extraction module that uses machine learning techniques to identify the article text. The new article extraction module parses HTML into blocks of text based on HTML markup and computes a set of features for each text block. 34 features are computed for each text block, based on simple surface characteristics of the text. For example, we use features such as the percentage of text that is punctuation, the number of HTML links in the block, the percentage of question marks, the number of characters in the text block, and so on. While the features are relatively language independent in that they can be computed for any language, the values they take on for a particular language, or web site, vary.

Training data for the system is generated using a GUI that allows a human to annotate text candidates with one of fives labels: "ArticleText", "Title", "Caption", "Image", or "Other". The "ArticleText" label is associated with the actual text of the article which we wish to extract. At the same time, we try to determine document titles, image caption text, and image blocks in the same framework. The "Title" tag is used to annotate the title of an article, while "Image" and "Caption" are used to indicate images and their captions. Columbia Newsblaster extracts and categorizes the images based on the caption text, and includes images in the summaries and an image browser. "Other" is a catch-all category for all other text blocks, such as links to related articles, navigation links, ads, and so on. The training data is used with the machine learning program Ripper (Cohen, 1996) to induce a hypothesis for categorizing text candidates according to the features. The article extractor module then uses the hypothesis to predict a category for each text block based on the features of the text block. This approach has been trained on web pages from sites in English, Russian, and Japanese as shown in Table 1, but has been used with sites in English, Russian, Japanese, Chinese, French, Spanish, German, Italian, Portuguese, and Korean.

The English training set was composed of 353 articles, collected from 19 web sites. Using 10-fold cross-validation, the induced hypothesis classify into the article text category with a precision of 89.1% and a recall of 90.7%. Performance over Russian data was similar, with a precision of 90.59% and recall of 95.06%.

Language	Training set	Precision	Recall
English	353	89.10%	90.70%
Russian	112	90.59%	95.06%
Russian	English Rules	37.66%	73.05%
Japanese	67	89.66%	100.00%
Japanese	English Rules	100.00%	20.00%

Table 1: Article extractor performance for detecting article text in three languages.

We also tried to use the English hypothesis to extract news from the Russian data set to test our hypothesis that rules tailored for each language would improve performance. As expected, the English hypothesis resulted in poor performance over the Russian data - precision was 37.66% and recall was 73.05%. We saw comparable results in Japanese where recall fell from 100.00% to 20.00%. Precision remained high though, as many fewer article text blocks were extracted. Adding new sites to this system is easy; a human annotates web pages using the GUI, and a new categorization hypothesis is learned from the new training data.

The article extraction system uses a back-off approach to dynamically choose the hypothesis to use for article extraction. Since using a hypothesis for a specific language or web site improved performance, we have implemented a back-off strategy to selecting the hypothesis to use. We attempt to use hypotheses in this order: website, encoding, default. The progression first checks for a hypothesis trained on the same website as the text. If one exists, it is used, otherwise, a hypothesis trained using the same encoding as the text is used. If no website or encoding hypothesis is found, the default hypothesis is used. The check for a hypothesis in the same encoding is used for convenience; it is slightly easier than checking based on the language. One of the more practical aspects of this back-off approach is that it is very easy to retrain the system for a specific website: simply create training data just for the website, train a hypothesis for it, and drop it in to the system without making any changes to the other hypotheses.

2.1 Title and date extraction

The article extraction component also determines a title for each document, and attempts to locate a publishing date for the articles. Title identification is important, since we present clusters of documents on the same topic to the user. These clusters are often very large, sometimes as many as 60 articles. In these situations, the only information the users sees are the titles for the articles - if our system chooses poor titles, they will have a difficult time discriminating between the articles. If a title is found using the Ripper hypothesis it is used. Unfortunately, titles for articles can only be found if the web site sets the titles apart in a manner that our HTML parser can recognize. Since this process is not always successful, we also have a variety of fall-back methods, including:

- Extracting the title from the HTML TITLE tag. This often is not successful, since many TITLE tags are uninformative - such as simply the web site name.
- Using heuristics to detect the title from the first text block. HTML markup (structural markup such as level headings, and presentation markup such as bold tags) are used to identify candidate titles.
- As a last resort, the first sentence, up to a certain character limit, is used with ellipsis inserted for long sentences.

The title identification code had to be slightly modified to support Unicode UTF8 character encoding.

The publication date of the article is important, since we would like to include only current news in our system. The DEMS summarization system also takes advantage of article publication dates for new information identification and sentence ordering. Earlier systems relied on the "last modified" field returned by the webserver, but our experience has shown that these headers do not accurately reflect the publication date of the article. For example, most large web sites use dynamic database-driven template engines to provide new ad content, new site navigation information, and so on with each web page. The modification headers for such web pages are often much more recent than the publication date of the news article on the web page.

To correctly extract dates for articles, we use heuristics to identify sequences of possible dates, weigh them, and choose the most likely date as the publication date. Identification of dates differs between languages. For example, in Japanese, dates have markers indicating the year, month, and day fields. Regular expressions for Japanese date extraction were added to the system, which already supported variations on both the American (MM/DD/YYYY) and European (DD/MM/YYYY) formats, along with heuristics to choose between the two in ambiguous cases. We plan to add support for other languages like Russian, Chinese, etc., in the near future.

3 Using simple document translation for multilingual clustering

The document clustering system that we use (Hatzivassiloglou et al., 2000) has been trained on, and extensively tested with English. While it can cluster documents in other languages, our goal is to generate clusters with documents from multiple languages, so a baseline approach is to translate all non-English documents into English, and then cluster the translated documents. We take this approach, and further use different translation methods for clustering and summarization.

Since many documents are clustered, we use simple and fast techniques for glossing the input articles when possible. We have developed simple dictionary lookup glossing systems for Japanese and Russian. While word sense disambiguation is important, our first implementations of glossing systems do not perform word sense disambiguation or other sophisticated disambiguation techniques. We are trying simple procedures as proof-of-concept that full translation does not need to be performed for clustering. Documents that are used in a cluster are later translated with a higher-quality method (currently, our interface to SYSTRAN's system via Altavista's babelfish.[2])

For languages where we do not have a simple translation mechanism available, we use the babelfish web interface to the SYSTRAN translation engine. The translated documents are then clustered as in the monolingual English version of Newsblaster. One potential difficulty is unknown words remaining in the translation - some unknown words or proper names are left in the document unaltered. This might bias documents from the same language to cluster together, as the untranslated terms add to the similarity measure. Further analysis should be performed to determine what sort of impact these untranslated terms have on the clustering algorithm.

4 A baseline approach to multilingual summarization

Our baseline approach to multilingual multi-document summarization is to apply our English-based summarization system, the Columbia Summarizer (McKeown et al., 2001), to document clusters containing machine-translated versions of non-English documents. The Columbia Summarizer routes to one of two multi-document summarization systems based on the similarity of the documents in the cluster (Hatzivassiloglou et al., 1999). If the documents are highly similar, the Multigen summarization system (McKeown et al., 1999) is used. Multigen clusters sentences based on similarity, and then parses and fuses information from similar sentences to form a summary. One area we are interested in investigating is whether we can use the parse information from grammatical English text with parse information from translated texts that might not be as well-formed. We have not focused on the Multigen summarization system yet, as the majority of clusters are routed to the other summarization system.

The second summarization system used is DEMS, the Dissimilarity Engine for Multi-document Summarization (Schiffman et al., 2002), which uses a sentence extraction approach to summarization. DEMS differs from traditional sentence extraction in its use of weighted concept sets to rank sentences as opposed to word frequencies, and an innovative measure of importance derived from the analysis of lead sentences from a large news corpus. When extracting sentences for inclusion in the summary, DEMS will not include sentences that are similar to sentences already in the summary, thus reducing repetition, and allowing maximal diversity in the selected material. The similarity decision is made based on the concept sets, so sentences do not have to use the same words to be judged as already covered in the summary. The resulting summary is then run through a named entity recovery tool (Nenkova and McKeown, 2003), which repairs named entity references in the summary by making the first reference descriptive, and shortening subsequent reference mentions in the summary.

The current multi-document multilingual summarization strategy is unchanged from monolingual English multi-document summarization; sentences in the input are scored based on the standard DEMS metrics, and sentences with a higher weight are extracted for the summary. The resulting summaries might contain sentences from translated documents, which are not grammatically correct. These summaries are highly dependent upon the quality of the translation systems used to translate the non-English documents. In recent work focusing on using translated text as input, the DEMS summarization system was modified to take this possible degradation of the input text into account. The modified version of DEMS prefers choosing a sentence from an English article if there are sentences that express similar content in multiple languages. We will soon apply this approach to multilingual Columbia Newsblaster, where a sentence receives a weight based upon the source language of the document. By setting lower weight penalties for languages with reliable translations, we can take the quality of the translation system for a given language pair into account.

[2] http://babelfish.altavista.com/

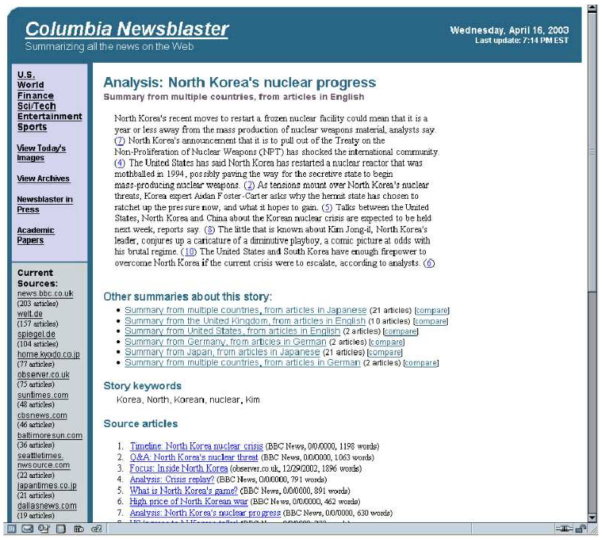

Figure 2: A screen shot of the summary page.

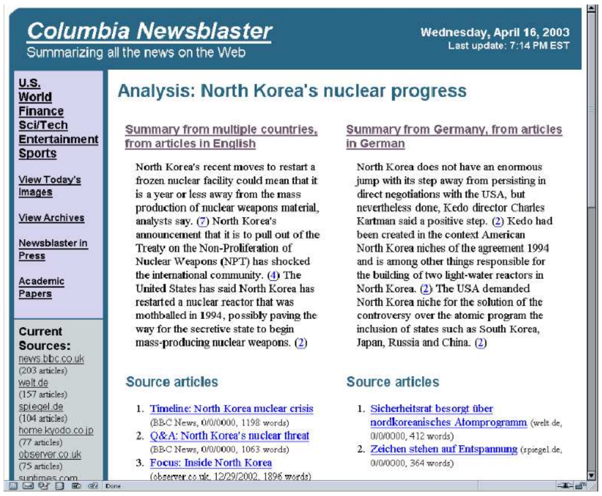

Figure 3: A screen shot comparing a summary from English documents to a summary from German documents.

4.1 Summary presentation

One of the goals of Columbia Newsblaster is to present large amounts of news in an easily processed form. Recent work has focused on presenting summaries of articles from different countries, and our multilingual works builds upon this. Summaries are generated for the entire cluster, as well as sub-sets of the articles based on the country of origin and language of the original articles. For example, the screen shot in figure 4.1 shows a summary of articles about talks between America, Japan, and Korea over nuclear arms. The summary covers articles in English (from America and the United Kingdom), Japanese, and German. The summary page first displays a summary which draws from all articles, and a list of summaries from articles in other languages and other countries allows the user to focus in on the countries or languages that interest them. We also allow the user to view two summaries side-by-side so they can easily compare differences between summaries from different countries, as shown in figure 4.1.

5 Evaluation

5.1 Multilingual Clustering Evaluation

We are performing an evaluation of the multilingual clustering component using glossing techniques as discussed in Section 3 over Russian text by manually examining clusters from a small test data set. The data set is a crawl over news from two Russian news sites (http://www.izvestia.ru/, http://www.mn.ru/), and English news from CNN.com, for a total of 880 articles. After translating the Russian documents with our glossing system and clustering the English and translated Russian documents, 448 clusters are produced. Of those, 7 clusters contained documents in both English and Russian. A hand-examination of the clusters showed that they were all high quality clusters – i.e., the topics of the English documents were tightly related to the topics of the Russian translated documents. We also compared to clustering runs using documents with slightly different translation processes (various methods of trying to emphasize proper nouns in the translated Russian and original English text) but these variations on the translation did not perform as well as the original glossing scheme. We are currently extending the evaluation to compare performance to using machine translation output from babelfish. We have not yet approached the task of looking at recall of the clustering, since even with this small data set, it would not be practical to examine the entire set by hand. The small number of multilingual clusters does not sound unreasonable, since even with English-only runs of Columbia Newsblaster, only a small number of clusters result from a large data set (from out of 2,000 - 3,000 input documents, generally only 300 clusters fit post-filtering requirements.)

5.2 Multilingual Summary Evaluation

Evaluation of multi-document summarization is a difficult task; the Document Understanding Conference (DUC)[3] is designed as an evaluation for multi-document summarization systems, but there have been problems defining quantitative metrics that show high agreement between humans. The ideal summary for a set of documents differs greatly depending on the intended application for the user. Columbia Newsblaster attempts address some of this need by providing different summaries based on the language and country of origin of the articles, but we have yet to evaluate our baseline approach to multilingual summarization. One method of summary evaluation that we have used in the past is to elicit qualitative ratings for a summary from users. We are currently integrating a feedback system into Newsblaster for use by users that use the system on a daily basis, and who have concrete needs focusing on an information analysis task. We plan to allow the users to score summaries based on the a variety of criteria, including: readability, usefulness for the task, and appropriateness of article titles. While these are qualitative judgments, they are useful since the feedback is from use in an actual information exploration task, as opposed to casual news browsing. The comments we have received have already helped to guide the future directions for the system.

[3] http://duc.nist.gov/

5.3 Experiments

As part of our multilingual summarization work, we are investigating approaches to summarization that use sentence-level similarity computation across languages to first cluster sentences by similarity, and then generate a summary sentence using information fusion techniques across languages as in the current MultiGen summarization system. The multilingual version of Columbia Newsblaster provides us with a platform to frame future experiments for this summarization technique. We plan to perform experiments using clusters with multilingual data that test the precision and recall of various approaches to first determining the multilingual sentence clusters. The similarity computation will be tested with translation at a variety of levels, first a system using full machine translation over the sentences, and English similarity detection. Second is using a set of simple part-of-speech based features for multilingual similarity detection in SimFinder MultiLingual (SimFinderML), a multilingual version of SimFinder (Hatzivassiloglou et al., 2001). Third is an experiment evaluating the usefulness of noun phrase detection and noun phrase variant detection as a primitive for multilingual similarity detection, using tools such as Christian Jacquemin's FASTR (Jacquemin, 1994; Jacquemin, 1999).

6 Challenges for robust multilingual document processing

While designing and working with programs that deal with multilingual text, one encounters many technical challenges, which are not the focus of research, but must be overcome in a practical system. We related some of the difficulties that we encountered in this section.

6.1 Crawling web pages

Minor modifications were made to the web crawler to add support for detecting the declared character encoding for web pages it crawls. The encoding is generally declared either in a header sent by the web server, or more likely in our experience, in the HEAD section of an HTML document using a META tag. A particularly difficult problem occurs when a web page declares an encoding, but the content is not valid for that encoding. Some pages encoded in the ISO-8859-1 character set actually contain characters from Microsoft's Code Page 1252 character set, a super-set of ISO-8859-1 which adds approximately 29 illegal character codes. Our article extractor has code to recognize this case, and the even less frequent case of valid ISO-8859-1 documents that contain HTML character entity escapes for windows code page 1252 characters.

The web crawler also attempts to detect a language for the documents it crawls using some simple heuristics based on the country of the domain name, character encoding, and some exceptions we map by hand. An alternative approach we plan to integrate soon is to use trigram based language prediction, although we have found the simple approach of table lookup to prove quite effective for our task. We have recently run into situations where language prediction based on content would be beneficial, such as determine Welsh from English content on the BBC web site.

6.2 Title and date extraction: How to properly use unicode in regular expressions

As explained in Section 2.1, the system attempts to extract a title, and a publication date for each article. Currently, we use regular expressions to identify candidate titles and dates, and heuristics to choose between them. The date extraction portion of the system is written in perl, which has facilities for using unicode character codes in regular expressions, but only from version 5.6.0 onwards. This has forced us to standardize on the version of perl that we run, and since our system uses many clients on different machine, prevents us from using some older machines as clients. Similarly, when extracting titles, the Perl system must be explicitly told that the data being processed is unicode utf8 character data; if not, perl might not correctly parse character boundaries, and accidently cut a character mid-stream, resulting in invalid utf8 data.

6.3 Use of translated documents

We plan to use many translation methods for document translation; a single system does not support all of the languages that we would like to target, some systems might perform better than others for certain language pairs, and so on. Our current experience using the babelfish web interface to Systran has shown that sometimes, either document translation fails, or our parsing of the returned translated web page fails, or some other component along the way fails. In certain modes of failure, the returned document comes back with errors in the encoding, resulting in invalid UTF8 characters, which we discard. When examining these files, it is unclear what they represent; the input document in another encoding, or some partial translation in an undetermined encoding.

Even when translation occurs, there are problems using the text for summarization. Sometimes, words that are not in the translation system's dictionaries are left verbatim in the output, or proper names that contain accented characters remain. This results in words like "Libération", which have the accented character encoded in UTF8. Unfortunately, not all of the tools that are used in the summarization process can handle UTF8, so as a work-around, we encode non-ASCII characters in their HTML unicode character entity reference forms[4]. Since the character entity references are encoded using only ASCII character codes, most programs will simply ignore them, and preserve them unmodified. We look forward to a day when all text processing tools can transparently handle text encoded in Unicode.

7 Future Work

There are many ares which can and should be addressed; here we focus on a few areas which

[4]http://www.w3.org/TR/REC-html40/sgml/entities.html

we plan to improve in the short-term. Discussion on some longer-term efforts, such as new summarization strategies tailored for translated text will be discussed as well.

7.1 Date extraction

Future work includes efforts to automate the date recognition process, and poor title identification. We would like to leverage existing date identification programs to identify dates in foreign language, instead of writing our own regular expressions for this task, and in fact would prefer to move to a machine-learning based system. Current performance with regular expressions developed for English and Japanese seem to perform well enough for the languages that we are testing with that we have not yet made this a priority.

7.2 Un-informative title elimination

For title extraction, one seemingly easy problem to attack is the rejection of clearly bad titles. Since Columbia Newsblaster runs every day, examining many thousands of articles, we believe it would be easy to identify titles that are clearly non-descriptive. Examples of such titles would be "Stock Market News" or "The Iraq War" - a simple procedure recording the titles seen each day, sorted by frequency, could be used as a list to determine titles to reject. Any title that has been seen with high frequency could be rejected as not descriptive enough for an article to give a clear idea of what it is about in a cluster of similar articles. While the title "Stock Market News" might be a good title, in a cluster about the stock market with 10 articles we would much prefer a title that describes a unique characteristic of the article over a more general one.

7.3 Summarizer improvements

As discussed in section 4, the summarization system should be modified to at least take into account the quality of the translation system used on a per-language basis. There are many areas to explore with the use of translated documents as input; can grammatical repair methods be applied to the summary to improve readability when combing sentences from translated documents and non-translated documents; can the information fusion techniques used in Multigen be modified to work with error-full input like the kind we have seen with translated documents as input?

We also plan to add more customization to the system, and support for query-driven reclustering, and on-the-fly summarization. Using client-side javascript to selectively hide clusters based on their classification (World, US, Financial, Sci-tech, Sports, or Entertainment,) cluster membership, language, or country of origin. On the server side, we would like to allow users to enter a few query terms which are used to re-cluster the input articles, and allow them to choose clusters to summarize based on the titles of the articles.

In this paper we have described a multilingual version of Columbia Newsblaster, a system that runs daily offering users an accessible interface to online news browsing. The multilingual version of the system incorporates two varieties of machine translation, one for clustering, and one for translation of documents for summarization. Existing summarization methods have been applied to translated text, with plans for an evaluation of the current method, and incorporation of summarization techniques specific to translated documents. The system presents a platform for further multilingual summarization experiments and user-oriented studies.

References

Hsin-Hsi Chen and Chuan-Jie Lin. 2000. A multilingual news summarizer. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 159–165.

William W. Cohen. 1996. Learning trees and rules with set-valued features. In *AAAI/IAAI, Vol. 1*, pages 709–716.

Vasileios Hatzivassiloglou, Judith L. Klavans, and Eleazar Eskin. 1999. Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning. In *In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-1999)*, June.

Vasileois Hatzivassiloglou, Luis Gravano, and Ankineedu Maganti. 2000. An investigation of linguistic features and clustering algorithms for topical document clustering. In *Proceedings of the 23rd ACM SIGIR Conference on Research and Development in Information Retrieval*.

Vasileios Hatzivassiloglou, Judith L. Klavans, Melissa Holcombe, Regina Barzilay, Min-Yen Kan, and Kathy McKeown. 2001. Simfinder: A flexible clustering tool for summarization. In *Proceedings of the North American Association for Computational Linguistics Automatic Summarization Workshop*.

E.H. Hovy and Chin-Yew Lin. 1999. Automated text summarization in summarist. In I. Mani and M. Maybury, editors, *Advances in Automated Text Summarization*, chapter 8. MIT Press.

Christian Jacquemin. 1994. Fastr: a unification-based front-end to automatic indexing. In *In Proceedings, Intelligent Multimedia Information Retrieval Systems and Management (RIAO'94)*, pages p. 34–47.

Christian Jacquemin. 1999. Syntagmatic and paradigmatic representations of term variation. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL'99)*, pages 341–348.

Chin-Yew Lin. 1999. Machine translation for information access across the language barrier: the must system. In *Machine Translation Summit VII*, September.

Kathleen McKeown, Judith Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. 1999. Towards multidocument summarization by reformulation: Progress and prospects. In *AAAI*, pages 453–460.

Kathleen R. McKeown, Regina Barzilay, David Kirk Evans, Vasileios Hatzivassiloglou, Min-Yen Kan, Barry Schiffman, and Simone Teufel. 2001. Columbia multi-document summarization: Approach and evaluation. In *Proceedings of the Document Understanding Conference*.

Kathleen R. McKeown, Regina Barzilay, David Kirk Evans, Vasileios Hatzivassiloglou, Judith L. Klavans, Ani Nenkova, Carl Sable, Barry Schiffman, and Sergey Sigelman. 2002. Tracking and summarizing news on a daily basis with columbia's newsblaster. In *Proceedings of the Human Language Technology Conference*.

Ani Nenkova and Kathy McKeown. 2003. Refernances to named entities: A corpus study. In *Proceedings of the Human Language Technology Conference*. To appear.

William Ogden, James Cowie, Mark Davis, Eugene Ludovik, Hugo Molina-Salgado, and Hyopil Shin. 1999. Getting information from documents you cannot read: An interactive cross-language text retrieval and summarization system. In *SIGIR/DL Workshop on Multilingual Information Discovery and Access (MIDAS)*, August.

Barry Schiffman, Ani Nenkova, and Kathleen McKeown. 2002. Experiments in multidocument summarization. In *Proceedings of the Human Language Technology Conference*, March.