EXTRACTING DOCUMENT FEATURES TO IMPROVE CLASSIFICATION AND CLUSTERING

Péter Schönhofen

Summary of Ph.D. dissertation

Supervisors: Hassan Charaf, Ph.D.
András Benczúr, Ph.D.

Budapest
2008
1 Introduction

During its 15 year history, the World Wide Web has become the largest information repository in the world thanks to its rapid growth and development. Currently it contains more than 40 billion pages (which number is of course continuously growing) in almost every language and about every topic: from medical science to hang gliding, from animal husbandry to course materials for university education. This huge pile of knowledge represents an enormous value, the work and effort of many million people; however, in order to fully take advantage of it, we have to build a sophisticated search engine which can cope with the particular characteristics of the medium. The pages in the World Wide Web are created and maintained by people of various social and cultural backgrounds who are interested in a wide range of topics, therefore an ideal search engine must be prepared to deal with unstructured pages whose format and content do not (or only partially) conform to standards, whose quality ranges widely, and may include abbreviations and technical jargon.

We can conduct searches in the World Wide Web – or generally speaking in any large collection of electronic documents – using three different approaches: document retrieval, question answering, and classification or categorization of documents.

In document retrieval, the user specifies his/her topic of interest using some method (keywords, selecting nodes from a topic hierarchy, through a visual interface etc), and the system searches and displays the relevant documents ranked according to how strongly they relate to the given area. The search relies on document extracts prepared by the system in advance, it might include shallow or deep natural language analysis, and it may even take into account the way the user interacted with the system during former sessions, discarding documents falling outside the user's general interests.

In question answering, our primary goal is not to retrieve documents dealing with the given question, but rather to produce a succinct and straightforward answer to it based on the content of documents available to the system. The answer can be generated by a system which is able to interpret natural language texts, or by extracting the relevant part(s) of the appropriate document(s); of course it is possible that the required knowledge is scattered across multiple documents. A special form of question answering is information extraction, when a form containing a set of pre-determined fields has to be filled out for each document (e.g. extracting data about company mergers).

In document clustering, the system has to group documents based on their subject (e.g. when organizing search results), but without any preliminary knowledge about either the number of groups or the topic of the documents. In document classification, these topic groups are previously specified, usually by giving a set of training documents pertaining to the given topics; the goal here is to determine which group a new document most closely relates to, or if no such group exists, to suggest the creation of a new group. A special type of classification is when there are only two groups: the documents the user is interested and not interested in (information filtering).

2 Previous work

There are several document search engines currently in operation (Google, Yahoo, Clusty etc), but despite their continuous improvement, their accuracy and simplicity of use still do not satisfies the needs of sophisticated users – it is not rare even today that a user would have to scan through hundreds or thousands of hit list items to find a relevant document to certain types of queries. Several attempts were made to increase efficiency: using natural language processing, focusing on concepts instead of mere...
word sequences, automatic correction or extension of the initial query. Because the query submitted by the user is compared to extracts of documents, not their full texts, a logical approach is to try to increase the quality of document extracts.

There are several methods to reduce document size [feature reduction]: omitting less important words; transforming documents to a set of concepts, then performing search in the concept space [latent semantic indexing]; or producing an actual, human-readable summary from the document. In the last case, the task is not only to decrease the document size – the extract has to be grammatically valid, fluid, meaningful in itself, and has to include every concept which is able to differentiate it from other document, although not necessarily plays an important role in the document subject. For example, if a page refers to the language PL/I only cursorily, but this concept is not present in any other page of the collection, it has to be part of the document extract.

In order to estimate the importance of words occurring in a document, we can also rely on a wide range of methods, which take into account the frequency of the word in the document \([tf, \text{term frequency}]\) and in the whole document collection \([idf, \text{inverse document frequency}]\); its position in the formal (title, headline, emphasis) or semantic structure of the document; its part of speech or grammatical role; or in case of hyperlinked documents, the number of times the given word is used as an anchor. Of course we might also estimate the importance of the documents themselves, either based on their popularity (e.g. via the number of clicks they received on the previously generated hit lists) or their position in the link graph – which not only influences the size of the extract but also makes it possible to represent a large set of semantically closely related documents with only a few samples on the search engine hit list.

3 Goals

Throughout my research I tried to answer two questions: (1) how can we estimate the importance of words, phrases or sentences present in a document better than existing methods; and (2) if we discard text fragments deemed as unimportant, how accuracy of search, clustering and classification of these documents will increase. Omitting the less important document parts might have beneficial effects on various document analysis techniques for two reasons. On the one hand, analysis can become more precise since we remove „threads” which are not closely related to the main document subject and therefore are usually misleading. On the other hand, reducing the amount of information to be processed in some case may significantly speed up analysis.

I regarded it very important that the methods and algorithms developed by me operate efficiently not only on HTML (or any other type of hyperlinked) pages, but rather on arbitrary texts; be as language independent as possible; and finally use all available information (e.g. formatting, document organization etc), but in such a manner as to not require intensive computations which would render them practically unfeasible.

4 Analysis methods

For the experiments with and evaluations of my methods I needed a document collection containing a large number of documents which covered a wide range of subjects, were ideally already categorized and of sufficient length, otherwise the results would have not reflected reliably the efficiency of the algorithms and proved their usefulness in real-word situations. Fortunately there were several corpora fulfilling these criteria: Reuters-21578 (21,578 categorized documents), Reuters Corpus Volume 1 (800,000, categorized), Glasgow Herald (56,000, not categorized), Los Angeles Times (110,000 not categorized documents from 1994, and 82,000 not categorized from 2002), Wall
Street Journal (173,000, not categorized), AP (243,000, not categorized) – all of them comprising of English-language news articles. In addition, WIPO (80,000 categorized documents) contained international patents, again written in English. The two Reuters corpora assigned documents to approximately 100 categories, some of them to only one, others to more than one, but in the latter case, I retained only the assignment to the most specific category (that containing the fewest documents). WIPO categorized documents in a hierarchical manner, four levels deep.

Steps of the experiments always followed the same general pattern. First, the document collections originally available in XML or SGML format were stored either in a relational database (IBM DB2 7.1, Oracle 8i, Oracle 9i) or in a special file suitable for further processing stages. Paragraph and sentence boundaries were recognized (by my own algorithm, by Zhang's Sentence Segmenter, or with the help of a utility program supplied as part of TreeTagger). Words were stemmed (either by a utility provided for WordNet, or by TreeTagger again), and stopwords were discarded (I always used a slightly modified version of the stopword list originally developed for the Smart search engine). Next, document extracts were produced, for which in certain cases I employed a backpropagated neural network, and finally these extracts were subjected to clustering and classification, representing the original documents.

For classification I used the naive Bayes component of the Bow toolkit; performance was measured here as the percentage of documents from the test set which were assigned to the correct category. For clustering purposes, I employed CLUTO, a program specifically designed for the analysis of very large number of documents, with default settings (repeated bisection, cosine-based similarity metric between documents, I2 criterion function); quality was now measured through entropy and purity, whose definition can be found at the discussion of Thesis 1. Finally, when I examined how well an algorithm managed to locate important words, I used precision and recall, the widely known measurements in the information retrieval community. Namely precision tells how much percentage of words or documents deemed as important by the system are actually important, while recall specifies how much percentage of important words or documents are identified by the system as such.

5 New scientific results

I grouped the novel results obtained during my research around six theses, whose descriptions will follow (mostly in chronological order of the corresponding papers).

**Thesis 1: Document extraction based on simple statistical measurements**

I have proved that if we select the words to include in the document extract not only based on the traditional tf and idf measurements, but also on a few additional features which take into account the word-document relation more deeply, then precision and recall of the selection of the most important words through a neural network, and quality of document clustering can be improved. The detailed discussion of the thesis can be found in Chapters 2 and 3 of the dissertation, and publications [1], [2], [11].

When we try to identify the role a given word plays in a document, we should take into account three factors: the position of the word inside the given document, the position of the word among the words of the whole corpus, and the position of the word inside the collection. In fact, three different models can be discerned, which attempt to capture the connection between words and documents through various relationships:

- How the given word or phrase relates to the central topic discussed in the document: it is more general than the topic, it entirely or partly describes the topic, it is more specific than the topic, or it is mentioned only in a cursory fashion.
What role the given word plays in the document collection, especially when the document subjects do not cover a wide range (like World Wide Web pages), but are restricted to a specific domain (e.g. corporate e-mail archives): the word is a stopword, is characteristic of the domain, it pertains/not pertains to the domain.

What is the approximate type of the document: overview, introduction, putting a topic in context, discussing some topic in detail, enumerating subtopic similarly to a catalog, comparing different topic and so on.

The statistical features measuring the behavior of words and word pairs can be split to two groups: local features regard a single document, global ones consider the whole collection (of course the distance between the members of a word pair measured in words or sentences should be limited). At both groups we can take into account the occurrence count, role in the grammatic structure, position inside the sentence, and if an external ontology is available, the location in it or the relationship with it.

After these preliminary ruminations the possible statistical features can be very easily enumerated. The most specific information is the frequency of a word pair in a given document, that is, we assign a count to a point in the document-word-word space (see Figure 1). When a dimension is removed, we get various subspaces, to which again a number can be assigned. A dimension can be removed in two ways: (1) we ignore the dimension; or just the opposite, (2) we measure the given statistical feature over all points of the dimension in question. For example in case of the document-word subspace we may mean the frequency of a word in a given document, or the frequency of word pairs, formed from a given word and any other imaginable word, inside some document. In the latter case, in order to produce a single value from the multitude of individual measurements, we have to employ some statistical operation (like average, standard deviation, maximum, minimum etc).

Statistical features describing grammatical roles and position inside a sentence can be calculated in the same manner; to form features focusing on the semantic relationship between words (where we can take into account the distance between their corresponding nodes inside the ontology) both word dimensions are required.

Traditional methods usually determine the importance of words present in a document with the help of the tf and idf measurements, where the former prefers word occurring more often, while the latter favors words which are rare in the whole document collection, and this way are better able to emphasize the differences between documents (filtering out the stopwords which are frequent but do not carry significant meaning). The set of features can be extended by four new items:

- frequency of word in the doc. ($LF_w$)
- frequency of word in corpus ($GF_w$)
- frequency of word pair in doc. ($LCF_{wv}$)
- frequency of word pair in corpus ($GCF_{wv}$)
that is, the horizontal dimension is the scope (document or corpus), and the vertical is the width (word or word pair). Two words were considered a word pair only if they were not located farther from each other in the text than 5 words. In addition to the above mentioned features, I introduced \( MGF, MGCF \) as the modified forms of \( GF \) and \( GCF \), respectively, which specify the number of documents containing a given word or word pair, instead of their total occurrence count; thus \( MGF \) corresponds to the traditional \( df \) [document frequency] measurement. Of course all values were normalized: \( LF \) by the appropriate \( GF \), \( GF \) by the number of words in the entire collection, \( MGF \) by the total number of documents, \( LCF \) by the geometric mean of the \( LF \) values for the individual words, finally \( GCF \) and \( MGCF \) by employing \( GF \) and \( MGF \).

Since I wanted to characterize words by statistical measurements, features involving word pairs had to be consolidated, which I did in two different ways: (1) for word \( w \) I took into account only the respective \( LCF \) and \( GCF \) values, computing their average, maximum and standard deviation; (2) I also took into account the related word \( v \) as:

\[
r_w = f_v(m_{wv}, m_{vw}, s_{vw}) \quad ; \quad s_1 = 1, \quad s_2 = LCF_{wv}, \quad s_3 = LCF_{vw} \times (1 - GCF_{wv}),
\]

where \( f \) is a function computing maximum, average or standard deviation; \( m \) is the feature in question (\( LCF \) or \( GCF \)); and \( s \) is a weighing factor, which again can assume three forms: \( s_1, s_2, s_3 \). As can be seen, variant 1 does not take into account the relations between the words; variant 2 prefers locally frequent pairs; finally, variant 3 rather emphasizes locally frequent, but at the same time globally rare word pairs.

By introducing a few additional, rather simple features (like document length, number of distinct words forming pairs with a specific word, difference between the global and local distributions of words forming pairs with a given word), the number of available features increased to 109, fed as inputs to a backpropagated neural network with two hidden layers. Training was performed out using 841 randomly selected articles from Reuters-21578, the test set comprised of 770 documents. I carried out two experiments: in the first I examined how well the neural network was able to pinpoint the sentence summarizing the document subject [lead sentence], which, due to the nature of the corpus, was always the first; while in the second that how well it could select words based on which efficient document clustering by CLUTO was possible.

Figure 2 shows the results of the first experiment; the multiple measurement points represent various ratios of positive/negative samples in the training set. It should be noted that out of the 109 input fed to the neural network, 38 (approximately 30%) can be omitted without significantly decreasing its performance.
In the second experiment, training of the neural network was not based on whether a given word was present in the first sentence or not, but on how well the word is able to emphasize the document category according to the following formula:

\[ r_w(d, t) = P_w(d, t) - N_w(d, t), \]

where \( P_w \) indicates the frequency of word \( w \) in documents assigned to the category of document \( d \), and similarly \( N_w \) specifies how often word \( w \) appears in documents assigned to other categories (the documents actually used from the corpus were assigned to a total of 15 categories). Thus after the neural network predicted \( r_w \) values for words present in a document of the test set, the document was represented during clustering by the \( m \) words with the highest such value – results are shown on Figure 3. The multiple measurement points correspond to various \( m \) values (which ranged from 3 to 25).

Figure 3. Quality of clustering; the continuous line represents the word selection method relying on the traditional tf and tf×idf measurements.

Quality is evaluated via two measurements: entropy and purity. Entropy indicates that documents originally pertaining to the same category to how much degree were assigned to the same group by the clustering algorithm; purity specifies to how much degree groups contain documents from the same category. Overall entropy and purity is computed by averaging entropies and purities of individual groups. It goes without saying that for entropy lower, for purity higher values correspond to better quality.

**Thesis 2: Document extraction based on the estimated relevance of sentences**

I have proved that if sentences occurring in a document are characterized by various measurements which capture the importance of words present in the sentence and the similarity to other sentences, then a backpropagated neural network is trained to select the most important sentence, then much more selection accuracy can be achieved in the test set than if we relied only on the tf×idf measurement. The detailed discussion of the thesis can be found in Chapter 4 of the dissertation and publication [5].

If one wants to produce document extracts, an obvious approach is to select sentences deemed as the most important, since on the one hand this way the extract will be readable not only by machines but also by humans, and on the other hand it is highly probable that the words or phrases characteristic of the discussed subject occur near each other, inside the document part intended as the summary by the original author. For sentence selection I again used the Reuters-21578 corpus (discarding too short articles from it), where I considered the first sentences to be the most important, similarly to
the previous thesis. The backpropagated neural network contained only one hidden layer, and I reserved 25% of the training set to prevent overlearning.

I assigned features characterizing sentences to three groups. The first group contained traditional measurements, namely the maximum, average and standard deviation of \( tf, idf \) and \( tf\times idf \) values computed for words constituting the given sentence. Moreover, it included the size of the sentence itself, naturally measured in words (\( s \)). This feature set represented the baseline whose performance I tried to surpass.

The second group comprised of features describing the semantic similarity between the given sentence and other sentences: size of the set of their common words (\( C_{xy} \)), difference between the positions occupied by these words, and a value calculated by the formula below, also characterizing similarity, but in a more sophisticated manner:

\[
m_{xy} = |C_{xy}|^{-1} \sum_{w \in C_{xy}} \log \frac{s}{f_w},
\]

where \( f_w \) specifies that how many other sentences of the current document word \( w \) is present also in. As in case of the first group, I again computed the average, maximum and standard deviation of the individual values; in addition, I carried out two kinds of normalizations, namely by the maximal such sentence score in the document and in the whole document corpus. An additional feature was that how many other sentences a given sentence has common words with (in the document or the entire collection).

Features in the third group were based on the so called coherent groups – a coherent group is a set of sentences where each two sentences have at least one word in common. The actual features were: size of the largest coherent group the sentence is member of, the maximal number of common words it shares with another sentence in this group, how far or near is the sentence located in the document from other member sentences. Consolidation of multiple values to yield a single word score (e.g. averaging), and normalization methods were the same as for the previous feature group.

![Figure 4. Performance of the neural network for various feature sets.](image)

Due to the large number of features I carried out principal component analysis, so the neural network was operated by only 50 inputs. The result (precision and recall of se-
lection) is shown in Figure 4; the multiple measurement points correspond to various ratio of positive/negative training samples. Line ,,T” represents traditional measurements; ,,A” indicates the effect of the inclusion of the second feature group; ,,B” and ,,C” the effect of the features related to coherent groups; finally, ,,D” corresponds to the application of few small changes, like doubling the number of positive/negative samples. Precision improved by 4-21%, recall by 10-36% (in absolute terms).

Of course it is not required to build the document extract word by word, it is also a perfectly valid approach that all words of the sentences deemed as the most important form the extract. In this case the previously mentioned features characterizing sentences can be used directly, their values do not have to be projected to words. However, precision and recall was computed again for words: how much percentage of words in the selected sentences are actually important (that is, member of the first document sentence), and how much percentage of important words occur in the selected sentences. Figure 5 shows the results: “T” and “S” represents word- and sentence-based selection with traditional features, “W” corresponds to the improved method.

![Figure 5](image_url)

Figure 5. Performance of word selection if the extraction unit is the sentence.

We can see that although switching from word-based to sentence-based selection, precision and recall increases significantly (by 5-7% and 13-15%, respectively) utilizing only the traditional measurements, the degree of improvement becomes even higher if we include all available features, both the traditional and the newly introduced.

**Thesis 3: Document extraction based on the frequency of word pairs**

I have proved that if we measure the frequency of word pairs formed from words present in the same sentence, select those occurring fewer or more often in the collection than would be expected, then documents are substituted by the set of words pertaining to the most of such pairs, then the quality of both classification and clustering improves, and the amount of information to be processed is reduced by 90%. See Chapter 5 of the dissertation and publication [6] for the detailed discussion of the thesis.
The words alluding most strongly to the subject discussed in a document are usually those which occur more often, or on the contrary, more infrequently than what would be expected from the textual context. For example, inside a collection of articles about computer science the mention of Bach is obviously remarkable, as well as the absence of word “symptom” in a document from a corpus of medical texts. Therefore it is logical to extract and utilize during later analysis as the document representative those words which are the member of the most word pairs of strange behavior.

The proposed method has four main steps. In the first step I calculated how many sentences a given word \( w \) is present in (\( f_w \)), and also counted that how many times word pairs formed from words in the same sentence in every imaginable combination appeared (\( f_{wv} \)); of course in both cases for all the documents in the corpus. However, I kept only those word pairs for which the following formula held true:

\[
\left| \frac{p_e - p_o}{2} \right| \geq 1.2, \quad p_e = \frac{f_w f_v}{N}, \quad p_o = \frac{f_{wv}}{N},
\]

where \( p_e \) is the expected co-occurrence probability of words \( w \) and \( v \), \( p_o \) is the actually observed co-occurrence probability, and finally \( N \) denotes the total number of sentences in the documents of the corpus. The value 1.2 was found to be a suitable threshold based on classification/clustering experiments in several corpora. I took the absolute value of the difference, as “negative” correlations are as important as “positive” ones.

In the second step I ranked words according to a score computed as the product of two numbers: one was the traditional \( \text{tf} \times \text{idf} \) measurement, while the other was the amount of salient word pairs which were formed with the given word and occurred inside the sentences of the current document. Because some documents were too short to yield enough salient word pairs to score every word, I introduced a secondary ranking containing the remaining words scored according to their \( \text{tf} \times \text{idf} \) values. The secondary ranking was appended to the primary with two empty ranks between them, to emphasize the lesser relevance of the former. For each document I kept only the words at the top \( P \) ranks as the most characteristic, \( P \) was chosen to be 10 based on experiments.

In the third step I computed the average ranking indexes of words over all documents in the collection, then I discarded those which was present in too few documents, or received too low ranks, further reducing extract sizes to \( R < P \) (\( R \) was set to 5, again...
based on several experiments). This way I was able to filter out the rare, statistically insignificant words, which would help neither classification nor clustering, and also those words which would be included in the extract only for lack of something better.

In the fourth step I collected the words \( v \) co-occurring with a given word \( w \) in extracts, and counted the number of these occurrences \( (f_{vw}) \). If a word appeared almost always in the company of some other word, and thus could be replaced by it (e.g. “Thomas” in “Thomas Edison”), I discarded it. The exact formula was the following:

\[
f_{vw} \geq 0.85 \max_z f_{wz}.
\]

Clustering results, more precisely the entropies and purities of groups established by CLUTO for various group numbers are shown in Figure 6. Compared to the analysis involving the full text \((A)\), 4- and 5-word extracts \((P_4, P_5)\) show a visible improvement, although document sizes was reduced by 90%, a compression rate higher than if we kept only titles. If we use only positive correlated pairs \((P_5^+)\), quality deteriorates.

**Thesis 4: Exploiting rare features during classification and clustering**

I have proved that when we classify or cluster documents, with the help of extremely rare words and \( n \)-grams (regular or skipping) the number of documents to be processed can be reduced by 5-25%, and in addition the quality of analysis can be improved by 0.5-1.6% (in absolute terms). The additional advantage of the proposed method is that it is very simple, does not require complex computations, and is language-independent. See Chapter 6 and publication [7] for detailed discussion of the thesis.

Extremely rare (those appearing in 2-5 documents at most in the entire corpus) words and \( n \)-grams are surprisingly reliable indicators of the fact that the documents which share them discuss the same subject. For example, experiments proved that in the case of 20 Newsgroups, 3-grams occurring twice in the whole collection predicted whether the two documents belonged to the same category by 10% more precisely (in absolute terms) than a naive Bayes classifier trained on the 30% of the corpus. The explanation: rare textual elements (typically pronouns, very specific technical terms, stylistic marks of a particular author) identify the document subject so close that they can be used effectively for almost any categorization aspect.

The underlying idea of the proposed method is that following the usual pre-processing (stemming, removing stopwords etc), I count the frequency of words and regular or skipping \( n \)-grams, and build a similarity graph based on this information. Namely the graph nodes correspond to documents, and an edge connects them if the documents share some rare items – edges are weighed according to the number and reliability of these shared rare items (the larger is \( n \), the more useful is the \( n \)-gram). Of course, the graph is exploited differently at classification and clustering; however, in both situations we modify the set of documents to be processed, and in the last step, we project the result obtained on the modified set to the original documents.

During classification, the location of documents \( X \) and \( Y \) connected in the graph with respect to the training and test set can be of three types. If \( X \) is in the training set and \( Y \) is in the test set, we immediately assign \( Y \) to the category determined for \( X \), and we either does not feed \( Y \) at all to the classification algorithm or add it to the training set. When both \( X \) and \( Y \) are part of the training set, we simple ignore the relationship between them. Finally, if both \( X \) and \( Y \) pertains to the test set, we unify their content and then pass this unified “virtual” document to the classification algorithm, so that it can hopefully detect its subject more accurately thanks to the enriched vocabulary. After the “virtual” document has been categorized, we assign its category to both \( X \) and \( Y \).

Figure 7 shows the results for the corpora 20 Newsgroups and RCV1.
Figure 7. Effect of rare features on the classification accuracy of the 20 Newsgroups and RCV1 corpora, for various training set sizes.

In case of clustering, the situation is simpler, as there are no separate training and test set: I merged the content of $X$ and $Y$, and passed the unified document to the clustering algorithm, and assigned both to the group selected for the unified document.

It is important to note that if a document is related to more than one other documents in the graph, we deal with only the document connected through the edge with the highest weight (for which the probability of a correct pairing is the highest). In case of classification, with the help of the following formula we can compute how much percentage of documents is expected to receive correct categorization:

$$A_r = 1 - [N_p r (1 - r) f_p + 2 N_p (1 - r)^2 (1 - f_o) f_p + 2 N_p (1 - r)^2 f_p + (N_d - 2 N_p) (1 - r) f_o]$$

where $r$ is the training set ratio, $N_d$ is the number of documents, $N_p$ is the number of document pairings actually used, $f_o$ is the error ratio of the classificator algorithm (that is, how much percentage of documents are classified correctly), and $f_p$ is the pairing error ratio (how much percentage of pairs have members in the same category).

**Thesis 5: Characterizing documents by Wikipedia categories**

I have proved that if a documents is connected to the Wikipedia articles whose titles (partly or entirely) appear in its text, and next we represent documents during classification or clustering by the most important words according to their $tf\times idf$ measurements and by the Wikipedia categories attached to the connected Wikipedia articles, then the quality of classification and clustering remains the same or even improves compared to the case when the full text of documents is used. The detailed discussion of the thesis can be found in Chapter 7 of the dissertation and papers [3], [8], [9].

Of course the extracts representing the document during classification or clustering do not have to be built from words or phrases directly lifted from the document text, it can be formed from concepts closely related to the document topic as well. Concepts are usually more general than words, so they can more efficiently convey the difference (similarity) of the document from (to) the other members of the collection. At the same time, however, they are harder to identify, as the text often refers to them only indirectly. Regrettably, currently there is no ontology having sufficiently wide coverage and depth to be able to characterize documents with arbitrary topics, but fortunately the Wikipedia category system with its 55,000 nodes can easily play this role.
Assignment of Wikipedia categories to documents was carried out in the following way. First of all, I pre-processed the Wikipedia corpus downloaded from the Internet: I removed articles which enumerated the various meanings of some phrase (e.g. CDC as an organization and a type of computer), or carried administrative information relevant only for authors (stylistic guidelines, list of articles to be expanded etc). From the remaining articles I kept only their titles and category assignments, I discarded their bodies; titles were stemmed with the help of TreeTagger, and both stopwords and supplementary descriptions between parentheses were deleted from them – naturally, stemming and stopword removal was carried out also on the documents to be processed. Next, I examined the documents one by one, collecting those Wikipedia articles whose titles appeared in the document texts (allowing a one word mismatch, because articles often mention pronouns, concepts in their abbreviated form, e.g. they refer to Thomas Edison simply as Edison). Finally, I gathered the Wikipedia categories these recognized articles were connected to, scoring the categories with the formula below:

$$R_c = \frac{v_c}{d_c} \times \sum_{a \in c} R_a,$$

where $d_c$ is the number of distinct words occurring in titles of all (not only recognized) articles assigned to category $c$, while $v_c$ is the number of distinct words in the titles of the recognized articles. The factor $R_a$, which estimate the importance of article $a$, is:

$$R_a = \sum_{w \in a} R_w \times \frac{1}{a_w} \times \frac{1}{n_a} \times \frac{S_t}{L_t},$$

where $R_w$ is the weight of words present in article title (see later), $a_w$ denotes the number of articles also carrying word $w$ in their titles, $n_a$ is the number articles with exactly the same title as the current article (due to stemming and the removal of stopwords originally different titles can easily become identical), finally, $S_t$ is the number of title words found in the document text, $L_t$ is the title length. $R_w$ is computed as:

$$R_w = tf_w \times \log \frac{N}{cf_w},$$

where $tf_w$ is the frequency of the word in the currently examined document, and $cf_w$ is the number of Wikipedia categories whose articles contain the word in their titles.

Figure 8. Accuracy of classification of the 20 Newsgroups and RCV1 corpora.

After words were ranked, I represented the document during classification and clustering by the top $n$ categories. As Figure 8 illustrates, although in case of the RCV1 cor-
pus categories alone were sufficient to surpass the performance observed when the full text of documents are used, real improvement can be experienced if the extract is augmented by the 20 words of the document having the highest $tf \times idf$ values.

**Thesis 6: Exploiting Wikipedia articles to translate query phrases**

I have proved that when translating Hungarian and German queries to Hungarian, the following method helps to select the most suitable translation of the available candidates. Let us connect the possible translations to Wikipedia articles with the same title, and rank these translations based on how strongly the corresponding Wikipedia articles relate to articles connected to other translations. If raw translation is carried out by freely available electronic dictionaries, using bigram statistics for prefiltering, Wikipedia-based selection improves the MAP of the translated queries by 7-12%. See Chapter 8 of the dissertation and publications [4], [10] for the details of the thesis.

Although the overwhelming majority of pages accessible on the Internet were written in English, several users do not or only superficially know this language, and therefore cannot construct efficient query phrases for search engines (in addition, users fluent in English might be interested in pages authored in other languages). We could easily translate queries from one language to another with the help of many freely available machine readable dictionaries, the real challenge is to find the translation most suitable to the given textual context from the multitude of possible translation, for which the concepts and their relationships stored inside Wikipedia can provide significant aid.

To prove the above statement, I utilized queries evaluated over the English language GH, LAT-94 and LAT-02 corpora, as part of the CLEF effort, and the search engine developed by SZTAKI. I submitted queries to the search engine in three versions: (1) official English form, (2) as a raw translation from a non-English language (German or Hungarian), and finally (3) as a translation improved through Wikipedia concepts. Quality of the hit list is shown in the table below, in case of 300 German queries – as can be seen, Wikipedia has managed to significantly increase each measurement.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Method</th>
<th>P@5</th>
<th>R@5</th>
<th>P@10</th>
<th>R@10</th>
<th>MRR</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>GH</td>
<td>English</td>
<td>34.12</td>
<td>27.78</td>
<td>27.00</td>
<td>36.66</td>
<td>0.5594</td>
<td>0.3537</td>
</tr>
<tr>
<td></td>
<td>raw</td>
<td>28.00</td>
<td>24.21</td>
<td>22.71</td>
<td>32.88</td>
<td>0.4940</td>
<td>0.3011</td>
</tr>
<tr>
<td></td>
<td>raw+Wikipedia</td>
<td>29.41</td>
<td>25.99</td>
<td>22.88</td>
<td>33.62</td>
<td>0.5078</td>
<td>0.3188</td>
</tr>
<tr>
<td>LAT-94</td>
<td>English</td>
<td>36.10</td>
<td>19.65</td>
<td>30.04</td>
<td>27.97</td>
<td>0.5746</td>
<td>0.2974</td>
</tr>
<tr>
<td></td>
<td>raw</td>
<td>27.56</td>
<td>15.36</td>
<td>22.80</td>
<td>22.29</td>
<td>0.4667</td>
<td>0.2055</td>
</tr>
<tr>
<td></td>
<td>raw+Wikipedia</td>
<td>29.43</td>
<td>18.34</td>
<td>24.39</td>
<td>24.98</td>
<td>0.4863</td>
<td>0.2327</td>
</tr>
</tbody>
</table>

During raw translation, I looked up the possible translations of the words and phrases present in the queries (which contained three parts: a succinct form with only a few words, a one- or two-sentence explanation, and a detailed description) in the appropriate machine readable dictionary. In case of multiple candidates, I selected that which, based on bigram statistics computed over the corpus, was most probable w.r.t. to the other candidate translations inside the same sentence. The score of translation $w$ was:

\[
S_w = \max (P_{vw, w1}, P_{vi, wn}) \quad P_{vi, wn} = \frac{f_{vi, wn}}{f_{vi}}
\]

where $v$ is the possible translation of an other phrase in the current sentence, of which $vi$ is the $i$th, and $vn$ is the last word; in addition, $f_{vw}$ denotes the probability of $v$ followed by $w$, and $f_i$ is the occurrence frequency of word $v$. Despite its simplicity, the method exhibited fairly good performance, but it still was not able to always select only one translation candidate, often several candidates received the same score.
As the first step of further filtering I connected the possible English translations and their compounds to the Wikipedia articles with exactly the same title (of course article titles were stemmed and stopwords were removed from them), scoring them as:

\[ S_c = L_c \times \frac{1}{1 + M_c} \times F_c \]

where \( F_c \) denotes the number of text locations which carry words referring to article \( c \); \( M_c \) is the number of articles competing with \( c \) (that is, whose titles share at least one word with the title of \( c \)); finally, \( L_c \) is the number of text locations carrying words which refer to articles related to \( c \) (they link to or are linked to by \( c \) inside Wikipedia). Next, for each position I kept only the articles with the highest scores, and then I ranked also the words themselves with a very simple formula, namely:

\[ R_w = \sum_q S_c \]

where \( q \) denotes text positions occupied by word \( w \) as part of a phrase referencing the Wikipedia article \( c \). I formed the query translation from the top three words, and extended their set by the words attached to phrases of the original query title; otherwise I would have lost the pronouns which are important during searching but do not always receive high scores due to their low frequency inside the target corpus.

6. Utilization of the new scientific results

The document extraction methods constructed by me can be useful in every situation where processing of the full document text would require a too large amount of computational resources, or where phrases only incidentally mentioned in the text might mislead the analysis or would reduce its accuracy. Some possible areas are:

- selecting words from the document text which should be indexed (and then compared to search criteria) as part of an information retrieval system;
- selecting document fragments or snippets to show in a result list produced in response to some search criteria entered into an information retrieval system;
- selecting words from documents which will be represent the documents during clustering (e.g. when organizing hit list elements produced by a search engine, merging e-mails based on their topic, collecting frequent phrases of a topic);
- selecting words from documents which will represent them during document filtering (e.g. filtering incoming e-mails according to user interest).

The most important advantage of the proposed methods is that they rely on very simple computations and tools, which are either already available for most languages or are relatively easy to develop. In addition, due to their generic nature they can be inserted in almost all text analysis systems at the appropriate processing phase.

7. Publications related to the dissertation

Papers published in Hungarian journals:


Papers published in foreign journals:


Papers presented at foreign conferences:


Citation:


Citations:


Citations:


Papers presented at Hungarian conferences:

8. Other publications
