PRACTICAL APPROACH TO AUTOMATIC TEXT SUMMARIZATION

JIRI HYNEK[1], KAREL JEZEK[2]

[1]inSITE, s.r.o.
Rubesova 29, Plzen, Czech Republic
e-mail: jiri.hynek@insite.cz
[2]Department of Computer Science and Engineering
University of West Bohemia
Univerzitni 22, 306 14 Plzen, Czech Republic
e-mail: jezek_ka@zcu.cz

The significance of automatic document summarization increases with the threat of information overload we are facing. Short summaries can be presented to users, for example, in place of full-length documents found by a search engine in response to a user's query. We have analyzed various approaches to document summarization, using some existing algorithms and combining these with a novel use of itemsets. The resulting summarizer is evaluated by comparing classification of original documents and that of abstracts generated automatically. Despite highly promising results achieved by this evaluation, readability of abstracts must be further improved by integrating additional heuristic approaches.

Keywords: document summarization, summarizer, condensation, abstract, abstracting, extraction, text, machine learning, classification, categorization, sentence selection, highlight, classifier, heuristics, itemsets, term frequency, evaluation

INTRODUCTION

Our recent research has been focused namely on the use of inductive machine learning methods for automatic document classification. We have developed a novel text categorization method called Itemsets Classifier, presented, for example, at ELPUB 2002 (1). By principle, this classifier is ideal for very short documents (30 to 50 significant terms). In order to use it for full-length documents, we need to find an efficient way of creating document summaries. It is essential to take an intelligent approach to document abstracting in order to preserve features that will be used later in the document classification phase (i.e. combinations of frequently occurring terms).

TYPES OF ABSTRACTS

Various types of summaries can be generated by document summarizers:

Indicative: Indicative summaries give brief information on the central topic of a document, preserving the critical portion of the content. These are useful in IR applications, being returned by search engines in place of full-length documents. Indicative summary is typically 5-10 % of the original text.

Informative (substantive): Informative summaries provide a substitute ("surrogate", "digest") for full document, retaining important details, while reducing information volume. Informative summary is typically 20-30 % of the original text.

Evaluative: Evaluative summary captures the point of view of the author on a given subject.

Topical: Based on a user-supplied statement of interest, a summary related to this topic is generated.

In the context of automatic document classification by Itemsets Classifier, we are interested primarily in *indicative* summarization.

Both fixed-length and best-length summaries can be generated. A fixed-length text summary will be preferred for the subsequent document categorization by Itemsets Classifier.

Automatic summarization usually consists in extracting sections (words, sentences, or paragraphs) of the original text. Therefore, resulting documents are often called "extracts".

METHODOLOGY

Besides other techniques, we are using co-occurrence of terms to summarize text documents. It is important to set up a limit for the distance at which any two significant words will be considered significantly related (a form of sliding-window application). The frequency of occurrence of various terms is essential. The importance of term frequency for document summarization was recognized by Luhn (2) as early as in 1958.

Luhn observes that the relative position of frequent terms within a sentence also furnishes useful measurements for determining significance of sentences. Significance of sentences can therefore be expressed as a combination of word frequency and position of these words. An intelligent summarizer should take into account linguistic implications, such as grammar, syntax, and possibly logical and semantic relationships. Speaking strictly of word frequency and word position, wherever the greatest number of frequently occurring different words are found in the greatest physical proximity to each other, the probability is very high that the information being conveyed is most representative of the document (2). It is important to set a limit for the distance at which any two significant words will be considered significantly related.

The document-cleaning phase (see below) of our document summarizer is based on quite a sophisticated stemming engine performing morphological normalization for Czech, English, German and other languages. The dictionary-based stemmer used for Czech currently includes over 3.3 million different word forms. Thanks to availability of the *ispell* open source dictionaries (see http://fmg-www.cs.ucla.edu/geoff/ispell-dictionaries.html), we can integrate other common languages into our system easily.

We are stressing the importance of language-independence of the actual summarization engine, as arbitrary text documents can arrive at its input. With the exception of the document cleaning phase, our document summarizer does not utilize language-dependent techniques.

Either whole sentences or whole paragraphs are selected for the resulting summary. Various heuristic approaches are utilized to improve final indicative abstracts. There are many improvements that can be made to the quality of the summaries, such as higher cohesion in sentence selection or sentence generation, and topic coverage across the set of topics mentioned within a document. We have shown that the quality of machine-generated summaries can be improved by utilizing heuristics and a combination of several summarization methods.

APPROACHES TO AUTOMATIC SUMMARIZATION

"Summary by extraction" has already been mentioned by Kupiec, Pedersen and Chen (3). The goal is to find a subset of the document that is indicative of its contents (sentences are scored and those with the best score are presented in a summary). A simple Bayesian classification function has been developed by Kupiec et al. in order to assign a score to each sentence, which can be used to select sentences for inclusion in a generated summary.

The following methods are most commonly used by document summarizers: sentence length cutoff (short sentences are excluded from summaries), use of cue phrases (inclusion of sentences containing phrases such as "in conclusion", "as a result", "in summary", "to sum

up", "the point is"), sentence position in a document / paragraph, occurrence of frequent words, relative position of frequent terms within a sentence, use of uppercase words, occurrence of title words (80 % of significant words occurring in the title correspond to the most frequent significant words in the document), use of author-supplied abstract (in case it is present), or intra-document links among passages of a document.

IMPLEMENTATION

GENERALIZED SUMMARIZATION ALGORITHM

The following phases of the summarization process can be identified:

1. Document cleaning

First, we convert documents into internal format (XML). We apply stemming, as occurrence of various word forms is not utilized by the summarization algorithm. Stop words are then removed from the text. Further processing of document collection is thus simplified.

2. Segmentation into passages

Documents are segmented into sentences and paragraphs (use of heuristic methods).

3. Indexing

Words are replaced by numeric values to speed up further processing.

4. Sentence scoring

Each sentence is assigned a numerical score depending on the summarization method being used. Sentences with the highest score are selected.

5. Synthesis (abstract generation)

Document abstract is composed of sentences with the highest score. In this phase we can remove unimportant parts of sentences. It is also possible to enhance abstract by using sentences bearing relatively low information value, but improving readability of the final abstract.

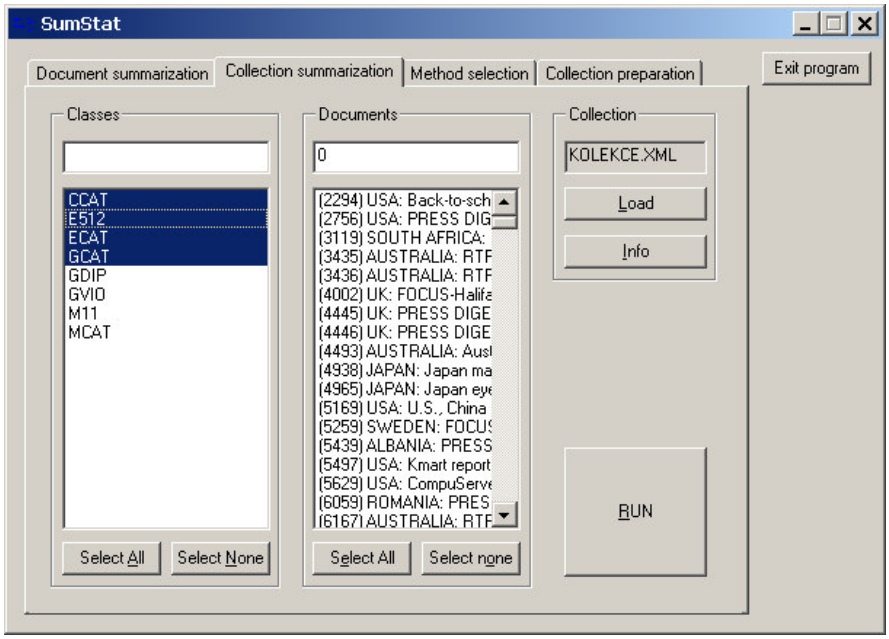

FIGURE 1 – SUMMARY EXTRACTOR – USER'S INTERFACE

FREQUENCY-BASED ABSTRACTING

The first experiments with abstracting based on term frequency (TF×IDF) date back to late 1950s. Terms occurring more frequently in a document are assigned higher weight. In

spite of this highly simplified approach, the method leads to surprisingly good results. Inverse document frequency of the term j is expressed as:

$$IDF_j = \log\left(\frac{m}{DF_j}\right) + 1,$$

where m is the total number of documents in the collection and DF_j represents the number of documents containing the term j.

Weight of term j in document i is expressed as:

$w_{ij} = TF_{ij} \times IDF_j$.

We have also used the following modification of TF×IDF for the purpose of document abstracting:

$$w_{ij} = TF_{ij} \times IDF_j = \frac{\log_2\left(\frac{words\ in\ collection}{term\ j\ occurrences\ in\ collection}\right)}{\log_2\left(\frac{words\ in\ document\ i}{term\ j\ occurrences\ in\ document\ i}\right)}.$$

ITEMSETS-BASED ABSTRACTING

Many researchers have been exploring which groups of items (itemsets) frequently appear together in transactions. Our transactions are represented by documents, and items by significant terms. The issue of itemsets and mining of association rules was described by Agrawal et al. (4).

Abstracts generated by our summarization engine should serve primarily to classify documents into predefined topic taxonomy. Using the TF×IDF method we select terms significant in a particular document, taking account neither the relations among words per se, nor associations between words and pertinent topics. These drawbacks are addressed by the itemsets method. When selecting frequent k-itemsets based on threshold c_k, we make this comparison:

$$f_{ij} = \frac{|D_{ij}|}{m_i} > c_k,$$

where f_{ij} represents frequency of itemset j in topic i, m_i is the size of topic i, and D_{ij} is a set of documents from topic i containing itemset j. Sets of itemsets are generated by means of apriori algorithm (5).

Over the course of the actual summarization, we have no a priori knowledge of the document's categorization into topics; therefore, we look for all itemsets from all topics. Each itemset is then weighed on the basis of its size and the number of topics in which the itemset is found frequent:

$$w_j = \frac{q_k}{\log T_j},$$

where q_k is user-defined weight for k-itemset, and T_j is the number of topics in which itemset j occurs frequently.

For itemsets larger than 1 it is beneficial to make adjustments for the mutual position of itemset terms in a document. As whole sentences are usually extracted from documents, it is efficient to compose itemsets of terms occurring in the same sentence only.

The question still remains how to set up threshold values for declaring an itemset frequent. For high threshold values relatively few itemsets get selected, leading to poor

summarization results. By decreasing the threshold, the number of frequent itemsets rises rapidly, resulting in very long processing time.

Regarding the size of itemsets being used, it is reasonable to look for itemsets up to the size of 4. We work with itemsets of size 1 and 2.

MUTUAL REINFORCEMENT

The mutual reinforcement technique is applied to documents containing terms already processed (weighed) by some other method. It is based on the idea that a word should be assigned higher weight if it occurs in a relatively high number of sentences with above-average weight.

First, for each document we generate a list of words and a list of sentences contained in the document. From these two lists we make a bipartite graph whose nodes are assigned weights identical to those of words and sentences already determined.

Then we perform the following iterative computation:

a) For all words u_i: $w_i = \sum_{u_i \in v_k} w_k$ $\qquad u = \frac{u}{\|u\|}$

b) For all sentences v_j: $w_j = \sum_{u_k \in v_j} w_k$ $\qquad v = \frac{v}{\|v\|}$

The above computation is stopped at the moment when values start to converge. We need to define a convergence criterion; therefore, weights of graph nodes computed in the previous step must be saved. We can express the difference:

$$d = \frac{\sum_i \left| w_{in} - w_{i(n-1)} \right|}{\sum_i w_{in}},$$

where w_{in} is the weight of node i in step n. The difference can be also expressed as:

$$d = \sum_i \frac{\left| w_{in} - w_{i(n-1)} \right|}{w_{in}}$$, or possibly as $$d = \max\left(\frac{\left| w_{in} - w_{i(n-1)} \right|}{w_{in}} \right).$$

As values converge very slowly (or not at all), we should specify the maximum number of iterations. A suitable limit is hard to define as it may depend on the length and nature of documents being used. According to our experience, a maximum of 30 iterations is satisfactory for our testing collections. For the majority of documents we have achieved convergence with d_{max} = 3 % upon 8 to 15 trials.

USE OF HEURISTICS

"Position" method

We can assume that the most important information is usually found at the beginning or end of a document. It may be also useful to work with sentence position in a paragraph, or paragraph position in a document. It may be the case (e.g. Reuters news collection being used in our experiments), that information is spread across documents with no observable patterns. The benefits of this approach are immeasurable in such a case.

This criterion is not currently implemented in our summarizer.

"Title words" method

Words contained in the title of an article, or words occurring in chapter names, are usually highly important, characterizing the whole document. These words are often contained in the document's list of keywords as well. In the implementation of our summarizer, we put higher emphasis on these words by multiplying their weights by a configurable constant c

(usually c=4). If we define a high value of c, title words usually get included in the extract, but we must exclude words occurring in titles only (not in the document's body).

This criterion is implemented in our summarizer.

"Cue Words" method

In order to use cue words (or phrases), we need to compose a specific (language-dependent) list of these terms, possibly including their weights. Because of the language dependence of this approach and its uncertain benefits, this criterion is not currently implemented in our summarizer.

COMPARISON OF METHODS IMPLEMENTED

The following notation is used in the paragraphs below:

m – the number of all terms in document collection
t – the number of distinct terms in collection
d – the number of documents in collection
c – the number of categories in collection taxonomy
m_c – the average number of terms per category
d_c – the average number of documents per category
t_c – the average number of distinct terms per category
t_d – the average number of distinct terms per document

Methods integrated into our summarizer can be compared namely in terms of time and memory requirements (discerning between the phase of collection preparation and one of the actual summarization).

"TF×IDF" method

In order to use this method, we need to create a list of all words and their frequencies in the collection. Memory requirements are therefore linear, expressed as O(t), time requirements as O(m).

"Itemsets" method

The preparatory phase consists in creating a list of all itemsets. Both time and memory requirements depend on the maximum size of itemsets being used. For 1-itemsets we must determine term frequencies for each category. Time requirements are therefore $O(c \times m_c)$. In case of mono-classification, we have complexity of O(m), otherwise it is O(m×k), where k is the average number of categories each documents is classified to.

Memory requirements for larger itemsets will be approximately the same, however, time requirements increase rapidly. We must determine the document frequency of all combinations of 1-itemsets. The time required for searching for 2-itemsets will be in the range of $O(c \times t_c^2 \times d_c)$, where the factor of $c \times d_c$ is linearly dependent on d. Time complexity can therefore be expressed as $O(d \times t_c^2)$. Complexity will also depend on the average number of categories per document (relationship between $c \times d_c$ and d), the language, and the nature of documents being used.

During the actual summarization using 1-itemsets, complexity will approximately equal that of the TF×IDF method. For 2-itemsets, time requirements increase rapidly. We must check individual 2-itemsets against all sentences. The number of sentences can be estimated from the number of words in the collection, the number of 2-itemsets increases with the second power of t_c. Time complexity will be in the order of $O(t_c^2 \times m)$.

"Mutual Reinforcement" method

This method does not require a preparatory phase. Summarization requires construction of a graph and subsequent multiplication of weights of sentences and words. The graph is represented by a list of nodes (terms and sentences) and edges linking these nodes. We have $e = u_1 \times u_2$ edges, where u_1 is the number of terms in a document and u_2 is the number of its sentences. We can neglect the number of nodes, and express memory requirements as O(e). Please note that contrary to the previous two methods, all data pertain to the current document only, ending up with very low memory requirements.

Time complexity is definitely worse. In each iteration we must parse all the edges twice and adjust weights of related nodes. The complexity for the whole collection is therefore expressed as:

$$k \times e_{av} \times d = k \times u_{1av} \times u_{2av} \times d ,$$

where k is the average number of iterations for one document, and e_{av}, u_{1av} and u_{2av} represent mean values for e, u_1 and u_2 respectively, across all documents.

Considering u_2 linearly dependent on u_1, time requirements are approximately $O(t_d^2 \times d)$.

LANGUAGE (IN)DEPENDENCE

Looking at the summarization approaches mentioned in this paper, the only language-dependent method is "cue-words". It is essential to summarize documents in various languages without user intervention. The language can be detected automatically. In our previous research we detected the language of a document by matching the document's terms with stop-lists for various languages.

COMBINATION OF SUMMARIZATION METHODS

All methods based on heuristics as well as the mutual reinforcement method are designed to be applied to results of TF×IDF and itemsets summarizers. Individual methods work with weights assigned to the document's sections of different size (i.e. paragraphs, sentences, or words). "Position" methods work directly with paragraphs.

Is it beneficial to combine TF×IDF and itemsets summarization methods? TF×IDF associates higher weights with words occurring frequently in a particular article, but not so frequently in the whole collection. The itemsets method, on the contrary, puts emphasis on words occurring frequently both in a particular document and the whole collection. The combination of these two methods is counter-productive.

RESULTS

APPROACHES TO EVALUATION

Evaluation of text summarization systems is discussed in detail by Firmin and Chrzanowski (6). Besides a number of other evaluation approaches, they mention the degree of domain independence.

There are two basic approaches to evaluation – intrinsic and extrinsic. Intrinsic testing is based on general requirements made on document abstracts, i.e. it explores their information content, coverage and correctness. Two methods of intrinsic testing are used in practice. The first method entails manual comparison of the resulting abstracts with the original documents. The other one is based on comparing resulting machine abstracts with a priori created abstracts made by humans. If the set of sentences/paragraphs selected by an automatic summarization method has a high overlap with the summary generated by humans, the automatic summarizer should be regarded as effective. We must (quite incorrectly) assume

that a human would be able to effectively identify the most important sentences or paragraphs in a document. However, there is fairly uniform acceptance of the belief that any number of acceptable abstracts could effectively represent the content of a single document. The essence of an idea can be captured by more than one sentence or phrase.

A modified version of intrinsic testing is often used: a group of human abstractors evaluates sentences in all testing documents, assigning each sentence a weight representing its suitability for inclusion in the resulting abstract. The final weight of each sentence is computed as the average value of weights assigned by individual abstractors. Upon generating an abstract automatically, sentences in this abstract are compared with sentence weights assigned by human abstractors.

The second approach to evaluation, extrinsic, is based on comparing the quality of abstracts with respect to their intended use. Extrinsic testing is the primary method used in our research experiments. We test the impacts of summarization and individual summarization methods on the success rate of document categorization. The quality of a document summarizer is measured by comparing classification of the original (full-length) document with that of the summarized document. Classification can be evaluated, for example, in terms of precision, recall or a combination of these (7).

With respect to this approach we must keep in mind that categorization is performed by a machine classifier (based on inductive machine learning) demonstrating some inherent errors. It is therefore necessary to differentiate between the error generated by a classifier and one caused by a summarizer.

The quality of an abstract can be also measured by the time required to read it. Firmin and Chrzanowski (6) compare the average time required for reading full-text documents, best summaries, and 10 % summaries.

TESTING COLLECTIONS

We tested our document summarizer using the Reuters Corpus Volume 1 (RCV1) collection (the first "official" Reuters corpus released to the community of researchers, containing over 800 thousand documents).

We prepared "Collection 1" by selecting RCV1 documents with lengths between 6,144 and 20,480 bytes. From these we excluded all documents containing tab characters (the original Reuters collection contains various lists of sports meetings unsuitable for summarization), and documents not assigned to any category. We also removed 49 unsuitable categories (i.e. those that are either non-leaf categories, or contain fewer than 100 documents).

Collection 2 was created from the original Reuters collection by selecting documents assigned to exactly one category (upon removing all non-leaf categories). Collection 2 includes documents with lengths between 5,120 and 20,480 bytes. Categories with at least 25 documents are kept (see Table 1 below).

TABLE 1 – LIST OF TOPICS IN COLLECTION 2

Category	GVIO	ECAT	GDIP	M11	E512	CCAT	MCAT	GCAT
# documents	27	287	25	43	36	1095	85	2151

Details of individual collections are shown in Table 2 below.

TABLE 2 – TESTING COLLECTIONS – DETAILS

	Collection 1	Collection 2
Number of documents	14,395	3,749

Total number of words	11,720,584	3,116,416
Number of words upon stemming	5,641,871	1,541,792
Number of distinct significant words	86,356	48,013
Average number of significant words per document	392	411
Number of categories	100	8

EXPERIMENTS

For space reasons, results applicable to Collection 2 only are shown. In our experiments we trained classifiers using 20 % of documents of the original collection, proceeding with summarization (using the setup shown in Table 3 below) and subsequent classification of abstracts.

Our novel NBCI classifier, used herein for testing, is described in (8). NBCI is based on the traditional Naïve Bayes classifier; however, we are using reduced feature space, working with frequent itemsets only, in place of all features (terms) found in documents.

TABLE 3 – COMPARISON OF BASIC SUMMARIZATION METHODS

Classifier used	NB	NBCI	Itemsets
Original documents	**87.97**	**92.32**	**74.95**
25 % sentences – random selection	86.90	89.86	70.60
18 % beginning of paragraphs, 7 % end of paragraphs	85.06	89.38	**74.26**
TF×IDF 25 %	87.89	88.10	61.87
Itemsets 25 %	86.42	**91.44**	70.15
TF×IDF key words	**88.53**	87.22	71.28
Itemsets key words	83.97	87.86	71.44

From the table we can observe interesting results for sentence extraction from the beginning/end of the file. This approach does not lead to particularly good results for NB or NBCI classifiers, but seems to be the best choice for Itemsets Classifier. This can be explained by the nature of Reuters collection documents – names of countries or newsrooms are usually contained in these parts of the file. Country and newsroom names can therefore become itemsets associated with high weights, which is important for the Itemsets Classifier in view of the relative small set of itemsets resulting from the classifier's inductive learning. These paragraphs are very short, providing insufficient information for the other two classifiers.

TABLE 4 – IMPACT OF HEURISTICS ON THE RESULTS OF ITEMSETS SUMMARIZER

	Mutual reinforcement	Position in text	Title words	NB	NBCI	Itemsets classifier
Original documents				**87.97**	**92.32**	**76.98**
Sentences (5 %)	-	-	-	81.52	83.44	58.10
	+	-	-	81.84	83.81	58.95
Sentences (15 %)	-	-	-	85.86	89.97	66.65

	+	-	-	85.62	89.22	66.59
Sentences (30 %)	-	-	-	86.32	91.12	71.22
	+	-	-	86.32	**91.17**	70.88
	-	+	-	**86.42**	91.14	**75.01**
	+	+	-	86.32	91.09	70.86
	-	-	+	86.41	91.09	72.26
Key words	-	-	-	85.73	88.26	70.42
	+	-	-	83.78	84.37	68.33

We can make several observations from the above table. First, the quality of classification of abstracts generated by the Itemsets summarizer does not depend much on the length of these abstracts. We can observe some dependency in case of Itemsets Classifier, which can be explained by a relatively small number of itemsets resulting from classifier training. Good results are achieved by combining the Itemsets Classifier with "position in text" heuristics. We can also see from Table 4 that mutual reinforcement method is not beneficial with respect to classification. In fact, none of the above heuristic approaches resulted in significant or consistent improvement.

The table below shows results for the TF×IDF summarization method. Document extracts containing 15 % of sentences were generated. We tested the impact of heuristic modifications of TF×IDF on classification results.

TABLE 5 – IMPACT OF HEURISTICS ON THE RESULTS OF TF×IDF SUMMARIZER

Mutual reinforcement	Position in text	Title words	NB	NBCI	Itemsets Classifier
Original documents			**87.97**	**92.32**	**76.98**
-	-	-	85.41	88.10	67.61
+	-	-	85.62	89.22	66.76
-	+	-	84.98	88.82	68.28
+	+	-	**85.65**	89.44	66.89
-	-	+	85.41	88.13	67.61
+	-	+	85.62	89.22	66.76
-	+	+	84.98	88.85	**73.71**
+	+	+	**85.65**	**89.49**	66.84

The results shown in Table 5 are quite balanced; nonetheless, we can observe that the use of heuristics can be beneficial. This is different from our observations applicable to the Itemsets summarizer.

DISCUSSION

Looking at the above tables, we can observe only a slight difference between the quality of classification of original documents and that of abstracts generated by our document summarizer. In other words, the core of information needed for classification is preserved in the document abstracts. Unfortunately, the way we evaluate the quality of a document

summarizer (extrinsic testing) depends on a combination of summarizer and classifier errors. By fine-tuning parameters of both applications we may certainly generate abstracts that will be classified perfectly, but we may sometimes go against the original intention: generate nice, readable abstracts.

CONCLUSIONS AND FURTHER RESEARCH

Document summarization methods have been known for a long time and new approaches occur very sporadically. Current research is therefore focused namely on modifications of the existing approaches, or their combination.

Modifications of document summarization methods include the effort to find an optimum setup of various parameters, but also the use of various thesauruses, or other dictionary-based methods (e.g. dictionaries of geographical locations, well-known personalities, etc.). When summarizing longer documents, these can be split into several sections (clusters), summarizing each section independently. The summarizer can be also improved by utilizing linguistic information contained in text documents.

We will soon combine the document summarizer presented herein with the user-profile identification system (ProGen) recently developed at our university (9), so that document summarization is optimized with respect to users' interests. The resulting system can be used, for example, to provide users with abstracts of non-visited and potentially interesting web pages.

This work has been partly supported by grants No. *MSM 235200005* and *ME494*.

NOTES AND REFERENCES

1. Hynek J., Ježek K. *Use of Text Mining Methods in a Digital Library*, In ELPUB: the Sixth International Conference on Electronic Publishing, 2002, Berlin : Verlag für Wissenschaft und Forschung Berlin, 2002, p. 276
2. Luhn H.P. *The Automatic Creation of Literature Abstracts*, IRE National Convention, New York, March 24, 1958
3. Kupiec J., Pedersen J., Chen F. *A Trainable Document Summarizer*, Xerox Palo Alto Research Center, Palo Alto, CA 94304
4. Agrawal R., Imielinski T., Swami A. *Mining Association Rules Between Sets of Items in Large Databases*. In: Proceedings of ACM SIGMOD, Washington DC, USA, 1993, p. 207-216
5. Agrawal R. Srikant R. *Fast Algorithm for Mining Association Rules*. In: Proceedings of the International Conference on Very Large Data Bases, Santiago, Chile, 1994, p. 487-499
6. Firmin T., Chrzanowski M. *An Evaluation of Automatic Text Summarization Systems*, Department of Defense, Ft. Meade MD 20755
7. Rijsbergen C.J. *Information Retrieval*, 1979. Retrieved from: http://www.dcs.gla.ac.uk/keith/Preface.html
8. Kučera M., Ježek K., Hynek J. *Kategorizace textů metodou NBCI*, In: Proceedings of the 2nd Annual Conference Znalosti 2003, Czech Republic : VŠB-Technická univerzita Ostrava, 2003, p. 33-42
9. Grolmus P, Hynek J., Ježek K. *User Profile Identification Based on Text Mining*, In: Proceedings of 6th International Conference on Information Systems Implementation and Modelling – ISIM '03 Brno, Czech Republic : MARQ, 2003, p. 109-116