Mining Textual Data in Croatian

Bojana Dalbelo Bašić, Boris Bereček and Ana Cvitaš
Department of Electronics, Microelectronics, Computer and Intelligent Systems
University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
Phone: (385) 1 6129 935 Fax: (385) 1 6129 653
E-mail: {bojana.dalbelo | boris.berecek | ana.cvitas}@fer.hr

Abstract – Business intelligence systems find textual data a very useful source of information. Text processing algorithms and systems in English and other world languages are well developed, which is not the case with Croatian language. This paper explores the applicability of existing systems and examines optimal parameters for Croatian.

The quality of input data strongly influences clustering and classification results. Experiments are significantly better run after reducing noise. The impact of input learning set size and dimensionality are also considered. Special preprocessing for Croatian language consists of morphological normalisation, a useful step towards better results. Non-croatian specialised text mining tools are also applicable.

I. INTRODUCTION

Organizations are drowning in huge amounts of data, trying to utilize them in order to make better, more informed decisions. To date, the focus has been on developing Business Intelligence (BI) systems designed to analyze data from the structured databases. By comparison the information hidden in text files has been almost totally disregarded. As BI value of text is being discovered, the focus is turning towards text mining techniques. Commercial text mining tools are now available, but due to complexity of the text they are still not equally practical as are data mining tools for numerical data. There are many motives to take on text-related challenges for BI. There is far too much critical information still inaccessible in documents which could be useful for quality analyses and important decisions. BI systems, based only on numerical values, are excellent at revealing what happened when, but are sometimes incapable of answering why. Also, document management systems work well with homogeneous collections of documents, but not with the heterogeneous knowledge faced every day. The only way that knowledge can be processed is to explore the text itself [1].

Most text mining methods are based and optimized according to the characteristics of English language and, sometimes, to those of other world languages. Croatian has been set aside and the suitability to process Croatian texts applying the methods designed for English was not explored.

The basic principles implemented in the existing methods can certainly be applied to all language structures. Our goal was to find an optimal combination of methods, parameters and preprocessing techniques and to find out if they significantly differ from those found practical for other languages.

In the academic society and business environment, SAS® system [2] is used for text and data mining. It has built-in preprocessing for English, German and French language. Similar tools are still inexistent for Croatian. Until a system specialized for Croatian language is not developed, we are exploring the suitability of SAS®.

One of the basic problems making the processing of Croatian difficult is the complexity of morphology. The difference is particularly obvious when compared with the English language. This is why textual data, before computer processing, need to be morphologically normalized. Morphological normalization is a process of conflating different forms of one word to a single representative form (e.g. English "culture": *kultura*, *kulturna*, *kulturan...*→ *kultura*) [3], [4]. Morphological normalization is also useful for processing the other languages, such as English language, but it can be predicted that it will have grater influence on more morphologically complex languages like Croatian. Morphological normalization is assumed to significantly improve the results, but its real value will be discussed after comparing results of experiments presented in this paper.

Experiments were run applying the SAS® system on huge amounts of data. Various modules integrated in the SAS® system process particular kinds of data (textual, numerical) and visually present the results. In our experiments we used the Enterprise miner module, provided for data mining, with the integrated Text Miner node specialized for text processing.

This paper is organized as follows. The structure of data collection used in experiments is described in Section II. Section III explains the most frequent text mining methods and Section IV the performance estimators. The results of experiments are presented in Section V.

II. DATA

One of the tasks of these experiments was to explore the efficiency of text mining methods in processing data in Croatian. For this purpose, we used the "Vjesnik" newspaper articles database [4], adopted from the Croatian National Corpus [5].

TABLE I
ARTICLE CATEGORIES

	Category
1.	National
2.	International
3.	Zagreb
4.	Business
5.	Culture
6.	Sports
7.	Crime File
8.	Topic of the Day

Articles are organized in eight categories, as in Table I. The efficiency of methods applied was measured by comparing the results by given categories.

In general, category labels represent well the topics of associated articles, except the *Topic of the Day* category, since it can include articles from any of the categories given in Table I. Also, an additional problem is the fact that some topics from categories such as *National* and *International* are represented much more often in this category then from others. There is much other overlapping in the collection, but none nearly as significant as this one.

The whole collection consists of 10.000 documents. All categories are equally represented, each category with 1.250 documents.

In order to explore the influence of Croatian morphology to text mining performance, the original database and its morphologically normalized version were used. The morphological normalization was executed aided by a morphological lexicon constructed by a rule-based automatic acquisition [6]. In our experiments we used a variant of the normalisation procedure having an effect similiar to that of stemming, usually performed on English texts.

III. TEXT MINING METHODS

A. Preprocessing

The database used for experiments was initially in XML format. The first step was to convert it to a CSV (Comma Separated Values) file, more suitable for input in the SAS system. The only data adopted from the initial file were category identification and the document text itself.

The complete document collection processing for large databases would be extremely time-demanding. That is the reason to create a subset of representative samples. A specified percentage or an exact number of documents can be selected by random, first n, stratified or applying other methods.

Besides decreasing the collection of documents processed in the experiments, in order to enable the proper usage of text mining methods, data is divided in three subsets. These are used for learning, validation and testing.

All the words in a text are not equally important regarding the content. Actually, some of them do not refer to the content of the text at all. The usual preprocessing method used here for this purpose is to expel these words (stop words) from the initial set of data.

In addition to the above, other useful methods for preprocessing are lists of synonyms, noun groups and identification of numbers, punctuation or terms occurring in a single document as terms [2].

B. Document Collection Representation

The entire collection of documents is transformed into a term-document matrix. It is a bag-of-words representation. All the words from the documents are included, but without any order or text structure. The bag-of-words representation can be extended by using word sequences (n-grams) instead of single words [7].

The row and column vectors of the term-document matrix A = [a_{ij}] in Fig.1. represent the words and documents in the corpus, respectively. Element a_{ij} represents the number of occurrences of the term i in the document d_i. The semantic relevance between documents is estimated by computing the angle cosine between two document vectors [8].

$$A = \begin{array}{cc} \left[\begin{array}{l} .a_{11}.... \\ ..... \\ ....a_{ij}.......... \\ . \\ a_{m1}. \end{array}\right] & \begin{array}{l} w_1 \\ \\ .w_i. \\ .. \\ w_m \end{array} \\ d_1,......d_j.....d_n & \end{array}$$

Fig.1. Term-document matrix

The elements of the term-document matrix A are usually transformed into a form more suitable for text mining purposes using the TF*IDF function. Function TF*IDF (term frequency*inverse document frequency) is used to compute the term weight (W) and is given by (1).

$$W_i = tf_i \cdot \log\left(\frac{N}{n_i}\right) \qquad (1)$$

Term frequency (tf_i) measures the frequency of occurrence of the term i in the text. N is the number of documents in the collection and n_i is the number of documents in the collection containing the term i.

The best terms are those occurring frequently in the text, but rarely in other documents of the collection. TF increases the term weight because a frequent term is expected to be more important for the content than others occurring less frequently. Term weight is also the inverse function of frequency of the documents this term is assigned to. The term occurring in many documents is not as content-relevant as those present only in specific texts. The influence of the IDF parameter is decreased by the usage of the common logarithm [9].

C. Dimensionality Reduction

One of the most successful dimensionality reduction methods is singular value decomposition (SVD). The process begins with a rectangular term-document matrix A that is decomposed into the product of three specialized matrices according to (1).

$$A = T_0 S_0 D_0{}', \qquad (1)$$

where T_0 and D_0 have orthogonal columns representing the left and the right singular vectors and S_0 is a diagonal matrix consisting of singular values [3].

These matrices represent the breakdown of the original relations into linearly independent components. Ignoring small irrelevant components, the approximate model with fewer dimensions is created. This reduced model is the best approximation of the document collection and all the term-term, document-document and term-document similarities are shown by this smaller number of dimensions [10].

Another dimensionality reduction method, called roll-up terms, selects N highly weighted terms from each document and creates the reduced model.

D. Clustering

Clustering algorithms partition a set of objects into groups or clusters with the goal to assign similar objects to the same cluster, and dissimilar objects to different clusters [3].

One of the used clustering methods is hierarchical clustering. In this method, one cluster can be completely included in the other, but no other overlapping is allowed. At the beginning of the procedure, each document is classified in its individual cluster. In the next phase, two most similar clusters are joined and the same procedure is repeated until the expected number of clusters is reached.

In Expectation Maximization (EM) method, primary and secondary clusters are formed. Primary clusters have higher density than the secondary due to a greater similarity between documents. Secondary clusters belong with the certain likelihood to one of the primary clusters [2]. In other words, this method allows degrees of membership, as well as membership in multiple clusters. It is the statistical model that starts with a set of random clusters and in each iteration estimates the parameters of the model until the expected convergence is achieved [3].

E. Classification

Classification is the process of assigning different objects to two or more classes or categories. The difference between classification and clustering is that classification is supervised learning technique and requires a set of previously classified objects for each group [3].

Unlike other classification algorithms, the K-nearest neighbor classifier uses only positive examples of the class. To classify a new object, the algorithm finds the object in a training set that is most similar and assigns the category of this nearest neighbor. Class overlapping is possible by finding *k* nearest examples.

Learning examples are points in n-dimensional space and Euclid metric is used as distance measure. In this case, the target function is presented by discrete values describing affiliation to specific categories [3], [11].

IV. PERFORMANCE ESTIMATORS

In order to assess the efficiency, it is necessary to define relations between real categories and the output of the system (confusion matrix), as in Table II. TP represents true positive, TN true negative, FP false positive, and FN false negative classified objects [12], [13].

TABLE II
CONFUSION MATRIX

		Class	
		positive	negative
Prediction	positive	TP	FP
	negative	FN	TN

Classification efficiency is usually represented with recall and precision measures. Recall (R) is the proportion of the category members that the system really assigns to that category (1).

$$R = \frac{TP}{FN + TP} \qquad (1)$$

Precision (P) is the proportion of members assigned to the category by the system that really are category members (2).

$$P = \frac{TP}{FP + TP} \qquad (2)$$

For comparing two or more classifiers, both recall and precision must be considered. The measure that combines these two factors is the F-measure (3).

$$F_\beta = \frac{(\beta^2 + 1) P \cdot R}{\beta^2 P + R} \qquad (3)$$

Factor β indicates the relative importance of recall and precision. When recall and precision have the same importance (β=1), F_β is transformed into F_1-measure (4) [8], [9].

$$F_1 = \frac{2P \cdot R}{P + R} \qquad (4)$$

In order to compute classifier efficiency for more than one category, micro-averaging may cover all of the results. Micro averaging is the process when arithmetic average of all the categories is computed [11], [12].

$$F_1^{micro} = \frac{2TP}{2TP + FP + FN} \qquad (5)$$

Performance estimators used in the following experiments were F_1 and F_1^{micro} measures.

V. RESULTS OF EXPERIMENTS

Two types of experiments were performed using the SAS® Text Miner: clustering and classification. EM Clustering and Hierarchical Clustering algorithms are used for clustering and *k-nn* algorithm for classification.

Both groups of experiments had the same type of document preprocessing: stop-words elimination and morphological normalization (described in [6]). After these preprocessing steps, documents were loaded into Text Miner. Term weights were determined by the TF*IDF measure and dimensionality reduction was performed by the SVD algorithm.

A. Clustering

This experiment can show the distribution of input data, respectively how the content of the document really conforms to the assigned category. For example, it is obvious that categories like *Sports*, *Culture*, Crime File and *Business* are very well separable. Categories *National* and *International* are less separable, and the *Topic of the Day* category is a good example of a badly chosen category. It is obvious that a certain document was assigned to the *Topic of the Day* category because of its actuality, but it will not be considered by any computer process. That is why experiments were performed with and without the *Topic of the Day* category.

Table III presents the results of EM clustering algorithm on all articles except those from the *Topic of the Day* category. The algorithm found five clusters.

TABLE III
CLUSTERS CREATED BY EM CLUSTERING ALGORITHM ON DOCUMENTS FROM 7 CATEGORIES (ALL CATEGORIES EXCEPT *TOPIC OF THE DAY*)

	Terms	**N**
1.	grad, tvrditi, reći, izjaviti, trebati, ministarstvo, kazati, čovjek, rad, vrijeme, novac, hina, banka, financija, uprava, cijena…	3031
2.	djelo, hrvat, istaknuti, otvoriti, program, prostor, suradnja, svijet, život, zagreb, europa, govor, ministar, vlada …	2244
3.	američki, banka, cijena, država, iznos, kuna, program, veto, nov, predsjednik, odluka, očekivati, država, veći …	1806
4.	automobil, bolnica, djelo, dogoditi, istražan, kazna, naći, nagrada, nesreća, obrana, odvjetnik, sudac, zatvor, županija …	862
5.	dinamo, dobar, igra, kolo, kup, hajduk, klub, liga, lopta, minuta, momčad, osvojiti, najbolji, nogomet, janica, skijaš, staza …	807

Categories *Crime File* and *Sports* were very well separated into clusters 4 and 5. Documents from cluster 3 clearly belong to the category *Business,* although it is slightly mixed up with the *National* category. Since the algorithm found only 5 clusters, the remaining categories (*National*, *International* and *Zagreb*) were placed into clusters 1 and 2, and thus were separated less successfully. In cluster 1, documents from categories *Zagreb* and *National* prevailed and in cluster 2 there were documents from categories *Zagreb* and *International* This is understandable, since the division between categories *National* and *International* is not so evident.

Hierarchical clustering can be used to determine graphical dependencies between clusters. The result of one such experiment is shown on Fig. 2. For that experiment, documents from three best defined categories were used: *Culture*, *Sports* and *Crime File*. The result of hierarchical clustering is the binary tree. It is important to note that in the resulting binary tree final clusters are only those represented by leaves of the binary tree.

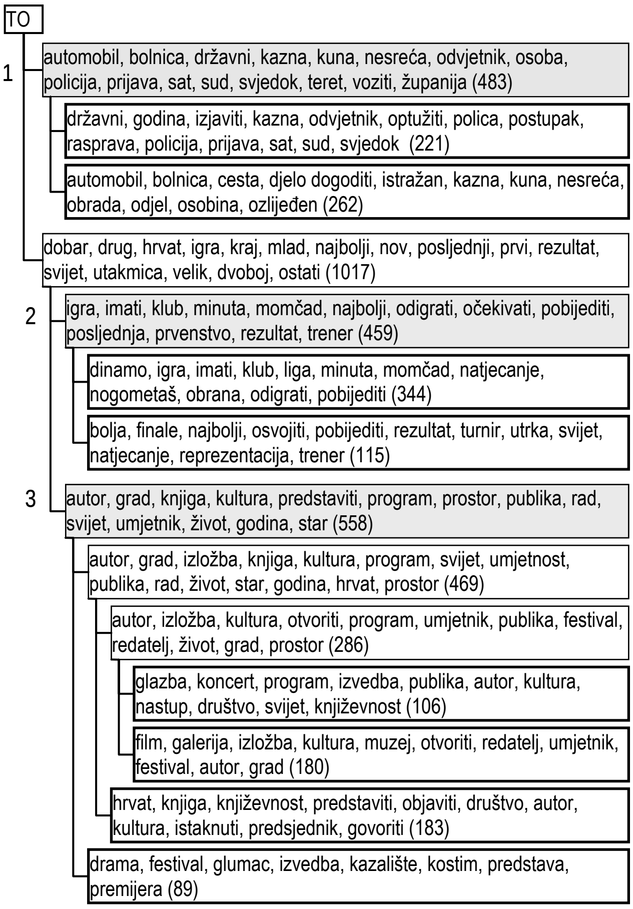

TO
1
automobil, bolnica, državni, kazna, kuna, nesreća, odvjetnik, osoba, policija, prijava, sat, sud, svjedok, teret, voziti, županija (483)
državni, godina, izjaviti, kazna, odvjetnik, optužiti, polica, postupak, rasprava, policija, prijava, sat, sud, svjedok (221)
automobil, bolnica, cesta, djelo dogoditi, istražan, kazna, kuna, nesreća, obrada, odjel, osobina, ozlijeđen (262)
dobar, drug, hrvat, igra, kraj, mlad, najbolji, nov, posljednji, prvi, rezultat, svijet, utakmica, velik, dvoboj, ostati (1017)
2
igra, imati, klub, minuta, momčad, najbolji, odigrati, očekivati, pobijediti, posljednja, prvenstvo, rezultat, trener (459)
dinamo, igra, imati, klub, liga, minuta, momčad, natjecanje, nogometaš, obrana, odigrati, pobijediti (344)
bolja, finale, najbolji, osvojiti, pobijediti, rezultat, turnir, utrka, svijet, natjecanje, reprezentacija, trener (115)
3
autor, grad, knjiga, kultura, predstaviti, program, prostor, publika, rad, svijet, umjetnik, život, godina, star (558)
autor, grad, izložba, knjiga, kultura, program, svijet, umjetnost, publika, rad, život, star, godina, hrvat, prostor (469)
autor, izložba, kultura, otvoriti, program, umjetnik, publika, festival, redatelj, život, grad, prostor (286)
glazba, koncert, program, izvedba, publika, autor, kultura, nastup, društvo, svijet, književnost (106)
film, galerija, izložba, kultura, muzej, otvoriti, redatelj, umjetnik, festival, autor, grad (180)
hrvat, knjiga, književnost, predstaviti, objaviti, društvo, autor, kultura, istaknuti, predsjednik, govoriti (183)
drama, festival, glumac, izvedba, kazalište, kostim, predstava, premijera (89)

Fig. 2. The result of hierarchical clustering of categories *Crime File*, *Sports* and *Culture* with 500 documents from each category.

Fig. 2. shows three main groups marked by numbers 1, 2, 3 and shaded. Number 1 marks the cluster representing the *Crime File* category. Number 2 marks the cluster containing documents from the *Sports* category. Number 3 represents the cluster with documents from the *Culture* category. Numbers of documents placed in each cluster are shown in brackets.

Fig. 2. shows that hierarchical clustering in SAS® Text Miner performs a very fine partition of data. For instance, cluster 3 representing the *Culture* category was divided into subcategories like Music, Film, Literature and Drama. Hierarchical clustering had very good results with categories which were very well defined.

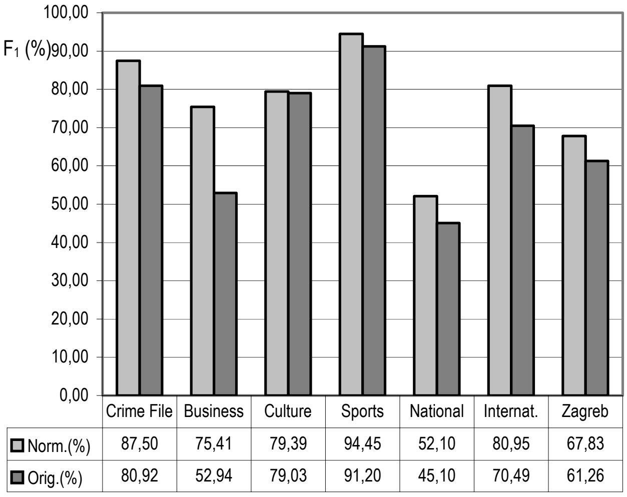
B. Classification

Classification calls for data divided in three sets: learning set, validation set and testing set. By running several experiments, it was determined that approximately 200 documents in the learning set were sufficient to achieve good classification results. By expanding the learning set with more documents, classification results were better by only few percent, but the classification time increased as well.

Dimensionality reduction was performed by the SVD algorithm. The result of SVD was a vector space with 100 dimensions. By reducing the number of dimensions, we obtained slightly better classification results, but the experiment time decreased drastically.

Classification was used to test three different features: the applicability of SAS Text Miner on articles written in Croatian, which was not originally supported by SAS Text Miner, the influence of morphological normalization, and the influence of the *Topic of the Day* category on classification results.

The comparison between results of original and morphologically normalized articles is shown in Fig. 3.



	Crime File	Business	Culture	Sports	National	Internat.	Zagreb
▢ Norm.(%)	87,50	75,41	79,39	94,45	52,10	80,95	67,83
■ Orig.(%)	80,92	52,94	79,03	91,20	45,10	70,49	61,26

Fig. 3. Comparison of classification results without the *Topic of the day* category for original and morphologically normalized articles

Fig. 3. shows that classification is better in cases when morphologically normalized articles are used, but the difference is not very significant. The reason is the SVD algorithm which decreases the influence of morphological normalization by dimensionality reduction. The influence of morphological normalization is minor on categories which were very well defined (*Crime File*, *Culture* and *Sports*), while the influence on similar categories (*National*, *International* and *Zagreb*) was much stronger.

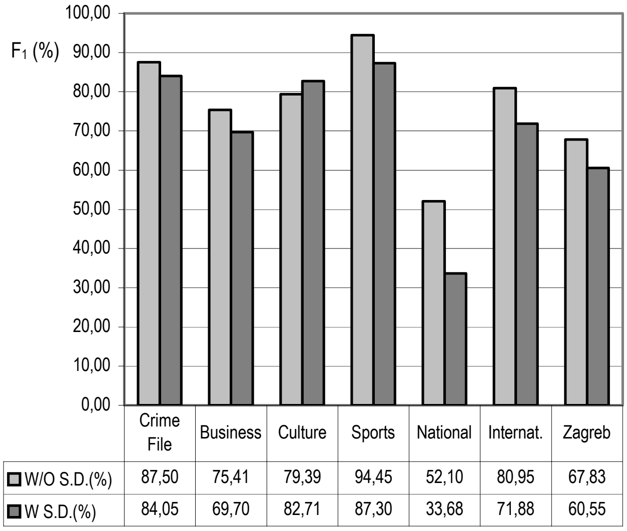
The micro-averaged measure for results on morphologically normalized articles is

$$F_1^{micro}(\text{norm.}) = 77.19\%, \qquad (4)$$

and for original articles, the micro-averaged measure is

$$F_1^{micro}(\text{orig.}) = 70.01\%. \qquad (5)$$

It can be observed in the past experiment that results are similar, so in the next experiment only morphologically normalized articles were used. The influence of the *Topic of the Day* category on classification results is shown in Fig. 4.



	Crime File	Business	Culture	Sports	National	Internat.	Zagreb
▢ W/O S.D.(%)	87,50	75,41	79,39	94,45	52,10	80,95	67,83
■ W S.D.(%)	84,05	69,70	82,71	87,30	33,68	71,88	60,55

Fig. 4. Results of classification with and without articles from the *Topic of the day* category

The results were as expected: classification results were worse for almost all categories. Categories *Crime File*, *Culture* and *Sports* had almost the same or insignificantly worse results, only showing that there were not many articles in the *Topic of the Day* category from these categories.

Contrary to the above, the results for the *National* category were significantly worse than in the experiment where *Topic of the Day* category was excluded. This shows that articles from the *Topic of the Day* category mostly belong to the *National* category.

The micro-averaged measure for morphologically normalized collection including the *Topic of the Day* category is

$$F_1^{micro}(\text{norm.+D.S.}) = 66.74\%, \qquad (6)$$

which is somewhat worse than the results of the experiment with morphologically normalized articles, but without articles from the *Topic of the Day* category.

VI. Conclusion

A. Input Data Influence on Quality of Results

Experiments have shown that clustering results are strongly influenced by input data. In situations when categories were manually very precisely and correspondingly divided with imperceptible overlapping, the results of the tested algorithms were very similar to the initial assumptions. The output clusters can be considered as very accurate, while alteration could also occur in cases when classification would be performed by a person. Significantly greater difference between the start and the output clusters is obtained when working with less clearly separated data. The algorithm is not capable of predicting that, many documents, correlated with one subject, could logically also belong to some other class. In our case, these documents belonged to *National*, *International* and *Zagreb*.

B. Influence of Learning Set Size and Dimensionality

It may be concluded that, for performed experiments on the selected database, stratified sample of 200 documents was sufficient for the learning process. By enlarging the learning set, results improve by just a few percent, with the learning time rising rapidly. According to this conclusion, learning can also be executed in situations with a lack of large amounts of classified data or when strict time limits are present.

A similar rule can be applied on the number of dimensions, as the result of the SVD algorithm. Acceptable results can already be obtained with 100 dimensions. An additional increase in the dimension number has a very inconsequentially small influence on the quality, with time demands significantly higher.

C. Influence of Noise

Comparison results of the categorization with and without *Topic of the Day* category, as the noise source, are as expected. Categories, with no *Topic of the Day* appearing, are considerably better separated. For some categories, the difference is not that clear and, despite noise presence, documents are classified properly. This effect may be explained by getting the insight into the subjects, since these topics occurred very rarely as the *Topic of the Day* category. Some other topics are often present in the *Topic of the Day*. It is impossible to divide those articles classified in their own category from those thematically equal, but grouped in *Topic of the Day*. That is why these categories are much more influenced by noise.

D. The Influence of Morphological Normalization

By comparing classification efficiency between morphologically normalized and original databases, it becomes obvious that morphologically normalized have the advantage. The application of the SVD algorithm for dimensionality reduction enables that different forms of the same word are recognized as similar terms and thus reduces the influence of morphological normalization. Categories divided less sufficiently, such as *National*, *International* and *Zagreb*, are more intensely dependent on morphological normalization.

According to the above results we conclude that despite that special preprocessing steps for Croatian language are not yet included in the SAS® systems, it is apparently suitable for Croatian language processing due to efficient dimensionality reduction techniques such as SVD. The presence of the noise in the data can significantly degrade the results and in that case morphological normalization is important to be performed.

ACKNOWLEDGMENT

We want to acknowledge SAS Adriatic for making SAS® Text Miner available through the SAS Academic Program for our research. The quantitative analysis part of our research has been conducted with SAS® Text Miner.

This work has been carried out within the projects “Text mining system” (2003-082) funded by the Ministry of Science Education and Sports of the Republic of Croatia.

REFERENCES

[1] D. Sulivan, “The Need for Text Mining in Business Intelligence”, *DM Direct Special Report*, 2004. http://www.dmreview.com/article_sub.cfm?articleId=8100 [3./3./2005.].

[2] SAS, www.sas.com [3./3./2005.].

[3] C. D. Manning, H. Schutze, *Fundations of Statistical Natural Language Processing,*, 6th Edn., MIT, Cambridge, Massachusetts, 2003.

[4] M. Tadić, *Language Technologies and Croatian Language*, Ex libris, 2003. (In Croatian).

[5] Croatian National Corpus, http://www.hnk.ffzg.hr/ [3./3./2005.].

[6] J. Šnajder, *Rule-based automatic acquisition of large-coverage morphological lexicons for information retrieval*, Technical Report, Dept. of Electronics, Microelectronics, Computer and Intelligent Systems, FER, Zagreb, 2005. (In Croatian).

[7] D. Mladenič, “How to Approach Data Analisys of Text”, *Jurnal of Information and Organizational Sciences,* vol. 28, 2004.

[8] F. Tang, “Automatic Identification in Document Indexing”, http://stat-www.berkeley.edu/users/vigre/undergrad/reports/tang.pdf [3./3./2005.].

[9] M. F. Moens, *Automatic Indexing and Abstracting of Document Texts,* University of Massachusetts, 2000.

[10] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, R. Harshman, “Indexing by Latent Semantic Analysis”, http://lsa.colorado.edu/papers/JASIS.lsi.90.pdf [5./3./2005.].

[11] E. Alpaydin, *Introduction to Machine Learning*, MIT, Cambridge, Massachusetts, 2004.

[12] M. Malenica, *Kernel Methods in Text Categorization,* Diploma thesis, University of Zagreb, Faculty of Electrical Engineering and Computing, 2004. (In Croatian).

[13] T. Joachims, *Learning to Classify Text Using Support Vector Machines,* PhD thesis, Universitat Dortmund, 2001.