Mendel
University
in Brno

MENDELU Working Papers
in Business and Economics
11/2011

Approaches to samples selection for machine learning based classification of textual data

František Dařena, Jan Žižka

MENDELU Working Papers in Business and Economics
Research Centre
Faculty of Business and Economics
Mendel University in Brno
Zemědělská 1, 613 00 Brno
Czech Republic
http://vyzc.pef.mendelu.cz/en
+420 545 132 605

Citation

Dařena, F. and Žižka, J. (2011). Approaches to samples selection for machine learning based classification of textual data. *MENDELU Working Papers in Business and Economics* 11/2011. Mendel University in Brno. Cited from: http://vyzc.pef.mendelu.cz/cz/publ/papers

Abstract

František Dařena, Jan Žižka: **Approaches to samples selection for machine learning based classification of textual data**

The paper focuses on retrieval of relevant documents written in a natural language based on availability of several candidate examples which are used as the basis for the automatic selection of only items that are similar to these predefined patterns. Presented approach should face problems related to processing user created content in natural language that include a poor control over the topic and the structure of the content and often also huge computational complexity. Three methods of selecting the best samples from a large set of candidate samples are presented – random selection, manual selection and a new approach called automatic biased sample selection, and measures based on Euclidean distance and cosine similarity are used for classification. The experiments are carried out with real world data consisting of customer reviews downloaded from amazon.com, converted to different representations based on bag-of-words procedure. The experiments and the results of the presented approach provided satisfactory values and can lead to an alternative approach to manual selection and evaluation of textual samples.

Key words

text classification, textual patterns, machine learning, natural language processing, text similarity

JEL: C38, C89

Contacts

Ing. František Dařena, Ph.D., doc. Ing. Jan Žižka, CSc., Department of Informatics/SoNet, FBE MENDELU in Brno, Zemědělská 1, 613 00 Brno, e-mail: frantisek.darena@mendelu.cz; jan.zizka@mendelu.cz

Acknowledgements

This paper is supported by the Research program of Czech Ministry of Education number VZ MSM 6215648904.

1 Introduction and objectives

Recent years have brought many opportunities to express people's opinions on a whole variety of topics through electronic channels. The places include electronic markets, recommender systems, social networks, personal blogs, discussion boards, electronic communication and others. Communication among people is represented by different kinds of textual documents. Facts contained in these documents that are useful for revealing the topic of the document can be discovered by various search engines, typically using the keywords. The results of such retrieval can be useful for individuals for finding the most suitable product (purchase decisions), identifying a community with similar interests, for web advertising companies to run successful contextual advertising campaigns, for politicians to discover public opinion, for efficient bibliographic search, for analyzing the results of marketing research and marketing intelligence activities and others (Žižka, Dařena, 2010; Broder et al., 2007; Dařena, 2007; Liu, 2006; Laver, Benoit, Garry, 2003).

Because the Web consists of huge amount of documents on diverse topics, naive queries created by the users often find matches also in many irrelevant documents. The user can obtain more relevant documents if he or she can formulate an appropriate query that consists of multiple keywords, which is often difficult for most users since it requires much more experience and skills (Oyama, Kokubo, Ishida, 2004). An alternative approach that can contribute to solving this keyword related problem can be used in the situation when the user has a few patterns (sometimes also called as models) of good examples. Using these patterns as the basis for the automatic selection of only items that are similar to the predefined (labeled) patterns, a user can collect items that belong to a relevant topic (Žižka, Dařena, 2010).

Characteristic feature of the web based document collections is their huge size. There are many problems related to processing and retrieving data from such large sets of unstructured textual data which often leads to high computational intensity. A collection of thousands of short textual entries can consist of tens of thousands of unique words and thus lead to very high dimensionality of structures used for representations of the documents.

Problems with processing user created content in natural language (like customer reviews) also embody a poor control over the topic and the structure of the content. A review that is supposed to evaluate a product can evaluate the seller, the shipping and delivery terms or any other general problem that is closely or loosely related to that product (e.g. some reviews related to a particular edition of the Bible discuss current position of Christianity, and several movie reviews discuss the book on which is the movie based). In the case that there are many textual items the probability

that such off-topic texts will have serious impact on the quality of the collection is lower. On the other hand, when the number of entries is relatively small, each of such "bad" documents can influence the collection relatively considerably. However, it is difficult to filter such bad examples automatically especially when there is no prior information about the nature of such examples. Manual process which might provide good results is, however, very demanding and sometimes can be subjectively influenced. In both cases, when the collection is large and small, bad examples cause some kind of overlapping of individual clusters formed of examples of individual classes (e.g. a review related to a book can be the same as a review of a cell phone when the only topic that is mentioned is the shipping agent, e.g. a postal service, that can be the same in the case of both products).

The paper is focused on the situation where there exist some number of candidate examples related to a given topic and it is necessary to carefully select good examples. The objective is to present a method for selecting such a set of samples that can be later used for machine learning based classification. The sample set should have a reasonable size which leads to reduced computational complexity (which is typical for processing large volumes of data) and the quality of samples should provide better results than an approach based on simple random selection. Three different methods are introduced and examined on real-world data sets created from the customer reviews at *amazon.com* e-shop. The results of two types of experiments are presented to support the findings.

2 Material and methods

2.1 Measuring the quality of classifiers

For measuring the quality of different classifiers, the values representing correctly and incorrectly classified examples are needed. In a two class classification, the classes might be labeled as positive and negative. The positive and negative examples that are classified correctly are referred as true positive (TP) and true negative (TN). False positive (FP) and (FN) represent misclassified positive and negative examples. The number representing the results of classification can be represented in a confusion matrix, see Tab. 1 (Christen, Goiser, 2007).

Tab. 1: Confusion matrix for a two-class classification

	predicted as positive	predicted as negative
actually positive class	true positive (TP)	false negative (FN)
actually negative class	false positive (FP)	true negative (TN)

Based on Table 1, further aggregate performance metrics can be defined (Gu, Zhu, Cai, 2009). Accuracy is the simplest and most intuitive measure. However, it provides no information about correct labels for of different classes and is not very suitable for imbalanced data.

$$\text{Accuracy} = (TP+TN) / (TP + FP + TN + FN)$$

Precision, also known as Positive Predictive Value is the measure of the extent the classifier was correct in classifying examples as positive.

$$\text{Precision} = TP / (TP + FP)$$

Recall assesses to what extent all examples that need to be classified as positive (negative) were so.

$$\text{Recall+ (Sensitivity)} = TP / (TP + FN)$$
$$\text{Recall– (Specificity)} = TN / (TN + FP)$$

2.2 Representation of textual data

In order to be able to classify textual data, they must be transformed to representation suitable for the learning algorithm and classification task. Textual data might be structured According to the level on which the data is analyzed, from sub-word level (decomposition of words and their morphology) to pragmatic level (the meaning of text with respect to context and situation). Ambiguities on each level can be solved using the next higher level (e.g. net level can help decide whether a word is a noun or a verb). Generally, the higher the level, the more details about the text is captured and the higher is the complexity of automatic creation of the representation. In many cases, words are meaningful units of little ambiguity even without considering the context and therefore are the basis for most work in text classification. A big advantage of word-based representations is their simplicity and straightforward process of their creation (Joachims, 2002).

In certain approaches, some of the words can be removed. These words usually include words that are very rare of very common in all classes and don't reduce the uncertainty during classification considerably. Also very short words, e.g. consisting from one or two characters can be removed. However, such an approach might require deeper analysis of the texts and might be also dependent on a particular language (Žižka, Dařena, 2010).

The texts are simply transformed to a bag of words, a sequence of words where the ordering is irrelevant. Each text example is then represented by a vector where individual dimensions

represent values of individual attributes of the text. Commonly, each word is treated as one such attribute (Joachims, 2002).

Values of attributes represent the weights of individual words (terms) in corresponding texts. Several possible methods for determining the weights of the words can be used (Nie, 2010):

- the weights are binary (0 or 1), representing the presence or absence of the term
- the weights correspond to numbers of times the word appeared in the text (term frequencies)
- the weights are calculated According tf*idf weighting schema, with the general idea that the more a term appears in a text, the more is important (tf factor), and the less the word is common among all texts, the more is specific and thus important (idf factor). Inverse document frequency (idf) can be calculated as

$$idf(t_i) = \log N / n(t_i),$$

where t_i is the term, N is the number of texts in the collection, and $n(t_i)$ is the number of texts containing term t_i (also called document frequency). To prevent a bias towards longer documents (having higher number of words), the measure for relative importance of the j-th word in the i-th document can be calculated as follows:

$$tf_{ij} = n_{ij} / \sum n_j$$

n_{ij} is the number of occurrences of word i in document j and $\sum n_j$ the number of all words in document j.

2.3 Example based document classification based on similarity

The similarity of textual documents can be measured as a distance, *L*, between the multidimensional points created by individual items. The coordinates of these points is given by the values of vectors used for representation of the documents. The closer the points appear, the more similar the text items are (Srivastava, Sahami, 2009). The simple computation employs the Euclidean distance L_E between two text documents, *j* and *k*, for each *i*-th pair of words $w_{j,i}$ and $w_{k,i}$ (i.e. dimensions of the vectors) within the two documents being processed (*m* is the number of unique words, i.e. the vectors sizes):

$$L_E = \sqrt{\sum_{i=1}^{m} (w_{j,i} - w_{k,i})^2} .$$

Alternatively, other measures can be also used, for example, the cosine (dot-product) similarity L_C based on an angle between vector pairs (Duda, 2004):

$$L_C = \arccos \frac{\vec{d}_j \cdot \vec{d}_k}{|\vec{d}_j| \cdot |\vec{d}_k|},$$

where L_C is actually the angle between vectors d_j and d_k. If $L_C = 0$, then both vectors are similar at most (zero angle), and for $L_C = \pi$ the vectors are similar at least.

The presented approach is inspired by the nearest neighbor algorithm, *k*-NN (Duda, 2004), which is a popular classification method that is often applied also to the text categorization (Hroza, Žižka, 2005). During the training phase the labeled samples of individual classes are stored. For each new unlabeled document its distance to all labeled samples is computed and then the $k \geq 1$ nearest patterns (neighbors) assign a respective label to it according to the most frequent category of its *k*-nearest neighbors.

A special case is the situation when the user has only a collection of examples on one positive (good) class and when it is desirable to find relevant text items from a collection of all kinds of unstructured natural-language textual documents. The user typically cannot process and utilize all relevant available entries and thus settles for a reasonable number of relevant entries. Unlabeled items can be therefore ranked in compliance with their similarity to the available positive patterns so the most similar items are at the top of the rank, and the least similar towards the bottom. Then, a user can expect the most relevant items near the rank top. It is up to a user's decision how many top-ranked items she or he selects or accepts (Žižka, Dařena, 2010). Such an approach based on processing only one class of texts is demonstrated in Experiment 1, see below.

2.4 Description of experiments

2.4.1 Data preparation

The textual data for the analysis were downloaded from customer review blogs on *amazon.com*. The authors decided to examine the approaches for sample selection on more than one data set. Therefore, the data with different characteristics and topics were considered (see Table 2). The products were selected relatively randomly, the intention was to have data sets with different topics and review lengths, but with enough reviews.

Tab. 2: Characteristics of analyzed data

Data source number	*Product*	*Product type*	*Length of reviews*	*Number of reviews*
1	Boldtext Pew Bible: King James Version	book	long	259
2	Toshiba Portable External Hard Drive	hardware	short	226
3	War of the Worlds	movie	medium	264

Amazon reviews contained following information (mandatory information is marked by *):

- title (*),
- text (*),
- author (*),
- rating – one to five stars (*),
- helpfulness expressed by other customers,
- comments by other customers,
- date (*).

For this experiment, only the text of the reviews was considered, although the remaining information might be useful and used for various analyses as well. The reason for considering only the text is the optionality of some other parts of reviews and the fact, that some pieces of information are not generally available in different types of systems that are sources of textual entries (e-shops, blog archives, newspaper articles etc.).

The texts of the reviews were cleaned so they contained only regular words (i.e. all HTML tags and entities, numbers, punctuation and other symbols were removed) and then converted to bag-of-words representation with following characteristics:

- minimal length of words – 1 character,
- minimal frequency of words in all reviews – 1,
- no stop words were removed,
- vectors representing the reviews contained term frequencies (TF).

All customer reviews were separated into two groups:

- a group of potential samples that was later used for selecting reviews that became the samples – set P,
- remaining reviews that were used for testing the quality of the samples – set T.

The former group (P) contained one hundred texts from which fifteen sample reviews were selected. The number fifteen was selected from several reasons:

- For manual selection of samples, it was very difficult to select very low number of the best samples (e.g. five) because the reviews were sometimes very heterogeneous even when they were highly related to given topic (e.g. they focused more on different aspects of the product that on the others). Selecting bigger number of samples was sometimes also not very easy because the quality of some data sets was not too high (some of the reviews were very similar, some was quite off topic etc.).
- The number was relatively small so the number of calculations is not very high and the results can be provided in a reasonable time.
- The number was relatively sufficient for having a representative set of samples. (Žižka, Svoboda, Dařena, 2011).

The authors successively used three different methods for selecting the samples that were later tested for their quality and obtained three sample sets:

- set R – was obtained using automated random selection from set P,
- set M – contained the samples that were selected manually from P with the intention to include the best possible samples,
- set B – was formed of samples that were selected through the process of automatic biased selection (described below).

Random sample selection

From the group of reviews (set P), desired number of reviews was randomly selected by the computer. The authors had no control over this selection process.

Manual sample selection

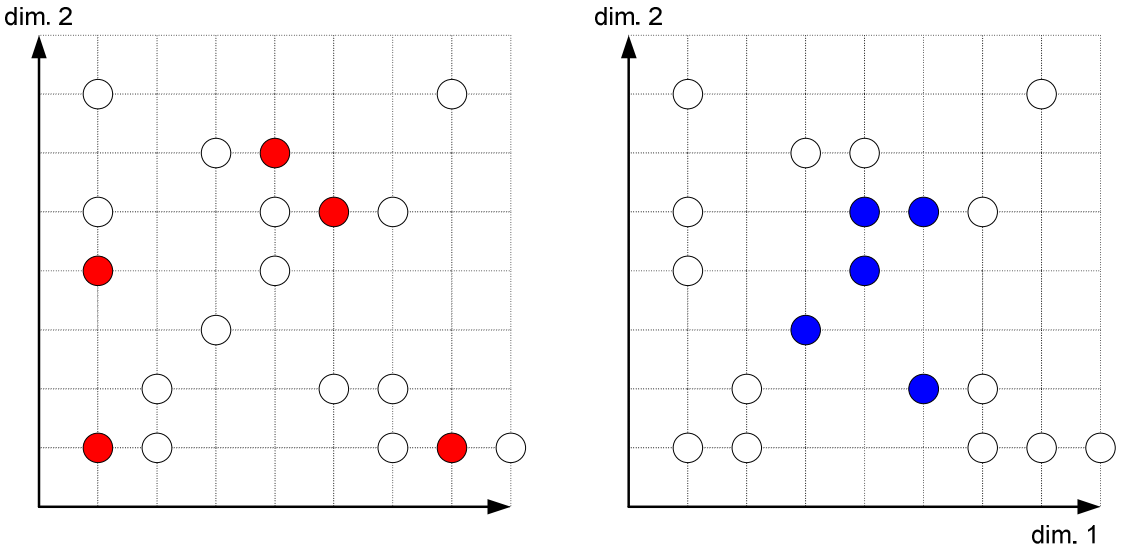
The authors examined each from one hundred reviews in the sample candidates set P. Fifteen reviews that were (according their opinion) most closely related to the corresponding product (topic) were selected.

Automatic biased sample selection

The idea of automatic biased sample selection is based on the hypothesis that the textual entries that are near the center of the group of entries of given class in k-dimensional space (where k is

the size of vocabulary for all texts) represent the class better than randomly selected documents from that class. This is demonstrated in Fig. 1, for the illustrativeness only two dimensions are considered.

Fig. 1: objects of one class represented by their position in a two-dimensional space



a) *(b)*

(a) red circles represent sample objects selected randomly, (b) blue circles represent sample objects selected with the bias (they are more in the "center of gravity" of the entire group).

2.4.2 Experiment 1

Creating the sample sets

Following operations were carried out for each class of the texts (i.e. for book, movie and hardware reviews) separately. For each text $t_i \in$ P the sum of distances to all other texts was calculated according to following formula:

$$D^i_{sum} = \sum d(t_i, t_j)\text{, for j = 1..n,}$$

where n is the number of texts, t_j is j-th text and d is the distance/similarity function. In our experiments, Euclidean distance and cosine similarity were considered as allowed alternatives for function d. Subsequently, only desired number of texts with the highest D^i_{sum} for cosine similarity or lowest D^i_{sum} for Euclidean distance were selected to form the sample set for given class (in this case 15 best texts were selected). These texts formed the set B. Set R was created by random selection and set M by manual selection.

Matching documents with samples

Remaining texts used for testing (set T) were compared to each of the documents in the sample set. Three such comparisons were carried out – for manually, randomly and with bias selected samples (sets M, R, and B). For each text t_i from the set of testing documents (set T) two similarity measures were calculated:

- the total similarity measure,
- the k-NN similarity where k=1.

The k-NN similarity S_i^N for text t_i was calculated as

$$S_i^N = nearest(d(t_i, t_j)) \text{ for j = 1..m,}$$

where t_j is *j*-th text from set T, *m* is the number of documents in the sample set (in our experiments *m* = 15), *d* is the distance/similarity function and *nearest* a function that selects the nearest document (for cosine similarity *nearest* ~ *max*, for Euclidean distance *nearest* ~ *min*).

The total similarity measure S_i was calculated as

$$S_i = \sum d(t_i, t_j) \text{ for j = 1..m,}$$

where t_j is *j*-th text from set T, *m* is the number of texts in the sample set (in our experiments *m* = 15), and *d* is the distance/similarity function.

The texts from set T were sorted according their similarity to all of the sample sets (R, M, and B). The most similar text had number 1, the second most similar number 2 etc., all texts were associated with six values measuring their similarity – similarity to three different samples using two methods, the nearest neighbor similarity S_i^N and the total similarity measure S_i.

2.4.3 Experiment 2

In the second experiment the texts from pairs of classes were processed together. For each of the classes (reviews for book, movie and hardware) the sample sets and testing sets were created as in the first experiment. In this experiment, the sample sets R, M, and B and testing set T had texts with two different labels. However, they can be considered set R_1 and R_2, T_1 and T_2 etc. Both testing sets were mixed together and each of the tested texts was then compared to samples (two sets representing samples for each class). Because the sample sets were created in three different

ways, three comparisons were made with each tested text. After the comparison, the tested text was assigned to the class of the most similar sample. Because the tested texts were labeled it was possible determine whether the text was marked correctly or not.

During the experiment, classification measures were calculated. For the random selection, the selection and matching processes were repeated ten times and the classification measures were averaged.

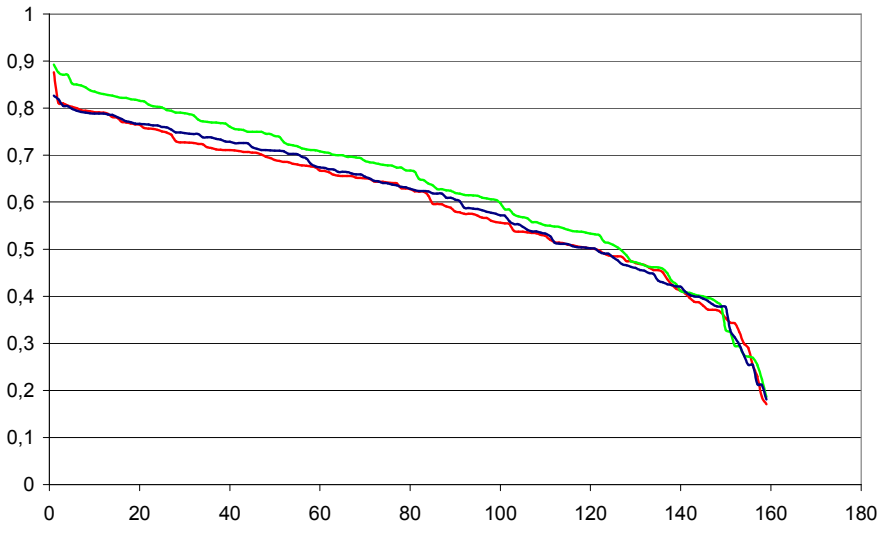
During this experiment, the texts were represented by two different representations – word frequencies and TF-IDF weights – to show whether the quality of sample selection methods are not dependent on the text representation.

3 Results

3.1 Experiment 1

The texts from set T were sorted according their similarity to the sample sets (one text could be of course ranked differently when compared to different samples) and the results of comparisons to differently created samples were analyzed to find out how the process of sample selection influences the similarity.

Fig. 2: Comparison of differently created samples using the k-NN (k=1) cosine similarity measure

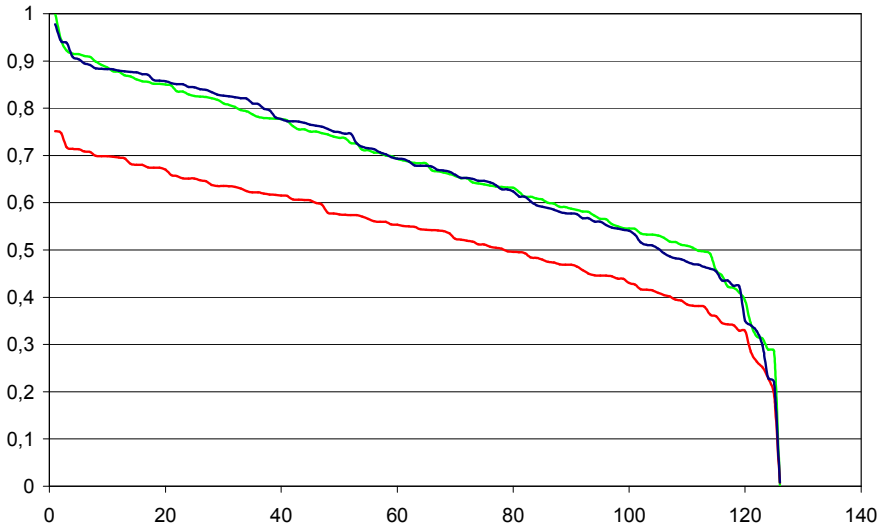


Red – comparison to R, green – comparison to M, blue – comparison to B. Vertical axis – value of S_i^N, horizontal axis – document number. Data source 1 (book).

When the cosine similarity measure was used, the similarity between the text and the sample set should be the highest at least close to the top of the list of ordered texts. When the Euclidean distance was used, the similarity should be the lowest for the most similar texts. At the end of the list the results were naturally worse because some of the texts were off topic, used very specific language of showed other deficiencies. When the ranking by similarity is displayed in a graph the curve that lies above another curve represents comparison to sample set with higher quality for cosine similarity and worse quality for Euclidean distance. Graphical representations of selected comparisons are shown in Figures 2 – 4.

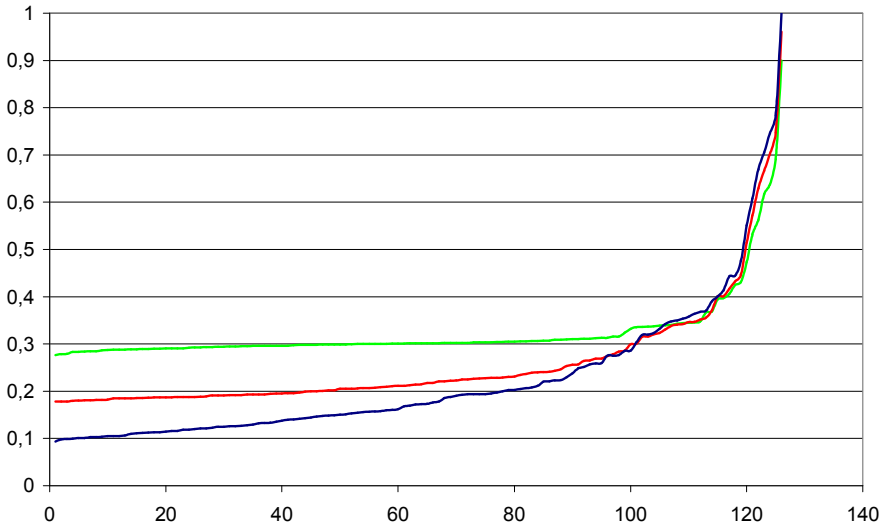
Using the k-NN (k=1) similarity/distance measure S^N the differences among differently selected samples were not very obvious (the curves were very close to each other), see Fig. 2. Therefore the total similarity/distance measure *S* was use to evaluate the methods of samples selection (see Fig. 3 and Fig. 4).

Fig. 3: Comparison of differently created samples using the total cosine similarity measure



Red – comparison to R, green – comparison to M, blue – comparison to B. Vertical axis – normalized value of S_i (all values were divided by the biggest value of S_i), horizontal axis – document number. Data source 2 (hardware).

Fig. 4: Comparison of differently created samples using the total Euclidean distance



Red – comparison to R, green – comparison to M, blue – comparison to B. Vertical axis – normalized value of S_i (all values were divided by the biggest value of S_i), horizontal axis – document number. Data source 2 (hardware).

The right ends of the graphs showed a significant change of the slope of the lines. This was caused by the fact that several texts were very short, off-topic or contained other deficiencies and thus were very different from the samples representing given document classes. More important were the texts with low numbers, i.e. texts most similar to the samples. The more to the left part of the graphs, the more important the texts were and the difference between the results based on comparisons with texts from differently selected samples was relevant.

In the case of using the cosine similarity measure the results provided by comparison to R were the worst. Samples selected manually (M) and with bias (B) provided very similar results. However, the effort of creating both sample set was incomparable – automatic creation by the sample set could be done by the computer within a few seconds without human interaction. Euclidean similarity provided completely different results. In this case, the manual selection of samples provided the worst results, even worse than for randomly selected samples. Comparisons with set B provided the best results for this kind of measure.

About one third of texts from T matched the samples in M and B better than how all texts from T matched samples from R using cosine similarity measure. In the case of Euclidean distance similarity measure, almost one half of the texts from T were matched to B better then all documents to R. Because the texts that were in top N (N is typically a relatively small number) best matching documents are usually relevant, these findings provide a good potential for future research.

3.2 Experiment 2

Following tables (Tab. 3 – Tab. 8) show the results of classification of testing data into classes based on comparisons to sample sets created in three different ways mentioned above. Each table represents the results of experiments with different pairs of classes and for different methods of representing the texts (term frequency and TF-IDF). Column Acc contains the Accuracy, T(*x*) and F(*x*) represent the samples of class *x* that are classified correctly/incorrectly – true/false. Prec(*x*) and Rec(*x*) represent aggregate metrics Precision and Recall for corresponding classes.

Tab. 3: Classes Book (1) and Hardware (2), text represented by TF

Sample selection method	Acc	T(1)	T(2)	F(1)	F(2)	Prec(1)	Prec(2)	Rec(1)	Rec(2)
R	0.74	74.7	74.1	25.9	25.3	0.75	0.75	0.75	0.74
M	0.81	81.0	81.0	19.0	19.0	0.81	0.81	0.81	0.81
B	0.81	72.0	89.0	11.0	28.0	0.87	0.76	0.72	0.89

Tab. 4: Classes Book (1) and Movie (3), text represented by TF

Sample selection method	Acc	T(1)	T(3)	F(1)	F(3)	Prec(1)	Prec(3)	Rec(1)	Rec(3)
R	0.81	86.2	75.4	24.6	13.8	0.78	0.85	0.86	0.75
M	0.88	90.0	85.0	15.0	10.0	0.86	0.90	0.90	0.85
B	0.83	79.0	87.0	13.0	21.0	0.86	0.81	0.79	0.87

Tab. 5: Classes Hardware (2) and Movie (3), text represented by TF

Sample selection method	Acc	T(2)	T(3)	F(2)	F(3)	Prec(2)	Prec(3)	Rec(2)	Rec(3)
R	0.85	92.4	76.9	23.1	7.6	0,80	0.91	0,92	0,77
M	0.87	92.0	82.0	18.0	8.0	0.84	0.91	0.92	0.82
B	0.84	92.0	76.0	24.0	8.0	0.79	0.91	0.92	0.76

Tab. 6: Classes Book (1) and Hardware (3), text represented by TF-IDF

Sample selection method	Acc	T(1)	T(2)	F(1)	F(2)	Prec(1)	Prec(2)	Rec(1)	Rec(2)
R	0.87	86.8	87.5	12.5	13.2	0.89	0.88	0.87	0.88
M	0.93	98.0	88.0	12.0	2.0	0.89	0.98	0.98	0.88
B	0.93	93.0	92.0	8.0	7.0	0.92	0.93	0.93	0.92

Tab. 7: Classes Book (1) and Movie (3), text represented by TF-IDF

Sample selection method	Acc	T(1)	T(3)	F(1)	F(3)	Prec(1)	Prec(3)	Rec(1)	Rec(3)
R	0.87	93.7	80.2	19.8	6.3	0.84	0.93	0.94	0.80
M	0.94	95.0	93.0	7.0	5.0	0.93	0.95	0.95	0.93
B	0.94	91.0	96.0	4.0	9.0	0.96	0.91	0.91	0.96

Tab. 8: Classes Hardware (2) and Movie (3), text represented by TF-IDF

Sample selection method	Acc	T(2)	T(3)	F(2)	F(3)	Prec(2)	Prec(3)	Rec(2)	Rec(3)
R	0.93	95.2	91.1	8.9	4.8	0.92	0.95	0.95	0.91
M	0.98	97.0	98.0	2.0	3.0	0.98	0.97	0.97	0.98
B	0.98	97.0	98.0	2.0	3.0	0.98	0.97	0.97	0.98

In all cases when a smaller set of sample data was created randomly from a larger set of potential samples, the results that were obtained had the smallest Accuracy, and also other metrics usually provided values worse than other methods. Both manual and automatic biased methods for sample selection provided better results. The method of representation of text entries (term frequencies and TF-IDF) provided different results – TF-IDF representation provided better results by decreasing relative importance of terms appearing in a high number of texts. However, the results offered by three different approaches for sample selection remained in the same relation – the random selection was generally the worst.

4 Discussion and conclusion

In all cases when a smaller set of sample data was created randomly from a larger set of potential samples, the results that were obtained had the smallest accuracy. Better results were obtained when the smaller sample sets were created using manual or automatic biased selection.

During the process of manual selection, several difficulties were discovered:

- some of the reviews were not addressing the product but rather the seller or the way how the product was purchased or shipped,
- some of the reviews were addressing one selected problem related to the product (e.g. the installation of the hard disk), a general problem related to entire group of products (e.g. problems with data backup and recovery) or a field related to the product (e.g. the problem of faith, religion and Christianity which is the topic related to the Bible, problems with reading the book before the movie in the case of movie reviews).

Manual selection is also always connected to several other issues. On one hand, the reviews that are off-topic or that show other deficiencies can be eliminated quite easily. On the other hand, selection of the best samples and deciding which reviews are still good enough and which are not, is not always clear and is always subjectively influenced. Also, in the case when the reviews are long (sometimes several hundred of words), manual selection can be very demanding and can last inadequate time. Further, mutual comparisons of two or more textual documents with such a long content (often with different sub-topics) and assessing their quality becomes infeasible.

Presented approach thus provides and alternative approach to manual selection and evaluation of textual samples. Preliminary experiments show that the measures of classifier quality for the presented method are close or better to the classification based on manual data preparation. The documents filtered and retrieved using presented method can be of course further processed (using e.g. the keyword based search) and the presented method then can be just a part of a sequence of document processing activities.

References

Broder, A., Fontoura, M., Josifovski, V., Riedel, L. (2007). A semantic approach to contextual advertising. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp 559–566.

Christen, P., Goiser, K. (2007). Quality and complexity measures for data linkage and duplication. *Studies in Computational Intelligence*, 48, pp. 127–151.

Dařena, F. (2007). Global architecture of marketing information systems. *Agricultural Economics*, 52, 9, pp. 432–440.

Duda, R. O. (2004). *Pattern Classification*. 2nd Edition. John Wiley and Sons. ISBN 0-471-70350-8.

Gu, Q., Zhu, L., Cai, Z. (2009). Evaluation Measures of the Classification Performance of Imbalanced Data Sets. *ISICA 2009*, 51, pp. 461–471.

Hroza, J., Žižka, J. (2005). Selecting Interesting Articles Using Their Similarity Based Only on Positive Examples. In *CICLing-2005, Proceedings of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics*. Mexico City: Springer, pp. 608–611.

Joachims, T. (2002). *Learning to classify text using support vector machines*. Norwell: Kluwer Academic Publishers. ISBN 079237679X.

Laver, M., Benoit, K., Garry, J. (2003). Extracting policy positions from political texts using words as data. *American Political Science Review*, 97, pp. 311–331.

Liu, B. (2006). Web data mining; Exploring hyperlinks, contents, and usage data. Springer. ISBN 978-3-642-19459-7.

Nie, J. Y. (2010). Cross-Language Information Retrieval. *Synthesis Lectures on Human Language Technologies*, 3, 1, pp. 1–125.

Oyama, S., Kokubo. T., Ishida, T. (2004). Domain-Specific Web Search with Keyword Spices. *IEEE transactions on knowledge and data engineering*, 16, 1, pp. 17–27.

Srivastava, A. N., Sahami, M. (Eds.), (2009). Text Mining: *Classification, Clustering, and Applications*. London, New York: Chapman Hall/CRC. ISBN 1-420-05940-8.

Žižka, J., Dařena, F. (2010). Automatic Sentiment Analysis Using the Textual Pattern Content Similarity in Natural Language. *Lecture Notes in Artificial Intelligence*, 6231, 1, pp. 224–231. ISSN 0302-9743.

Žižka, J., Svodoba, A., Dařena, F. (2011) Selecting text entries using a few positive samples and similarity ranking. *Acta Universitatis agriculturae et silviculturae Mendelianae Brunensis*, LIX, 4, pp. 399–408. ISSN 1211-8516.