Computing and Informatics, Vol. 32, 2013, 949–967

APPROACHES TO SAMPLES SELECTION FOR MACHINE LEARNING BASED CLASSIFICATION OF TEXTUAL DATA

František Dařena, Jan Žižka

Department of Informatics
Faculty of Business and Economics
Mendel University in Brno
Zemědělská 1
613 00 Brno, Czech Republic
e-mail: {frantisek.darena, jan.zizka}@mendelu.cz

Communicated by Deepak Garg

Abstract. The paper focuses on the process of selecting representative sample documents written in a natural language that can be used as the basis for automatic selection or classification of textual documents. A method of selecting the examples from a larger set of candidate examples, called automatic biased sample selection, is compared to random and manual selection. The methods are evaluated by experiments carried out with real world data consisting of customer reviews, with different document representations and similarity measures. The presented approach, that provided satisfactory results, faces problems related to processing user created content and huge computational complexity and can be used as an alternative to manual selection and evaluation of textual samples.

Keywords: Text classification, textual patterns, machine learning, natural language processing, text similarity, information retrieval

Mathematics Subject Classification 2010: 68U15, 68T50

1 INTRODUCTION

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 [1, 6, 13, 14, 21].

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 [18]. A 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 [22]. The problems are more obvious when a ranking-based model that is more effective than a simple Boolean model is used [2]. Problems with processing user created content in natural language (like customer reviews) also embody a poor control over the topic, the structure of the content and the language. 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 the current position of Christianity, and several movie reviews discuss the book on which the movie is based). The text written by many internet users is therefore often unsuitable as data for Natural Language Processing tasks such as machine translation, information retrieval and opinion mining [5]. Retrieving acceptable amount of relevant documents is therefore related to several problems – huge amount of existing irrelevant documents, high computational complexity, and problems related to control over the content of natural language documents.

The paper focuses on an alternative approach to information retrieval that can contribute to solving the above-mentioned problems. It can be used in the situation when the user has a few patterns (sometimes also called 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. Such a procedure using a ranking function obtained by machine learning algorithms is an important component of information retrieval systems. However, such methods may perform poorly when human relevance judgments are not available [8]. The previous experiments showed that the mentioned ranking-based method using the similarity between members of a relatively small subset

of labeled patterns and unlabeled text-data samples provided acceptable results given the achieved accuracy [21, 23] for document retrieval and also for document classification.

When the samples are collected during a process over which the user has not a complete control the quality of each sample document might not be high (when a document contains a certain keyword it might be not necessarily related to the topic in which is the user interested). In the case that there are many textual items in the labeled sample collection the probability that some irrelevant documents will have a 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" examples 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 when processing documents of more than one class (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, that can be the same in the case of both products).

The paper is focused on a situation when it is necessary to carefully select *good examples* from a bigger number of *candidate* examples related to a given topic. The objective is to present a method for selecting such a set of samples that can be later used for *machine learning based classification* or *ranking based information retrieval*. The set of good examples 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 a 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 REPRESENTATION OF TEXTUAL DATA

In order to be able to process the data using machine learning algorithms, they must be transformed into a representation suitable for the algorithm and the particular task. Textual data might be generally structured according to the level on which the data are analyzed, from sub-word level (decomposition of words and their morphology) to the pragmatic level (the meaning of the text with respect to context and situation). Ambiguities on each level can be resolved on the following higher level (e.g., syntactic level can help decide whether an English word is a noun or a verb). Generally, the higher the level, the more details about the text are 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. The texts are simply transformed into a bag of words, a sequence of words where the ordering is irrelevant. Each document 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 [12].

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 [17]:

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

$$idf(t_i) = \log \frac{N}{n(t_i)},$$

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

$$tf_{ij} = \frac{n_{ij}}{\sum n_j},$$

where 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. The weight of the i^{th} word in j^{th} document is then calculated as

$$w_{ij} = tf_{ij} \cdot idf(t_i).$$

The quality of vector representation can be increased by using n-grams, enhancing by semantics, removing very frequent of very infrequent words, removing stop words, application of stemming, and others. Although using n-grams, syntactical phrases, stemming, and stop words removal can influence the results of text mining algorithms, their effects are often marginal [9].

3 EXAMPLE BASED DOCUMENT RANKING AND CLASSIFICATION BASED ON SIMILARITY

To be able to perform ranking or classification of documents based on existing available patterns the way how to measure mutual document similarity must be defined.

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 are given by the values of vectors used for representation of the documents. The closer the points appear, the more similar the text items are [20]. 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 [7]:

$$L_C = arccos \frac{\overrightarrow{d_j} \cdot \overrightarrow{d_k}}{|\overrightarrow{d_j}| \cdot |\overrightarrow{d_k}|},$$

where L_C is actually the angle between vectors $\overrightarrow{d_j}$ and $\overrightarrow{d_k}$. 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 [7], which is a popular classification method that is often applied also to text categorization [11]. 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 of 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. This situation is known as a one-class classification [16]. 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 [21]. Such an approach based on processing only one class of texts is demonstrated in Experiment 1, see below.

4 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 to as true positive (TP) and true negative (TN). False positive (FP) and false negative (FN) represent misclassified positive and negative examples [4].

Based on these values further aggregate performance metrics can be defined [10]. *Accuracy* is the simplest and most intuitive measure. However, it provides just overall information about correctly assigned labels for all classes and is not very suitable for imbalanced data.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

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

$$Precision = \frac{TP}{TP + FP}$$

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

$$Recall + (Sensitivity) = \frac{TP}{TP + FN}$$

$$Recall - (Specificity) = \frac{TN}{TN + FP}$$

5 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 1). The products were selected relatively randomly, the intention was to have data sets with different topics and review lengths, but with enough reviews.

Data source No.	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

Table 1. Characteristics of analyzed data

Amazon reviews contained the 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 the 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 text of the documents used in the experiments was cleaned so it 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 the following characteristics:

- minimal length of words – 1 character (words of all lengths were preserved),
- minimal frequency of words in all reviews – 1 (rare words were not removed),
- no stop words were removed,
- vectors representing the reviews contained term frequencies (TF) in experiment 1 and also term frequencies weighted by *idf* (tf-idf representation) in experiment 2.

6 DESCRIPTION OF EXPERIMENTS

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 randomly selected reviews from which fifteen sample reviews were subsequently selected. The number fifteen was chosen for several reasons:

- For manual selection of samples, it was very difficult to select a very low number of the best samples (e.g. five) because the reviews were sometimes very heterogeneous even when they were highly related to a given topic (e.g., they focused more on different aspects of the product than the others). Selecting bigger number of samples was usually also not very easy because the quality of some data sets was not too high (some of the reviews were very similar, some were quite off-topic, etc.).
- The number was relatively small so the number of calculations was not very high and the results could be provided in a reasonable time.
- The number was sufficient for having a representative set of samples [23].

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

- set R – was obtained using automated random selection,

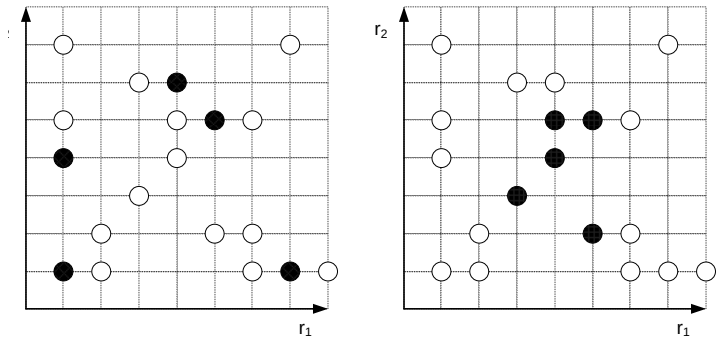
- set M – contained the samples that were selected manually with the intention to include the best sample representatives,
- set B – was formed of samples that were selected through the process of automatic biased selection (described below).

All methods used for samples sets creation were later tested for the quality of the samples they provide. This process is described in Sections 6.1 and 6.2.

Random sample selection. From the group of potential sample reviews (set P), the desired number of reviews were 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 a 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 graphically demonstrated in Figure 1, for an illustration only two dimensions are considered.



a) b)

Figure 1. Objects (white and black circles) of one class characterized by values of their attributes r_1 and r_2, represented by the positions in a two-dimensional space: a) black circles represent sample objects selected randomly, b) black circles represent sample objects selected with the bias (they are in the "center of gravity" of the entire group).

6.1 Experiment 1: One Class Classification

In the first experiment, the reviews from different classes were processed separately and the following operations were carried out for each class of the reviews (i.e. for book, movie, and hardware):

- creating sample sets B, M, and R,
- matching testing documents with samples,
- evaluating the results.

Creating sample sets. For each text $t_i \in P$ the sum of distances to all other texts was calculated according to following formula:

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

where n is the number of texts in P, t_j is the 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 the 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 a 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. Sets B, M, and R were not necessarily disjunctive.

Matching testing documents with samples. The 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:

- S_i – the total similarity measure (the sum of values of distance/similarity measure for all samples). The S_i similarity for text t_i was calculated as

  $$S_i = \sum_{j=1}^{m} d(t_i, t_j)$$

  where t_j is the j^{th} text from set T, m is the number of documents in the sample set (in our experiments $m = 15$), and d is the distance/similarity function.
- S^1_i – the value of distance/similarity measure for the best match (this is actually the k-NN similarity where $k = 1$). This similarity measure for document t_i was calculated as

  $$S^1_i = \mathit{nearest}(d(t_i, t_j)) \text{ for } j = 1..m$$

where t_j is the j^{th} text from set T, m is the number of texts 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 $\sim$ max, for Euclidean distance nearest $\sim$ min).

Evaluating the results. The texts from set T were sorted according to their similarity to each of the sample sets (R, M, and B) and an ordinal value was assigned to them. 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 total similarity measure S_i and the similarity S_i^1 for the best match.

6.2 Experiment 2: Two-Class Classification

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 in the same way as described in the previous section. In this experiment, the sample sets R, M, and B and testing set T contained texts with two different labels. Thus, they could 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 to determine whether the text was marked correctly or not.

During the experiment, classification accuracy 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 word frequencies with *idf* weight (tf-idf) – to show whether the quality of sample selection methods are not dependent on the text representation.

7 RESULTS

The following subsections demonstrate the results of the two experiments described above.

7.1 Experiment 1: One Class Classification

The texts from set T were sorted according to 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 sample sets were analyzed to

find out how the process of sample selection influences the similarity with a simple assumption that the higher is the quality of samples the higher is the similarity of sample and test data.

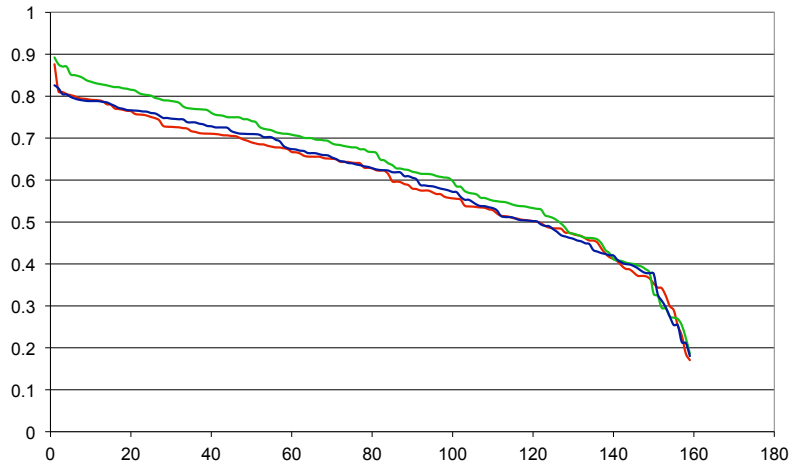


Figure 2. Comparison of differently created samples using the S_i^1 cosine similarity measure. Red – comparison to R, green – comparison to M, blue – comparison to B. Vertical axis – value of S_i^1, horizontal axis – document number. Data source 1 (book).

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 or showed other deficiencies. When the results of ranking by similarity are displayed in a graph the curve that lies above another curve represents a 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 S_i^1 similarity/distance measure the differences among differently selected samples were not very obvious (the curves were close to each other), see Figure 2. Therefore the total similarity/distance measure S_i was used to evaluate the methods of samples selection (see Figures 3 and 5).

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 ordinal numbers, i.e. texts most similar to the samples. The more to the beginning of the horizontal axis on 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.

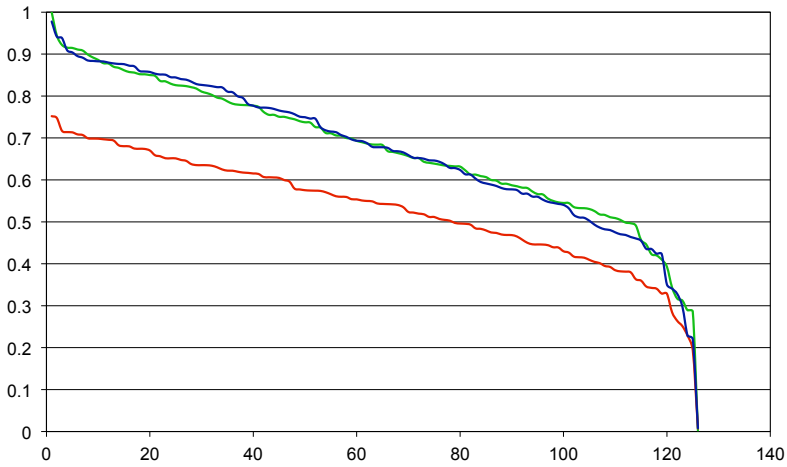


Figure 3. Comparison of differently created samples using the total cosine similarity measure S_i. Red – comparison to R, green – comparison to M, blue – comparison to B. Vertical axis – value of S_i^1, horizontal axis – document number. Data source 2 (hardware).

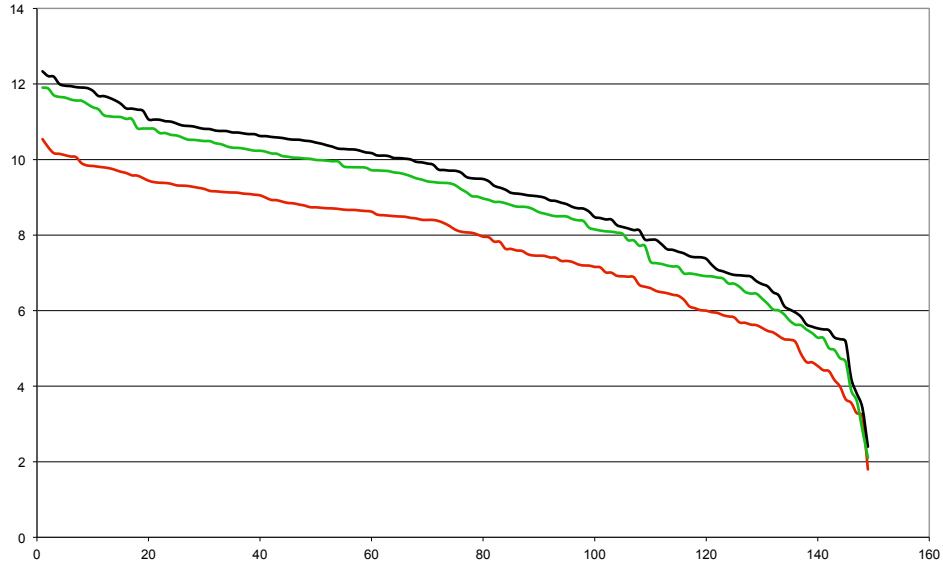


Figure 4. Comparison of differently created samples using the total cosine similarity measure S_i. Red – comparison to R, green – comparison to M, blue – comparison to B. Vertical axis – value of S_i^1, horizontal axis – document number. Data source 3 (movie).

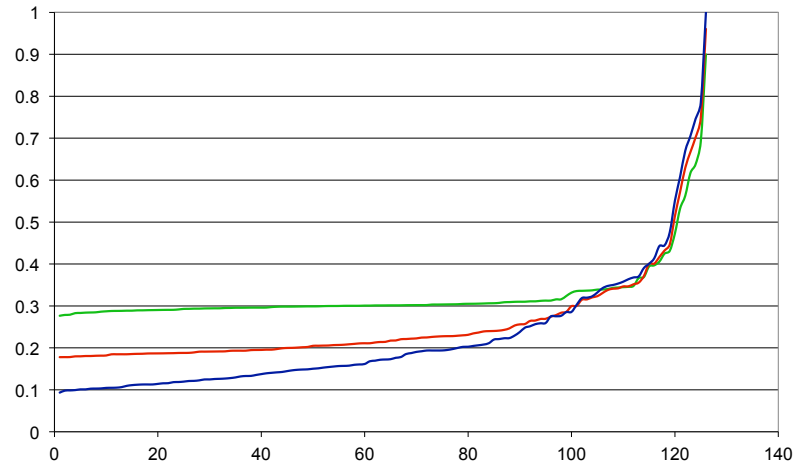


Figure 5. Comparison of differently created samples using the total Euclidean distance S_i. Red – comparison to R, green – comparison to M, blue – comparison to B. Vertical axis – value of S_i^1, horizontal axis – document number. Data source 2 (hardware).

When using the cosine similarity measure the results provided by comparison to R were the worst. Samples selected manually (M) and with a bias (B) provided very similar results. However, the effort of creating both sample sets was incomparable – automatic creation of the sample set could be done by the computer within 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 the cosine similarity measure. In the case of Euclidean distance similarity measure, almost one half of the texts from T were matched to B better than all documents to R. Because the texts that are in top N best matched documents are usually relevant (N is typically a relatively small number representing a number of documents that a user is able to utilize), these findings provide a good potential for future research.

7.2 Experiment 2: Two Class Classification

The following tables (Tables 2–7) show the performance measures of classification of testing data into classes based on comparisons to sample sets created in three different ways mentioned above. Each table contains the values of selected performance metrics for experiments with different pairs of review categories and for different vector representations of the texts – term frequency (TF) and tf-idf. Column *Acc*

contains the Accuracy, $T(x)$ and $F(x)$ represent the percentage of documents from class x that were classified correctly (True) and incorrectly (False), and $Prec(x)$ and $Rec(x)$ represent aggregate metrics Precision and Recall for corresponding classes.

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	25.3	25.9	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	28.0	11.0	0.87	0.76	0.72	0.89

Table 2. Effectiveness evaluation of classification of Book (1) and Hardware (2) reviews, TF vector representation

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	13.8	24.6	0.78	0.85	0.86	0.75
M	0.88	90.0	85.0	10.0	15.0	0.86	0.90	0.90	0.85
B	0.83	79.0	87.0	21.0	13.0	0.86	0.81	0.79	0.87

Table 3. Effectiveness evaluation of classification of Book (1) and Movie (3) reviews, TF vector representation

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	7.6	23.1	0.80	0.91	0.92	0.77
M	0.87	92.0	82.0	8.0	18.0	0.84	0.91	0.92	0.82
B	0.84	92.0	76.0	8.0	24.0	0.79	0.91	0.92	0.76

Table 4. Effectiveness evaluation of classification of Hardware (2) and Movie (3) reviews, TF vector representation

In all cases (except one), when the sample documents were selected randomly from a larger set of potential samples, the results of the classification achieved the lowest Accuracy. Other measures usually had their values worse than in the cases when other sample selection methods were used as well. Both manual and automatic biased sample selection methods thus enabled to achieve better classification results. The method of vector representation of text entries (term frequencies or tf-idf) had expectedly an impact on classification performance metrics. Classification with tf-idf representation achieved better results in terms of classification metrics values by decreasing the relative importance of terms appearing in a high number of reviews. However, the values of performance measures of classification based on the three

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	13.2	12.5	0.89	0.88	0.87	0.88
M	0.93	98.0	88.0	2.0	12.0	0.89	0.98	0.98	0.88
B	0.93	93.0	92.0	7.0	8.0	0.92	0.93	0.93	0.92

Table 5. Effectiveness evaluation of classification of Book (1) and Hardware (2) reviews, tf-idf vector representation

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	6.3	19.8	0.84	0.93	0.94	0.80
M	0.94	95.0	93.0	5.0	7.0	0.93	0.95	0.95	0.93
B	0.94	91.0	96.0	9.0	4.0	0.96	0.91	0.91	0.96

Table 6. Effectiveness evaluation of classification of Book (1) and Movie (3) reviews, tf-idf vector representation

presented approaches to sample selection remained in the same relation – the random selection process was generally the worst.

8 DISCUSSION AND CONCLUSION

During the process of creating a set of representative sample documents that can be used for classification or for document ranking by similarity several procedures can be applied. In the paper, we examined three methods for selecting a set of sample documents from a set of potential samples – automated random selection, manual selection, and automatic biased selection. The methods were used in experiments processing real world data – customer reviews from `amazon.com`.

In the experiments where the sample sets were created using random selection the achieved results were associated with the worst values of selected performance measures. In the process of ranking by similarity, the similarity between sample and testing data was the lowest. During classification, the classification performance

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	4.8	8.9	0.92	0.95	0.95	0.91
M	0.98	97.0	98.0	3.0	2.0	0.98	0.97	0.97	0.98
B	0.98	97.0	98.0	3.0	2.0	0.98	0.97	0.97	0.98

Table 7. Effectiveness evaluation of classification of Hardware (2) and Movie (3) reviews, tf-idf vector representation

measures generally provided the worse values in terms of Accuracy, Precision, and Recall. Improved results were achieved when the sample sets were created through manual or automatic biased selection. We might therefore conclude that random selection is not a suitable method for the presented tasks. Both methods (manual and automatic biased selection) provided comparable results in terms of selected performance measures. However, several issues related to each of these two methods can be found.

During the process of manual selection, several difficulties that make the selection more difficult have been 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 an entire group of products (e.g., problems with data backup and recovery) or an issue more or less 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 of representative samples has also several other aspects. On one hand, the reviews that are off-topic or 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 hundreds 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.

The presented approach of automated biased selection thus provides an alternative approach to manual selection and evaluation of potential textual samples. The experiments showed that the measures of classifier quality for the presented classification were close to or better than those for the classification based on manual data preparation. Also the documents retrieved and filtered using the presented method based on ranking by similarity showed higher similarity.

The presented approach can be thus used during processing large amounts of documents and as a part of more sophisticated document processing procedures, such as filtering results of Internet search engines [3], in a meta search engine [19] or in text summarization [15].

Future research will focus on processing other languages. In order to obtain better results, certain language properties could be accepted as well, for example, eliminating meaningless words bringing no information to the process or improving the preprocessing phase using selected linguistic tools (removing stop-words, stemming).

Acknowledgements

This research work published in this paper was supported by the Research Program of Czech Ministry of Education VZ MSM 6215648904.

REFERENCES

[1] Broder, A.—Fontoura, M.—Josifovski, V.—Riedel, L.: A Semantic Approach to Contextual Advertising. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2007, pp. 559–566.

[2] Cambazoglu, B. B.—Aykanat, C.: Performance of Query Processing Implementations in Ranking-Based Text Retrieval Systems Using Inverted Indices. Information Processing & Management, Vol. 42, 2006, No. 4, pp. 875–898.

[3] Chau, M.—Chen, H.: A Machine Learning Approach to web Page Filtering Using Content and Structure Analysis. Decision Support Systems, Vol. 44, 2008, No. 2, pp. 482–494.

[4] Christen, P.—Goiser, K.: Quality and Complexity Measures for Data Linkage and Duplication. Studies in Computational Intelligence, Vol. 48, 2007, pp. 127–151.

[5] Clark, E.—Araki, K.: Text Normalization in Social Media: Progress, Problems and Applications for a Pre-Processing System of Casual English. Procedia – Social and Behavioral Sciences, Vol. 27, 2011, pp. 2–11.

[6] Dařena, F.: Global Architecture of Marketing Information Systems. Agricultural Economics, Vol. 52, 2007, No. 9, pp. 432–440.

[7] Duda, R. O.—Hart, P. E.—Stork, D. G.: Pattern Classification. John Wiley & Sons, New York, 2001.

[8] Duh, K.—Kirchhoff. K.: Semi-Supervised Ranking for Document Retrieval. Computer Speech & Language, Vol. 25, 2011, No. 2, pp. 261–281.

[9] Figueiredo, F.—Rocha, L.—Couto, T.—Salles, T.—Goncalves, M. A.—Meira, W.: Word Co-Occurrence Features for Text Classification. Information Systems, Vol. 36, 2011, pp. 843–858.

[10] Gu, Q.—Zhu, L.—Cai, Z.: Evaluation Measures of the Classification Performance of Imbalanced Data Sets. ISICA 2009, Vol. 51, 2009, pp. 461–471.

[11] Hroza, J.—Žižka, J.: 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, 2005. Mexico City: Springer, pp. 608–611.

[12] Joachims, T.: Learning to Classify Text Using Support Vector Machines. Norwell: Kluwer Academic Publishers, 2002.

[13] Laver, M.—Benoit, K.—Garry, J.: Extracting Policy Positions from Political Texts Using Words as Data. American Political Science Review, Vol. 97, 2003, pp. 311–331.

[14] Liu, B.: Web Data Mining; Exploring Hyperlinks, Contents, and Usage Data. Springer, Heidelberg 2006.

[15] Lloret, E.—Palomar, M.: Text Summarisation in Progress: A Literature Review. Artificial Intelligence Review, Vol. 37, 2012, No. 1, pp. 1–41.

[16] Mazhelis, O.: One-Class Classifiers: a Review and Analysis of Suitability in the Context of Mobile-Masquerader Detection. South African Computer Journal, Vol. 36, 2006, pp. 29–48.

[17] Nie, J. Y.: Cross-Language Information Retrieval. Synthesis Lectures on Human Language Technologies, Vol. 3, 2010, No. 1, pp. 1–125.

[18] Oyama, S.—Kokubo. T.—Ishida, T.: Domain-Specific Web Search with Keyword Spices. IEEE Transactions on Knowledge and Data Engineering, Vol. 16, 2004, No. 1, pp. 17–27.

[19] Radovanović, M.—Ivanović, M.—Budimac, Z.: Text Categorization and Sorting of Web Search Results. Computing and Informatics, Vol. 28, 2009, No. 6, pp. 861–893.

[20] Srivastava, A. N.—Sahami, M. (Eds.): Text Mining: Classification, Clustering, and Applications. Chapman & Hall/CRC, Boca Raton, FL, 2009.

[21] Žižka, J.—Dařena, F.: Automatic Sentiment Analysis Using the Textual Pattern Content Similarity in Natural Language. Lecture Notes in Artificial Intelligence, Vol. 6231, 2010, pp. 224–231.

[22] Žižka, J.—Dařena, F.: Mining Significant Words from Customer Opinions Written in Different Natural Languages. Lecture Notes in Artificial Intelligence, Vol. 6836, 2011, pp. 211–218.

[23] Žižka, J.—Svoboda, A.—Dařena, F.: Selecting Text Entries Using a Few Positive Samples and Similarity Ranking. Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis, Vol. LIX, 2010, No. 4, pp. 399–408.

František Dařena works as an Associate Professor and Head of Information Systems working group at the Department of Informatics of the Faculty of Business and Economics, Mendel University in Brno, Czech Republic; he is also a member of SoNet research center. Formerly, he was an analyst and developer in a team developing one of the most successful information systems for universities in Europe. His research and pedagogical work is concerned with the utilization of information systems and technologies for supporting business activities, intelligent data processing, text mining, and sentiment and opinion analysis. He is also the author of several publications in international scientific journals, conference proceedings, and monographs, a member of program committees of several international scientific conferences, a member of editorial boards of international journals and Editor-in-Chief of International Journal in Foundations of Computer Science & Technology.

Jan Žižka works as an Associate Professor at the Mendel University in Brno, Czech Republic. He is engaged in research and teaching of advanced artificial intelligence, machine learning, soft computing, and data and text mining. A member of AAAI, Editor-in-Chief of International Journal of Computer Science & Information Technology, and International Journal of Information Sciences and Techniques, a member of editorial boards and program committees of several other international scientific journals and conferences. He is also the author and co-author of many journal and conference peer-reviewed articles, and co-editor of several books in the area of informatics. His current research interests are in the sentiment/opinion analysis area using large real-world textual data in different natural languages as well as in applying data-mining to various areas including business intelligence. Jan Žižka worked at universities and research centers in several countries and is keeping on the international cooperation inside and outside the European Union.