ACTA UNIVERSITATIS AGRICULTURAE ET SILVICULTURAE MENDELIANAE BRUNENSIS

Volume LIX 47 Number 4, 2011

SELECTING TEXT ENTRIES USING A FEW POSITIVE SAMPLES AND SIMILARITY RANKING

J. Žižka, A. Svoboda, F. Dařena

Received: February 25, 2011

Abstract

ŽIŽKA, J., SVOBODA, A., DAŘENA, F.: *Selecting text entries using a few positive samples and similarity ranking.* Acta univ. agric. et silvic. Mendel. Brun., 2011, LIX, No. 4, pp. 399–408

This research was inspired by procedures that are used by human bibliographic searchers: Given some textual and only 'positive' (relevant, interesting) examples coming just from one category, find promptly and simply in an available collection of various unlabeled documents the most similar ones that belong to a relevant topic defined by an applicant. The problem of the categorization of unlabeled relevant and irrelevant textual documents is here solved by using a small subset of relevant available patterns labeled manually in advance. Unlabeled text items are compared with such labeled patterns. The unlabeled samples are then ranked according their degree of similarity with the patterns. At the top of the rank, there are the most similar (relevant) items. Entries receding from the rank top represent gradually less and less similar entries. The authors emphasize that this simple method, aimed at processing large volumes of text entries, provides initial filtering results from the accuracy point of view and the users can avoid the demanding task of labeling too many training examples to be able to apply a chosen classifier, and at the same time, they can obtain quickly the relevant items. The ranking-based approach gives results that can be possibly further used for the following text-item processing where the number of irrelevant items is already not so high as at the beginning. Even if this relatively simple automatic search is not errorless due to the overlapping of documents, it can help process particularly very large unstructured textual data volumes.

unlabeled text documents, one-class categorization, text similarity, ranking by similarity, pattern recognition, machine learning, natural language processing, non-semantic documents

Typically, today's Internet users, and not only they, have very often to fight a battle against enormous numbers of textual documents or messages – written in any free common natural language – provided as the raw result by standard search engines, email servers, browsers, database answers, and so like. One of typical tasks is to select R positive (that is, interesting or relevant) news, documents, or messages from a very large collection of n unlabeled items that have no substantial structure which could support the necessary filtering. This is still one of actual problems, having various particular solutions, see for example Quan, Fang, Xiaoguang (2009) and Wu *et al.* (2009). In this case, standard and proved classification methods, based on the inductive supervised machine learning from labeled examples (Bishop, 2006; Hastie, Tibshirani, Friedman, 2009; Sebastiani, 2002; Srivastava, Sahami, 2009) cannot be applied, even if it expectedly and intuitively would return the best satisfactory results as, for example, the 'naïve' Bayes method typically applied to filtering spam. Preparing large volumes of training examples from thousands or tens of thousands (or even far more) examples by their 'manual' categorization into relevant and irrelevant class samples takes inevitably a very long time and considerable effort for authorized people, let alone high financial expenses. Another possible way is to employ one of clustering algorithms (unsupervised machine learning), however, the results are often not very satisfactory due to the very high dimensionality and occurrence of irrelevant attributes (words, terms) playing the role of noise (Hu *et al.*, 2009; Srivastava, Sahami, 2009).

This article proposes an alternative, relatively simple approach for those situations when it is desirable to find relevant text items from a very large collection of all kinds of unstructured natural-language textual entries without the primary high *accuracy, A,* requirement due to the big data-volume:

$$A = (TP + TN) / (TP + TN + FP + FN), \quad (1)$$

where *TP* stands for the number of *true positive*, *TN* for *true negative*, *FP* for *false positive*, and *FN* for *false negative* selected items; in other words, the number of correctly accepted relevant plus the number of correctly rejected irrelevant items to the number of all accepted and rejected ones, correctly and incorrectly (Sebastiani, 2002). Similarly, the *inaccuracy, E,* represents the error of document selection:

$$E = 1.0 - A. \quad (2)$$

When there are too many possible relevant entries, a user typically cannot process and utilize all the relevant available entries. This is quite typical when searching for textual documents that deal with just one specific topic: A user cannot make use of hundreds or thousands documents despite the fact that all of them could be relevant. Instead, such a user settles for a 'reasonable' number of relevant entries (typically tens or so). When dealing with a not too high number of such entries, the user can also tolerate a low number of wrong items because he or she can easily discard them 'manually', without fully relying on an automatic process. After such initial filtering, additional methods aimed at increasing the basic accuracy *A* can be later applied, if necessary, including supervised or semi-supervised learning (Abney, 2008).

Our suggested procedure works in the following way:

- Firstly, we can 'manually' label just a few – units or tens – carefully chosen patterns (sometimes also called as models) of good, typical positive examples.
- Secondly, using these patterns as the basis for the automatic selection of only positive items that are similar to the predefined (labeled) patterns, a user can avoid that demanding work of collecting and labeling too many training samples.

The chosen and labeled patterns actually bring more initial information into the process of the pattern recognition, therefore a user can expect better results than applying only common clustering algorithms (Abney, 2008). Expectedly, the standard classification approach uses much more initial information, so generally the filtering results can bring better outputs. However, one of the goals here was to overcame the expensive data-preprocessing phase caused by the inevitable 'manual' labeling hundreds, thousands, or even far more training samples. Naturally, three principal questions remain to be answered here:

- How to determine the *degree of similarity* of unlabeled items to predefined patterns?
- How *many patterns* are necessary?
- What *accuracy* of such the filtering process can users expect?

Such a process of textual entry retrieval, based on similarity to predefined patterns, can be applied to many opportunities, beginning with looking for similar topics or opinions in social networks and ending in sophisticated processing of customers' opinion data in business intelligence, and so like (Shmueli, Patel, Bruce, 2007; Žižka, Dařena, 2010). In the following sections, the paper attempts to provide answers and demonstrate experimental results with some publicly available real-world data.

Similarity of Text Items

Thematic and content similarity of text items is given by their mutual resemblance. In natural languages, it is mainly, but not only, semantic of the items that suggests a certain similarity between the items. Individual words, sentences, and paragraphs, their contextual meaning and reciprocal relationships transmit understandable information hidden in natural language elements connected together by specific grammar rules. This is how human beings use it. Unfortunately, machines cannot use it in the same way because of their quite different reasoning principles.

Text Representation

Various more-or-less successful approaches have been tried and applied to the problem of natural language processing. In this article, the chosen approach comes from the effective and principally simple method called *bag-of-words*, BoW (Sebastiani, 2002), which is an unordered collection of words, disregarding word order and preserving no original linguistic, grammatical, or semantic relations. This evident loss of information is to a certain degree replaced by 'quantity', which means that a machine has to use the number of training patterns higher by several orders of magnitude than a human being to be able to 'understand' the meaning of text items.

Each word represents a coordinate on its dimension (axis). Words in text documents are represented mainly (but not only) in these common ways: either a word is (1) or is not (0) in a text-item – binary representation; or by its frequency (playing the role of a word-weight); or using the TF × IDF value (term frequency times inverse document frequency). Then, a whole text-item is considered to be a multidimensional point (or a vector) in an abstract space with coordinates given by the word representation. The BoW-based methods employ instances of just individual words (or sometimes pre-defined phrases) without keeping their original position in a textual entry and relations between neighboring word sections. Such an approach significantly facilitates the large data processing, however, on the reverse of this coin is the inevitable

loss of information. Still, the huge number of applications shows that this method is quite acceptable, see also many references in (Sebastiani, 2002), if it brings expected good results from the practical or empirical point of view.

The basis of the procedure is created by dictionary that contains items defined as words which are in the available samples of textual entries. The dictionary of available *training* text items is represented by matrix $n \times m$, where n is a number of all documents (that is, a number of rows) and m is the number of unique words (that is, a number of columns) in the dictionary. Then, every matrix word element, w, is either a binary, integer, or real number depending on the applied representation of words. For example, the word occurrences can be simply expressed binary as yes/no (or 1/0), or numerically as individual word frequencies in documents, or even more ingeniously as so called TF × IDF values, see for example (Salton, Buckley, 1988): the word importance (that is, its weight) increases proportionally to the number of times a word appears in the document but is counterbalanced by the frequency of the word in the word-corpus. Similarly, the individual documents are represented by m-dimensional vectors that, for massive data volumes, are usually very sparse, with most of coordinates equal to zero because of many absent words that are in the joint dictionary.

Such the common techniques can be variously supplemented by other methods that can often be more or less specific for different languages, for example, excluding stop (common) words that do not bring any discriminating information (because they occur more or less evenly in all classes), transforming words to their stems (decreasing the very high dimensionality of the problem) if possible and necessary, and so like (Sebastiani, 2002). Each document is therefore treated as a vector and the mutual document similarity is then determined either as the vector similarity or it can be expressed as a certain distance between multidimensional points that represent individual textual entries.

It is worth to note that the high dimensionality is the cause of computational complexity. In addition, not only correct language words determine the number of dimensions that is usually thousands and more. Typically, the authors of textual entries use very often informal language (colloquial) terms, words distorted by grammar errors or mistypings, and so like. As a result, the number of dimensions increases and those incorrect words work as noise, making the data processing more difficult and increasing the possible inaccuracy. All such problems are generally well known.

Text Similarity

The similarity of text-item pairs can be measured as a distance, L, between the multidimensional points created by individual items – the closer the points appear, the more similar the text items are (Srivastava, Sahami, 2009). The distance computation depends on a specific situation, however, the simple (and in most cases used) 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 $w_{k,i}$ within the two documents being processed:

$$L_E = \sqrt{\sum_{i=1}^{m} (w_{j,i} - w_{k,i})^2}. \tag{3}$$

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):

$$L_C = \arccos \frac{\vec{d_j} \times \vec{d_k}}{|\vec{d_j}| \times |\vec{d_k}|}, \tag{4}$$

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.

Experiments, described in the following sections, used the frequency representation of words in their documents, and both the Euclidean and cosine distance between documents (to compare the both similarity measures). To avoid sometimes demanding data pre-processing that can be strongly dependent on a specific natural language and possibly very time consuming, no special pre-processing like, for example, Porter stemming (Porter, 1980) for English was performed. It is true that especially (but not only) for the English language, there exist various well developed pre-processing methods. However, many other languages are missing them, or due to belonging to a different language families (Romance, Slavonic, Finno-Ugric, and so like) such methods are rather complicated and often not so far sufficiently developed or available, even if a lot of people speak, read, write, or nowadays communicate via the Internet in different languages.

Similarity to *k*-Positive Patterns

The presented approach is somehow 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). Generally, its training phase is trivial – just storing labeled samples of individual classes. A new unlabeled item computes its distance to all labeled samples and then the $k \geq 1$ nearest patterns (neighbors) assign a respective label to that unknown item according to the most frequent category of its k-nearest neighbors. However, if there is actually only one known class (in our case, a small amount of positive samples), see, for example, (Manewitz, Yousef, 2001), the k-NN algorithm cannot be applied directly. A certain disadvantage of k-NN is that for big numbers of patterns its computational complexity can be very high. Given a set M of patterns, the running time is proportional to $O(kw)$, where k is the cardinality of M and w (the number of distinct words in the dictionary) is

the dimensionality of *M*. However, the method described here supposes using a relatively low number *k* of patterns, just units or tens (not more), because users wish to avoid labeling too many patterns and waiting too long for the results. On the other hand, *w* is usually very high, thousands or tens of thousands, even more. In addition, the common search engines (for example, *Google*) have a difficult work to select desired items using key words – their returned result can be ordinarily millions of answers related to given key words, and each answer must be compared with the given *M* patterns via computing its distance to each of them.

As it was mentioned above, the k-NN computational complexity *O(kw)* is negatively ruled by the large number of distinct words, *w*. Even if the research described in this article did not directly investigate the efficiency viewpoint, the computational complexity clearly showed itself in cases when *w* went beyond approximately 5000 and more. This is a typical problem in text mining. The parameter *w* can be lowered by eliminating words that do not contribute to class discrimination like stop words, or by stemming, for example, *be, am, are, is, were, was to be*, plurals to singurals, and so like, see (Porter, 1980) and (Sebastiani, 2002). Such methods are well developed for English, particularly from the available software point of view. The approach described in this article aims to broader generality concerning the language viewpoint; however, different language groups need different stemming tools. On the other hand, stemming also decreases the information contents, which could result in lower class discrimination accuracy.

Modern multi-processor computers with large memory can contribute to the computational complexity problem solution using the algorithm scalability, that is, algorithms which can utilize parallel processing and grid computing. This topic was, however, beyond the research objective of this article. Interested readers can find more in, for example, (Bogdanov, Singh, 2010), where the authors deal with the k-NN algorithm and its scalability.

Ranking by Similarity

Therefore, instead of the usual classification procedure, the unlabeled items can be ranked in compliance with their computed similarity to the available positive patterns, having just some labeled samples of only one class. Such a similar ranking-based approach has already been published and successfully applied to the European Commission's data *Europe Media Monitor NewsBrief*, see (Žižka *et al*., 2006). Because the text similarity to the predetermined patterns can be quite different – from very similar to very dissimilar items – and because the goal is to select the most similar items from large data volumes, the unlabeled processed items are sorted: The most similar at the top of the rank, and the least similar towards the bottom. Then, a user can expect the most relevant items among the first *r* ones near the rank top. It is up to a user's decision how many top-ranked items she or he selects or accepts. As the optimal result, all the *R* relevant text items should take the first *R* positions, while the rest places would be occupied by the remaining irrelevant (*R* – *n*) items.

Generally, users can expect some incorrectly ranked text items within the whole rank, including its top because machines are not perfect and very often it is not easy to define a crisp border between neighboring categories. However, it would be natural also in the case when even well-trained human beings would carry out the same task with a lot of text documents or messages. The important thing is to keep the errors near the top as low as possible because users are usually expected to accept just a small number of items from the top ones. The pragmatic reason is that users are able to process (read) only a limited quantity of documents, papers, or messages – reading papers is only one of supporting tools for most of professions. In addition, within the accepted text items, like papers or email messages from the top, there is often a good chance to find links or references to other items that were, maybe mistakenly, placed lower in the rank, behind the limit defined by a user.

DATA AND EXPERIMENTS

The experiments described below were using specific, publicly available data sets (from 20 Newsgroups and amazon.com) so that it would be possible to verify the results. In addition, the used data are commonly employed in research experiments because they represent the real-world textual entries very well.

Data and Their Preprocessing

The ranking method, described in this paper, was tested using the popular and publicly available data-sets known as 20Newsgroups, see (20Newsgroups, 2009) plus additional data downloaded from the customer opinion area provided freely by the amazon Internet shop for its customers, see (Amazon.com, 2010). The 20Newsgroups data sets contain 20 different topics and each topic has 1 000 contributions (there is a couple of exceptions when the number is slightly lower than 1 000 – the explanation is available at the download site). The amazon text data sets (customer reviews) were in the original form as the customers wrote them, with all mistypings, grammar errors, and so like. For the amazon data, we used as the topics simply the sold product names (that is, a specific book, hardware, and film).

Experiments were based on mutual comparisons of various pairs of different topics where one topic was defined as relevant and the other one represented the irrelevant items. Before starting the experiments, the 20Newsgroups data had to be preprocessed by removing all headers of individual newsgroups contributions because such headers

could represent certain kind of meta-data that are not common for all types of text documents, therefore the results could be distorted. The research aims at situations where the text entries are quite unstructured, without any higher-level description. The amazon data were cleaned from all tags, so that only the text body of the review remained. Then, all the data were pre-processed in the same way: all the HTML tags, numbers, punctuation, or special symbols were removed, thus only the regular words were left in the text entries. Any other special modifications, often used for the standard training of classifiers when more classes are available, were omitted (for example, no stop-word removing, no leaving out 'short words' having less than 3 letters, and so like). These modifications are usually dependent on a specific language and the described research wanted to avoid it. On the other hand, omitting the wider modifications usually provides worse results – however, it should be another research chapter.

So, the bag-of-words were prepared very simply. Then, the authors intentionally selected pairs of topics so that the ranking method could be applied both to similar as well as dissimilar newsgroups topics. The reason was that the tested ranking algorithm was expected to give better results for dissimilar pairs of topics than for the similar ones. However, even for the relatively more similar pairs, the separation was expected to work, too, probably and naturally with a larger error for the more lexically and topically similar newsgroups categories. The selected pairs were topically the following, using the names of the 20Newsgroups topics:

- Baseball versus Atheism,
- Baseball versus Hockey,
- Baseball versus Windows software,
- Foresale versus Guns (politics),
- Christian religion versus Autos,
- Macintosh hardware versus Guns (politics).

From the amazon data, the following pairs of topics were considered:

- Book versus Film,
- Hardware versus Film,
- Book versus Hardware.

Because of time, not all possible combinations have been tested; in addition, not all tested pairs are demonstrated here because many results were very similar. The mentioned pair Baseball vs. Hockey is topically similar (two sports) while the remaining items are more or less dissimilar. The number of words in each main dictionary for 20Newsgroups data was around 21–23 thousands for each pair, while for amazon data around one thousand. The individual contributions were relatively short and their word number was much lower than in the main dictionary, so the vectors representing the individual contributions were typically very sparse, containing mostly zeroes.

EXPERIMENTS

Words can be represented by different means. In this case, experiments used frequencies of individual words per each textual contribution. In each experiment, one topic from a pair was used as 'interesting, relevant' and the other as 'uninteresting, irrelevant'. Then, like a simulated user approach, 50 randomly selected basic samples for 20Newsgroups and 15 for amazon data were used as 'good, interesting' patterns. Here, a simulated user would want to get similar contributions from the remaining set.

The reason for selecting just relatively small number of patterns originates from a set of preliminary experiments the goal of which was to find a reasonably low but sufficiently high number of positive samples. These preliminary experiments showed that more than the small number of patterns did not improved the results while less patterns brought rather unstable results in repeated experiments with the random selection of the patterns.

After this data preparation, each unlabeled item from both interesting and uninteresting topic was compared with each of the interesting patterns: for each item, its Euclidean distance L_E to each of the basic patterns was computed. Altogether, these distances represented the similarity degree: the lower the sum of the distances is, the higher similarity exists, and vice versa. Here, the sum of distances represented the total similarity to the basic patterns. For each pair, the experiment was repeated ten times, each time with a new randomly selected set of basic patterns. In each round, the items were sorted according to their similarities with the basic patterns from the most to the least similar ones. The optimal result would contain the interesting items in the top and the remaining uninteresting items in the bottom of the rank.

As it could be expected, the results were not quite errorless. The top part of the ranked items included a certain number of uninteresting items (false positive) and some interesting items occurred in the bottom half of the rank (false negative). The error (or accuracy) is given by the number of incorrectly placed items: irrelevant in the upper part and relevant in the bottom one. The most important is to obtain the top of the rank as accurate as possible because a user is expected finally to accept only a relatively small proportion of the all similar items. And of course, the closer is an item placed to the boundary between the group of interesting and uninteresting items, the higher is the expected possibility of errors because it is often not quite obvious even for human beings where to place a not very crisply topically defined (according to the word contents) text item.

To verify the suggested ranking procedure, the authors investigated 20Newsgroups in details. The initial group of experiments was carried out using all words in contributions. Then, the supplementary

experiments used limited number of words: all words having their frequency ≤ 3 (for example, very rare words as some strange interjections generated by some newsgroups contributors) or ≥ 170 (for example, very common words like *a, an, the, be,* and so like) were excluded, so the size of the main dictionary was decreased to some 7 000 unique words. However, the results were very similar to ones of the unlimited case, so they are not shown here. The final result of each experiment was given as an average error depending on the position within the rank, see the following graphs in the next section.

To illustrate the behavior of the ranking procedure, the experiments employed also other real-world data sets. The authors decided to use the amazon data mentioned above because these data represented similar text entries (natural language, not very long items), however, they were from another area – not a newsgroup but customer opinions. The results were similar and readers can find them in the graphs.

RESULTS OF EXPERIMENTS

The graphs are showing how the error (inaccuracy) increased along the rank from top to bottom, or, in other words, when the user wanted to include more and more textual documents into the result. Because only part of textual entries was similar enough to randomly selected patterns, the increasing number of requested entries gradually increased the inaccuracy (false positives and negatives).

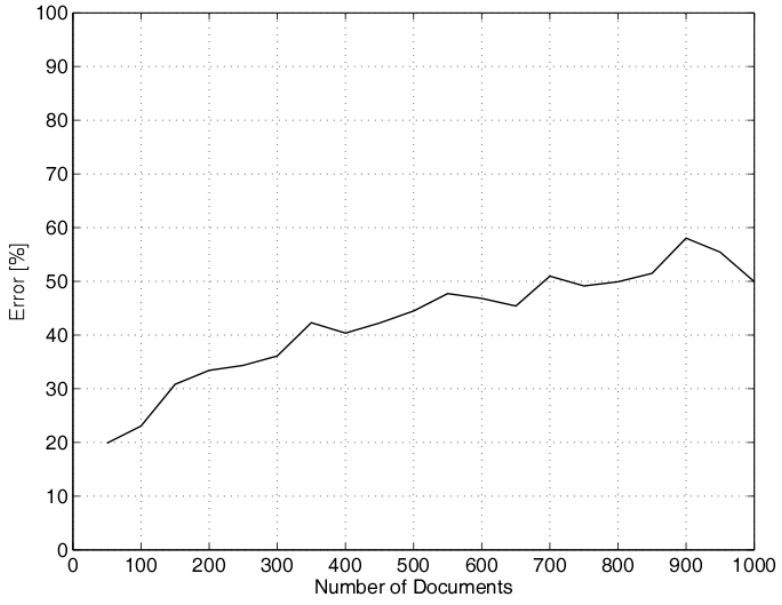
Ranking Algorithm

The following graphs demonstrate the results obtained by the experiments. Each graph depicts the error percentage, Error [%], depending on the position, Number of Documents, within the rank. Here, Number of Documents stands for the first *M* occurrences of items ranked according to their similarity degree to the basic patterns. For example, in Figure 1, for the first one hundred of items the error is some 23 %, and so like. At the end of a curve, the error was approximately 50 %. The curves represent the average errors taken from 10 experiments with randomly selected 50 basic patterns for the 20Newsgroups data.

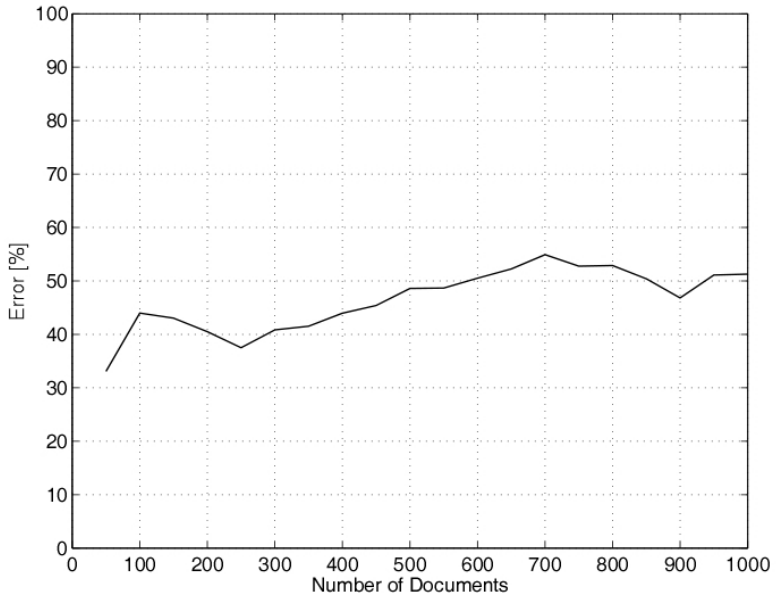
Figure 1 shows how the error increases for the topic Baseball (an interesting one) against Atheism (an example of uninteresting one). Near the top of the rank, that is, within the first few hundreds, the error is relatively low and the chance to get interesting items is higher. If a user would like to get more items, she or he can naturally expect more errors. Generally, at the first top positions, usually 10–25, there were only interesting items, so the lower absolute number of text samples the user wants, the more correct items are returned from a mixture of all samples. This approach is preferred, for example, by users which look for relevant papers and they do not need too much because they would not have time to study tens or hundreds of them.

Classification and Clustering

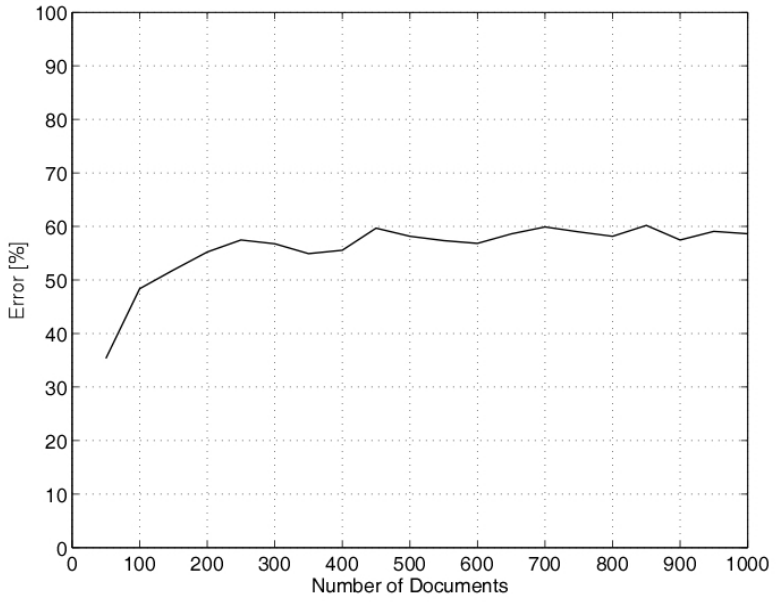
Classification and clustering was not the goal of the mentioned research. The ranking-algorithm error, even near the rank top, is relatively high, however, one should keep the conditions in mind:



1: *Baseball versus Atheism*



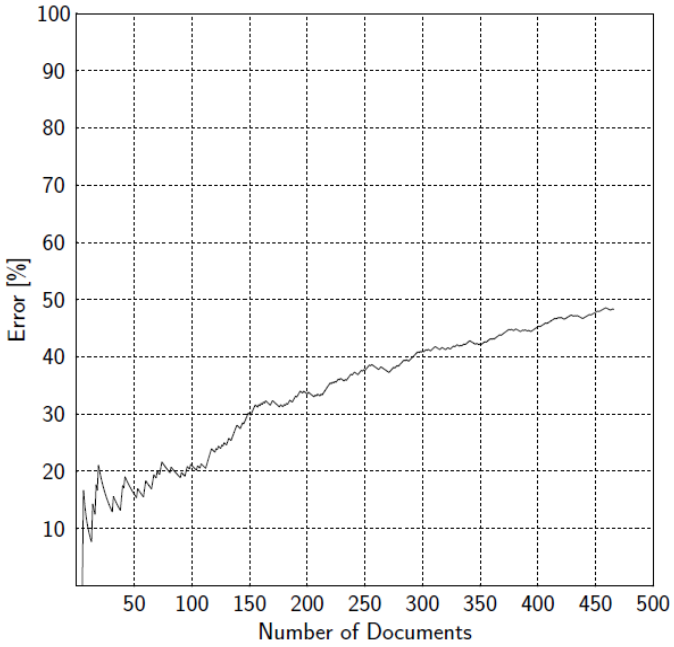
2: *Baseball versus Hockey*



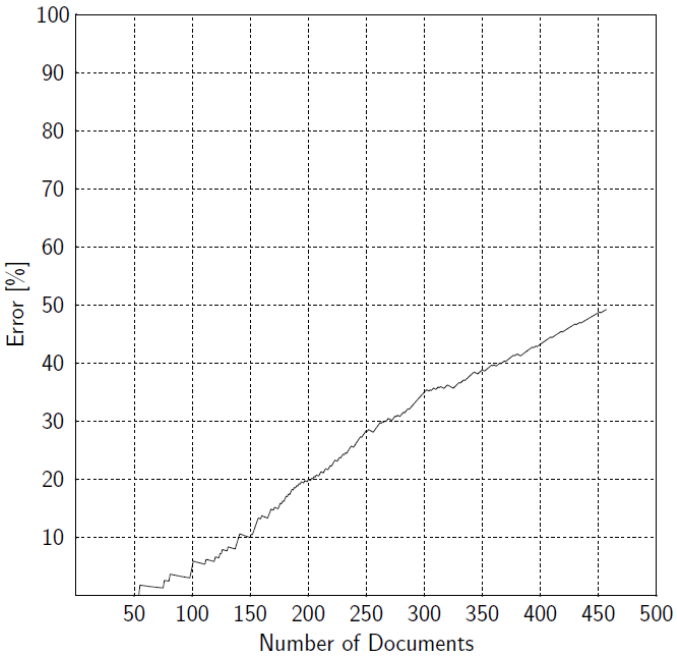
3: *Christian versus Autos*

no labels of examples are supposed to be known. Many various *classification* experiments with the 20Newsgroups data were published and some of the results were excellent. On the other hand, very simple experiments with the same data sets were carried out also as part of this research to compare results of using a small set of patterns (ranking) or no patterns (clustering). Clustering generally gave very bad results. For example, the simple k-means (for $k = 2$) algorithm generally put all items (except a negligible handful) into just one group, reaching almost 100% error. Therefore, the results are not described here.

Some authors applied special procedures to clustering textual documents (using non-numerical vectors and tables), see for example (Jo, Jo, 2008). The results were good, however, they needed a very special data processing that depended also on a specific language, therefore it is an open question how general such results can be for other languages.



4: *Book versus Hardware*



5: *Film versus Hardware*

CONCLUSIONS

The experiments showed that the described ranking-based method using the similarity given by the Euclidean distance between members of a relatively small subset of labeled patterns and unlabeled text-data samples provides acceptable results given the achieved accuracy. Such an approach is an alternative procedure for processing the *one-class filtering problem*. The main goal was to find a method that would be simple enough and produce acceptable results by finding only relevant results at the top of the rank constructed by sorting the unlabeled samples according their similarity to the very small set of labeled patterns.

One of possible applications is filtering results of Internet search engines, another one could work for looking for potential plagiarism as an auxiliary tool. Several potential users of the presented method suggested its application to selecting not too many interesting text documents on a given topic because such users can process only a few relevant items while such items usually contain references to other ones, which means that even unselected items are generally not lost.

Similarly, there is interest in processing large amounts of textual entries provided by on-line customers of Internet e-shops. The users can write their opinion concerning certain products or services, providing a useful feedback that should be processed and used in, for example, data mining as part of business intelligence.

The continuing research aims at testing the presented approach with data in various languages and topics. One of the multilingual areas is processing sets of many text documents in several languages, which leads to a very high-dimensional problem due to a lot of unique words. Results of the suggested rank-based algorithm can also be used for the following processing by other, more sophisticated algorithms that would not start from the large volume of unlabeled data.

Also, the modern area of natural language processing now includes a hot topic that deals with analyzing opinions obtained from social networks, sentiment analysis. Due to very large data volumes (a lot of textual entries produced by social-network members), it is often practically impossible to apply traditional machine-learning based classifications that employ labeled training samples. Therefore, alternative methods should be tested and developed.

Acknowledgements

This paper was supported by the Research program of the Czech Ministry of Education, No. MSM 6215648904.

REFERENCES

20Newsgroups, http://people.csail.mit.edu/jrennie/20Newsgroups/ [cit. November 2009]

Amazon.com, http: //www.amazon.com/, [cit. March 2010].

ABNEY, S., 2008: Semisupervised Learning for Computational Linguistics. Chapman & Hall/CRC. ISBN 978-1-58488-559-7.

BISHOP, C. M., 2006: Pattern Recognition and Machine Learning. Berlin: Springer. ISBN 0-387-31073-2.

BOGDANOV, P., SINGH, A. K., 2010: Scalable Nearest Neighbors with Guarantees in Large and Composite Networks. Technical report, September 2010, Department of Computer Science, University of California, Santa Barbara, CA. Available also at the URL https://www.cs.ucsb.edu/research/tech_reports/reports/2010-17.pdf [cit. March 2011].

DUDA, R. O., 2004: Pattern Classification. 2nd Edition. John Wiley and Sons. ISBN 0-471-70350-8.

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, 608–611.

HASTIE, T., TIBSHIRANI, R., FRIEDMAN, J., 2009: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd Edition. Berlin: Springer. ISBN 0-387-84857-0.

HU, X., ZHANG, X., LU, C., PARK, E. K., ZHOU, X., 2009: Exploiting Wikipedia as external knowledge for document clustering. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. Paris: ACM, 389–396.

JO, T., JO, G. S., 2008: Table Based Single Pass Algorithm for Clustering Electronic Documents in 20NewsGroups. IWSCA-2008 IEEE International Workshop on Semantic Computing and Applications, 66–71.

MANEWITZ, L. R., YOUSEF, M., 2001: One-Class SVMs for Document Classification. Journal of Machine Learning Research, 2: 139–154. ISSN 1533-7928.

PORTER, M. F., 1980. An Algorithm for Suffix Stripping. Program 14, 3: 130–137.

QUAN, H., FANG, X., XIAOGUANG, L., 2009: A Comparative Study on Feature Window Selection in Text Filtering. International Forum on Information Technology and Applications, 3: pp. 209–212.

SALTON, G., BUCKLEY, C., 1988: Term-weighting approaches in automatic text retrieval. Information Processing & Management, 24, 5: 513–523. ISSN 0306–4573.

SEBASTIANI, F., 2002: Machine Learning in Automated Text categorization. ACM Computing Surveys, 34, 1: 1–47. ISSN 0360-0300.

SHMUELI, G., PATEL, N. R., BRUCE, P. C., 2007: Data Mining for Business Intelligence. John Wiley and Sons. ISBN 0-470-08485-5.

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

WU, Y., KUN, S., ZHU, W., YUE, X., LUO, H., 2009: A Web Text Filter Based on Rough Set Weighted Bayesian. Dependable, Autonomic and Secure Computing. Chengdu: IEEE, 241–245.

ŽIŽKA, J., DAŘENA, F., 2010: Automatic Sentiment Analysis Using the Textual Pattern Content Similarity in Natural Language. Lecture Notes in Artificial Intelligence, 6231, 1: 224–231. ISSN 0302-9743.

ŽIZKA, J., HROZA, J., POULIQUEN, B., IGNAT, C., STEINBERGER, R., 2006: The selection of Electronic Text Documents Supported by Only Positive Examples. In: JADT-2006, Proceedings of the Eight International Conference on the Statistical Analysis of Textual Data. Besanon, Presses Universitaires de Franche-Comte, 1001–1010.

Address

doc. Ing. Jan Žižka, CSc., Ing. František Dařena, Ph.D., Ústav informatiky, Mendelova univerzita v Brně, Zemědělská 1, 613 00 Brno, Ing. Arnošt Svoboda, Katedra aplikované matematiky a informatiky, Ekonomicko-správní fakulta, Masarykova univerzita, Lipová 41a, 602 00 Brno, e-mail: jan.zizka@mendelu.cz, frantisek.darena@mendelu.cz, arnost@econ.muni.cz