Revealing Potential Changes of Significant Terms in Streams of Textual Data Written in Natural Languages Using Windowing and Text Mining

Jan Žižka, František Dařena
Department of Informatics, FBE, Mendel University in Brno
Brno, Czech Republic
zizka, darena@mendelu.cz

***Abstract*—The presented research deals with analyzing continuous streams of textual data written in natural languages. One of problems is revealing possible significant concept changes in Internet blogs, discussions, etc., together with discovering what represents such data, if it is more-or-less topically invariable or changing, and what kind of change occurred. A real-world textual dataset is analyzed using text-mining with automatically generated decision trees to find significant words that affect correct assignment of document labels (classes) and can be used for detecting noticeable changes. The changes and their detection are here modeled by assorted gradual mixture of two languages and the change degree is measured by cosine, Eucledian, and Jaccard distance (similarity), which provide qualitatively the same result. The monitoring procedure is based on analyzing successively adjacent couples of data-windows in the stream using the comparison of the current and its previous window, both represented by their lists of relevant features expressed in words. The presented results demonstrate that the suggested method provides reliable results.**

I. Introduction

Currently, the Internet provides once unimaginable quantity of data including incessantly growing volumes of textual ones. Quite naturally, it is expected that with the growing amount of data the volume of information – hidden in that data – grows as well. Regardless of the constantly increasing possibilities of hardware and software tools, data volumes are growing too fast. Consequently, data mining as well as text mining constitutes a very topical problem. To reveal knowledge hidden in data, modern and mature machine-learning (ML) algorithms are often employed [1], [2]. However, most of today's ML-based tools can process data only as a batch, which needs very large, sometimes insufficient computer memory [3], [4].

When collecting data during longer time periods, its volume starts soon to be too large for batch processing independently on available RAM (or even virtual) memory. In addition, the computational complexity of data-mining algorithms grows usually non-linearly with the increasing number of data items and attributes (variables) describing those items. The main question was how to automatically reveal possible knowledge (here: concept) changes in continually incoming textual documents from the Internet. Experiments with real-world textual data coming from reviews of hotel services, available for instance at websites like *booking.com* [5], showed that for reviews written in English the training time of a decision tree *c5* [6] reached soon almost 500 CPU hours (i. e., three weeks) for just 160,000 reviews while the whole available review set included ca two million items. The starting set size had 2,000 items and gradually additional 2,000 items were added, enlarging the *window* through which more and more data items were available as demonstrated in [7]. The considerably growing elapsed CPU time for the increasing number of training samples is depicted in Fig. 1. The CPU time includes also testing, which is, however, quite negligible. That research was aimed at investigating what number of training samples could be useful and processable for revealing conceptually significant words in relation to a given class (positive and negative reviews). To get around the problem how, for example, to train a classifier when the number of training samples exceeds the hardware or software abilities, there are several ways as processing random selections, parallel processing, and so like, which can provide acceptable results [8].

Such kind of batch-based knowledge discovery may be worth considering when the goal is to discover knowledge in the whole collection of data items, which can be later used for classification or prediction. However, another task is when the goal is to reveal whether and how the knowledge changes during progressive incoming of newer and newer items. From looking for changes within the item-sequence viewpoint, which is the topic of this paper, such a task is frequent, for example, when analyzing various Internet blogs, discussion groups, sentiment expression, and so like, where significant features typically are moving, fading away, or new ones are arising.

Discovering changes in data streams is an idea that has been investigated many times from various viewpoints. A new and very good overview can be found, for instance, in [9]. In a well arranged way, this work summarizes the main possible solutions of the problem related to so called concept drift, which primarily refers to an online supervised learning scenario when the relation between the input data and the target variable changes over time.

This paper deals with a problem related to monitoring long sequences of textual data in order to discover potential changes in significant attributes (relevant features, here words) characterizing conceptually the data items (textual documents) from the generalized point of view – that is, a certain kind of knowledge hidden in investigated data. That knowledge is revealed with the help of a trained decision tree that can

ISBN 978-952-68397-0-7 (paperback), ISBN 978-952-68397-1-4 (PDF)

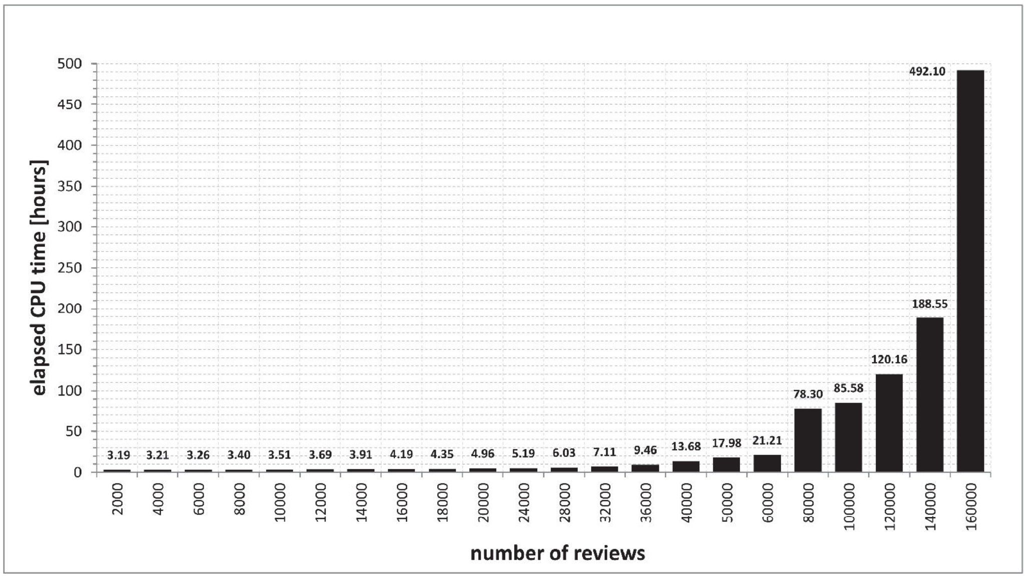

Fig. 1. The demonstration of fast non-linearly growing elapsed CPU time for text-mining based on training decision trees. The analysis aimed at discovering significant words using a whole batch of textual reviews written in English by customers of hotel services

select relevant features – the tree role is not here to classify; the classification is only an auxiliary procedure. From the classification point of view, continuously accumulated data can be analyzed incrementally by continuous update or by retraining a classifier (or a set of classifiers when using boosting or parallel processing) but the computer memory limit is crucial. If, given certain initial classes (which might be, for example, topics), the goal is to watch possible changes of attributes that are determinative for belonging to a class, the situation is different and considerable attribute changes can even lead to changes of topics. Such a case is also typical for data incoming as streams, which results in analyzing them in a single – not multiple – pass, considering changes in concepts as well as not so accurate solution like in the case of training a classifier with a batch of input data.

In this research, the described problem was simulated by replacement of words in input textual documents. To verify whether a suggested method based on discovering relevant features and their changes would be able to indicate the word modifications, the analysis used various mixture of documents in two different languages that described the same classes. Such an analysis can be primarily useful for data changing during time (some series of textual documents as Internet blogs) but at the time of investigation of this problem, no real-world data large enough were available.

II. Decision trees as text-mining tools

The reason for selecting a decision tree algorithm as a basic text-miner for discovering significant words was that such trees can be easily used for finding relevant attributes in relation to given classes – that is, for revealing significant words representing each class. The popular *c5* algorithm looks for attributes that divide a set containing members of several classes between subsets that have much more (or only) members belonging to just one class. In this case, a set ”purity” is measured by *entropy* (chaos) , $H(X)$, recursively computed using a well known formula $H(X) = -\sum_{i=1}^{n} p(x_i) \cdot log_2\, p(x_i)$, where X is a discrete random variable with possible values $x_1, x_2, \ldots, x_n$, and $p(x_i)$ is the probability of selecting a value x_i from a certain class. If a set contains members from only one class, its entropy value is 0.0 (minimum), while when all classes are represented in a given set equally, the entropy is 1.0 (maximum) when using log_2 [10].

The entropy value determines *information gain* of each attribute (here a word). The source set is divided between subsets so that the average entropy of the generated subsets would be minimized. The algorithm stops when no additional splitting lowers the entropy. Because of getting at least minimal generalization, a tree leaf must contain two or more items. For the generalization reason, *post-pruning* is then applied; more details may be found in [10]. As a result, the generated tree contains nodes determining a branch to a leaf. The nodes ask for values of attributes – the most significant attribute is in the root because it is tested every time, while other attributes gradually play less significant role as they are closer to leaves. A branch is actually a rule representing a certain combination of attributes on the left side; the right side is the label (a class name).

Using the *c5* decision-tree algorithm is not the only possible way. The problem with data streams and big data has been investigated from several points of view with employing various algorithms. One possibility is applying *Hoeffding trees*, which is an incremental decision tree induction algorithm based on a precondition that the distribution generating examples does not change over time. Hoeffding trees exploit the fact that – using Hoeffding bound – a small sample can often be enough to choose an optimal splitting attribute. Promising applications of Hoeffding trees were published in, for instance, [11]. Additionally, in [12], the authors researched and compared three algorithms (multinomial naïve Bayes, stochastic gradient descent, and Hoeffding trees) applied to a very large Twitter streaming data (micro-blogs), using a sliding window Kappa statistic for evaluation in time-changing data streams.

III. Investigated data

The textual data mentioned in the *Introduction* section – reviews of the hotel-service quality – has been investigated from different viewpoints, see for example, [13], [14], [15]. It is a collection of reviews in many languages (more than 30) where English data prevails but some other ”big” languages (Spanish, French, German, Russian, and others) are naturally represented as well, depending on a number of corresponding travelers. Often, some passengers did not use their native language (”small” languages as Czech, Hungarian, and so like) and wrote their opinions in English, or sometimes in two languages simultaneously in the same review, which was then assigned to the English group; for example, verbatim quotations for Polish and English: *W oliwkowym gaju nieduża hacjenda z dopracowanymi dodatkami dała nam wyciszenie, nasłonecznienie i pyszne jedzenie. In the olive grove the small hacienda with supplements touched up gave us the calm, the solar exposure and the delicious food.*, or Czech and English (with gross grammatical errors): *Co se mi nelíbylo byl hluk z ulice. There was litlle bit noise from street.*

The reason for using English is that it is today a kind of international language and, for instance, a hotel staff in Argentina or Japan only hardly would understand reviews in Czech. Here, only English and Spanish reviews were used. A customer of a hotel service is – after using that service – allowed to freely write down her or his opinion via the Internet. In all languages, the reviews were written sometimes with grammatical errors, unusual interjections, and so like. Without spell-check (which is not generally used for such kind of data), wrongly written words artificially extend dictionaries and play a role of noise, increasing also computational complexity.

For the described investigation, 500,000 reviews (250,000 positive and 250,000 negative) were randomly selected for English and the same number for Spanish in order to have two languages balanced 1:1 from the number of examples point of view. In the original datasets, there were fewer Spanish reviews than the English ones, approximately in the ratio 1:2 (the total English data contained over 1.2 million of positive and almost 750 thousands of negative reviews). Unbalanced classes need additional, not always easy, and sometimes rather artificial preprocessing [16]. Thus, altogether 1,000,000 reviews in two languages were analyzed, although not at once (see the section *Design of experiments* below).

Previous investigations showed that reviews in different languages contained almost the same words expressing positive or negative meanings (like *location, staff friendliness, food, wi-fi, noise*, and so on) [17]. As for this million of analyzed reviews, here is a simple statistics: The arithmetic average number of words per review was 23 (with standard deviation $\sigma = 21.63$), median value = 16; the shortest ones had just one word while the longest one contained 231 words.

Just for a brief illustration, here are some randomly chosen English examples of positive and negative reviews. They are, however, quite typical also for Spanish (or any other language). The content is original and may occasionally include various grammatical errors:

Positive examples:

- I only stayed 1 night as a course I attended was held there.
- The pool was a bonus after driving for 5.30 hours.
- Very easy to find, room small but comfortable and not too much.
- A stone-throw distance from the city centre, amazing authentic building and atmosphere, nice breakfast.
- Comfortable, clean, staff helpful & polite.

Negative examples:

- The restaurant and the breakfast was poor. It would not be so difcult to make the restaurant more plesant and serve more than one sort of bread.
- There were ants in the cantine with fryed eggs.
- I always miss youghurt and cheese.
- The beds were very har and there were no mattress availebel at the hotel. I had to fetch one at the pool-side and put in the bed , so that I at least could sleep a little. Bed was not large enough for two reasonable sized people.
- Carpet wanted renewing. No mirror in bedroom for make up and water leaked in bathroom after having a shower, (Not caused by having curtain outside bath).
- Freezing cold shower!

A. Data preprocessing

To process the textual data, certain standard and proved textual-data preprocessing steps had to be applied: all the data were transformed to lower-case, then all special characters and numbers were removed, so only the terms (words) having alphabetic characters remained [18]. The number of unique terms in a document-space defines the space dimensionality. For the decision tree generator *c5*, the upper bound of the computational complexity estimation, $O[f(m, n)]$, is $m \cdot n^2$, where *m* is the number of training samples (here rows in the matrix) and *n* is the number of words in the dictionary of reviews, therefore the computational time depends linearly on the number of training samples and quadratically on the number of words; for details see [19].

B. Data representation

The described data mining process cannot use the words as strings, so it is necessary to select a suitable numerical representation. One possibility is to represent a word by its frequency in a document it belongs to. If such documents have mutually different sizes (that is, the numbers of terms), an alternative frequency-based representation known as *tf-idf*, or *term frequency times inverted document frequency*, $tf\text{-}idf = t_f \cdot log(N_{doc}/N_{tdoc})$, is used, where t_f is the frequency of a term in a document, N_{doc} is the number of all documents, and N_{tdoc} is the number of documents containing the term t. This is actually a weighted frequency, and it was used in this research. Thus, each document was represented as a vector with coordinates given by the weighted frequencies of words, *tf-idf* [18]. Such vectors – as in this case – are usually very sparse because a document contains only a very small fraction of the dictionary defined by the whole document set, so the coordinates mostly equal zero.

The figure Fig. 3 illustrates one of the typical generated trees for English words. This tree size was 149 nodes and its classification accuracy error related to two evaluating labels (positive ”+” and negative ”−” represented by squared nodes) was 8.9%. The dashed lines point to next subtrees, which due to the limited page space cannot be shown. In the tree root, there is the most important word *location* followed by *no* and *friendly* on the next level, and so on. Eliptical nodes represent questions to words' weighted frequencies *tf-idf*. The corresponding window's dictionary contained 3925 words but only 60 played the role in the labeling process as relevant attributes. Because of the very sparse vectors, trees mostly asked if *tf-idf* > 0.0 or *tf-idf* $= 0.0$, which indicates that the problem is principally binary: *Is a word in a review or not?* Here, the relevant words, according to their significance, were *location* (100%), *friendly* (83%), *excellent* (79%), *beautiful* (72%), ..., *room* (1%). Following the individual branches to their leaves, significant word conjunctions may be discovered; for example, if a branch (that is, a review) contains terms *location* and *friendly* and *not* and *clean* and *helpful*, the review is positive. Such an evaluation is also understandable from the human point of view.

IV. Design of experiments

The first problem was how to detect changes in the sequence of reviews. Because of required generalization (to obtain knowledge from concrete individual review samples), it was necessary to choose a certain subsequence of samples having sufficient information to be generalized. The quality of generalization was measured using the accuracy of tested classification of a trained classifier, which was the above mentioned decision tree generator *c5*. Good generalization means a low classification error on a testing dataset. As a baseline, a decision tree trained with 50,000 samples was chosen because various experiments showed that such a data volume provided a reliable estimate of the classification error, which was under 10%, and the following increase of the number of training samples did not significantly improve that error [7], [17].

Then, a constant subsequence size = 10,000 training samples was found to be large enough to provide a close accuracy (the error was ca 12–14% for a testing dataset). Such a number of training samples defined the *window size*, that is, the subsequence size. All the windows kept the original order of reviews given by their source language. Having 1,000,000 review samples, the total number of windows was therefore $1000000/10000 = 100$.

To define a window size is not quite an easy task because it depends on a specific data and situation, including the monitored concepts. Several ways were described and tested, for example, in the *Flora* system [21]. However, the various published methods were aimed at different data type and did not provide satisfactory results for the investigated textual data described here. After some experiments, the mentioned size 10,000 reviews demonstrated an adequate size from the information contents point of view – that is, a possibility to be generalized with acceptable low accuracy error. In addition, this size was also acceptable from the computational complexity standpoint.

In order to find out whether the proposed method is able to discover the concept changes of documents' content, a data set with known content needed to be prepared. Under real circumstances, the documents would contain some topics and/or sentiment related to those topics. From a certain moment, the topics or opinions might change. For example, a renovation of hotel facilities can terminate complaints about some problems that were topical before such an event. A natural disaster might also significantly change the topics of interest of people involved in an on-line discussion regarding a particular destination.

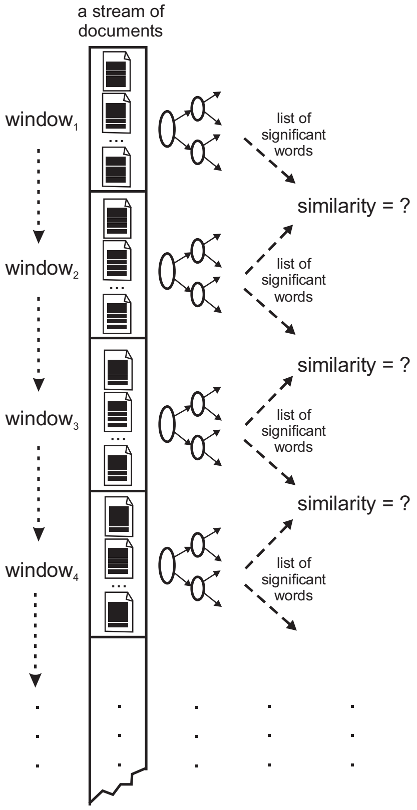

Fig. 2. The illustration of a data-stream divided between windows, which are used as sources of significant words related to a concept

No matter what kind of change happens, different topics manifest themselves in different vocabulary used to express the messages. Because it was not possible to indisputably determine the topic distribution in every single processed window and thus quantify the changes in concept content to have a baseline for subsequent analysis, the changes were simulated artificially by mixing documents written in two different languages.

Changes caused by occurrences of new significant terms (or their fading) in the stream of reviews were detected using lists of relevant words generated by decision trees trained for each window. The first 10 windows contained only English reviews, representing a practically invariable state. Then, suddenly, as a jump shift, 10 consecutive Spanish windows appeared. That very strong change was repeated again by using 10 English windows. Subsequently, the sudden changes were replaced by slower changes when the English reviews were gradually replaced with the Spanish ones and vice-versa – a kind of different information mutually overlapping to various degrees. Next, only English windows were again given, and so on – see also the upper part in Fig. 5.

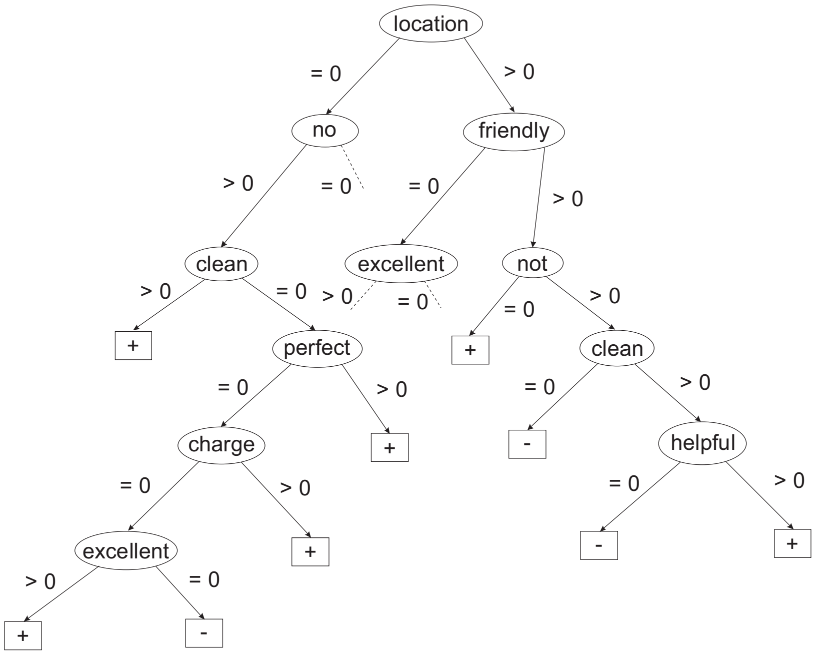

Fig. 3. An example of part of a tree generated using the *c5* algorithm applied to one of the windows. This whole tree size (number of nodes) was 149

After training *c5* for a given window, the tree returned a set of relevant attributes (a list of words) for that subset of reviews. Such a set was compared with the set of the previous window how much it differed in relevant words as it is illustrated in Fig. 2.

Each word in the list can also be considered as a coordinate, thus the list can be represented as a vector (the percentage of a word, given by the frequency how often it is used for questioning by the tree, is here employed as a term numerical representation, that is, its importance weight). The similarities of both sets, A and B, were measured by Jaccard index, $J(A, B) = |A \cap B|/|A \cup B|$; Euclidean distance between points A and B, $d_E = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$, where a_i are coordinates of A and b_i coordinates of B; and cosine similarity (or dot product) of two vectors A and B, $cos(\theta) = (A \cdot B)/(||A||||B||)$, where θ is an angle between both vectors [20]; see also Fig. 4. It was expected that the set distance would express the difference caused by the change of significant words according to the two given classes.

V. Results and discussion

As expected, the distance between two adjacent windows expressed the word change evidently. Each pair of English or Spanish windows generated very similar set of relevant attributes and the distance was small for all three measuring methods (Jaccard, Euclidean, cosine). When the languages were suddenly replaced or gradually mixed, the change manifested itself clearly. It is illustrated in the following two graphs, Fig. 5 and Fig. 6.

The upper part of the graph Fig. 5 shows the changes in the data stream. The horizontal axis represents the sequence of windows containing English, Spanish, or mixed terms. The vertical axis represents the percentage of the English or Spanish reviews in every window. This is actually a graphically expressed input of the investigated textual data.

The lower part of Fig. 5 illustrates graphically represented changes of the significant word contents for the three applied similarity (distance) measures. The horizontal axis is the same as in the upper part. The vertical axis shows measures (cosine similarity, normalized Euclidean distance, Jaccard index) reflecting the changes depicted in the upper part. It is obvious how the similarity of a current analyzed window changes or remains. The horizontal lines (for no changes) are not perfect lines because the adjacent windows, even if without any language changes, are never identical (each window is a subset containing reviews from the original set shared by all windows, and all mutual subset intersections are empty).

The graph Fig. 6 depicts how the distance between adjacent windows may look like when no significant changes occur. Ideally, it would be horizontal straight lines but typically no reviews are identical, so there are moderate deflections within a certain strip. For the investigated data and the scale between 0 and 1, the unvarying concept was represented by a belt less than 0.2 wide for Jaccard index and normalized Euclidean distance, and less than 0.1 wide for cosine similarity as illustrated in Fig. 6. Higher values indicated changes worthy of notice. The steepness of each line segment in Fig. 5 (its lower part) can be used for numerical expression of the change intensity.

As for the real contents of the windows, dictionary sizes fluctuated between 3823 words (one language) up to 5179 ones (mixed English and Spanish), so the difference was ca 25%. English dictionaries per window had less words, 3946 on average, while Spanish ones had 4125. When English:Spanish was 1:1, the dictionary contained 4932 words. The largest dictionary was for the mixture Spanish:English 70:30. Trained decision trees selected only a small fraction of words as relevant attributes. While the dictionaries contained thousands of words per window, only tens of them were relevant for assigning the right label: either a positive or negative review. Due to the limited space, it is not possible to show all 100 lists

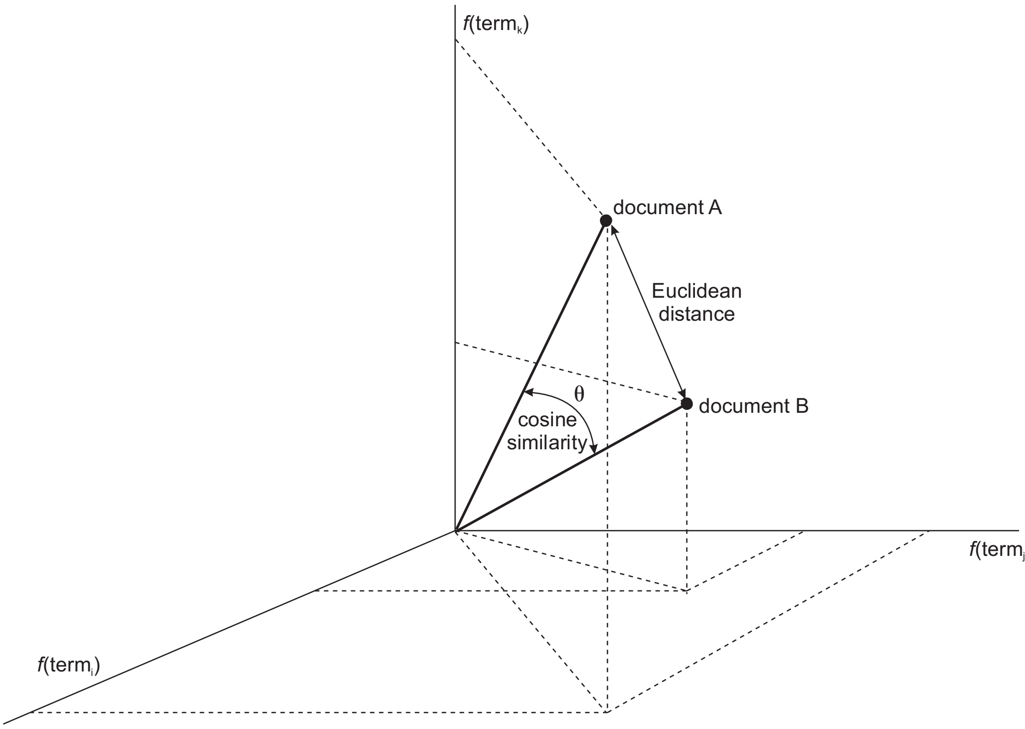

Fig. 4. The illustration of the Euclidean distance and cosine similarity between two documents A and B, here for simplicity's sake in a three-dimensional document space containing terms *i, j,* and *k*, represented by their frequencies $f(term_i)$, $f(term_j)$, and $f(term_k)$. If the angle or distance is zero, the documents are identical

of relevant words, so only three randomly selected windows and their first 25 relevant attributes are here shown (pure English, pure Spanish, and 1:1 mixed English and Spanish windows) – see Table I. The percentage shows how often a word is used for asking (100% means *always,* which is typical for the most relevant words in roots).

TABLE I. AN EXAMPLE OF TEXT-MINED RELEVANT ATTRIBUTES FOR PURE ENGLISH, PURE SPANISH, AND MIXED 1:1 REVIEWS (THE FIRST 25 ATTRIBUTES FOR EACH WINDOW)

pure English	pure Spanish	EN:ESP 1:1
100% location	100% no	100% location
83% friendly	80% excelente	91% fantastic
77% excellent	76% amable	91% debera
71% spacious	74% demasiado	90% friendly
71% not	73% poco	87% poner
70% relaxing	69% ubicación	87% ascensor
70% helpful	68% escaleras	86% pobre
67% relaxed	68% ruido	85% excellent
66% beautiful	67% agradable	84% olor
66% helpfull	67% escaso	83% suelo
65% amazing	66% poca	83% tampoco
65% convenient	66% mala	82% escaso
65% comfortable	64% olor	82% demasiado
64% wonderful	64% mal	81% poca
64% proximity	64% caro	81% amabilidad
61% friendliness	63% mejorar	80% tranquilidad
60% lovely	62% ruidoso	78% ruido
60% shopping	62% nada	77% nadie
59% quiet	61% lejos	77% falta
58% station	61% planta	76% pleasant
56% clean	61% pequenas	75% beautiful
52% welcome	60% algo	75% poco
52% good	60% pequena	72% funcionaba
48% nice	59% necesita	72% lovely
47% cleanliness	58% ascensor	71% convenient

The total difference between the pure English and pure Spanish windows is evident. In the mixed case, there are some words both from Spanish and English reviews, including ones that are not in the pure lists.

The experiments also tried to find out whether using not only strictly separated but half-and-half overlapping adjacent windows could play a significant role. However, the results were quite negligible because the mined relevant words were practically the same – sometimes their order slightly changed by several percent only. Such a behavior is likely given by data and their change type – if the data suddenly changes in strong jump shifts, it might be beneficial to investigate that situation using overlapping windows with looking for an appropriate window-overlapping size. This procedure might reduce the possible jump shifts coming from sharp separation of only slowly changing data contents and thus the main theme.

Regarding the computational complexity, the analysis of a window was fast. The reason was that lower number of instances per window resulted also in lower number of attributes because of not so extensive vocabulary unlike much larger windows – note that the time computational complexity depends quadratically on the number of words. Therefore, the experiments could be performed using a common PC the memory of which (RAM 8 GB) was sufficient. Based on previous experience, to process the whole set of reviews as a batch would need a computer with RAM-size probably at least in the order of hundreds of GB, and the computational time could probably take at least months.

The selected window size provided enough information for recognizing which words played decisive role for correct assignment of a class (label) to a review, not exceeding the accuracy error (estimated by 10-fold cross-validation) more than ca between 12% and 13%. However, the classification was not the primary goal of the presented research.

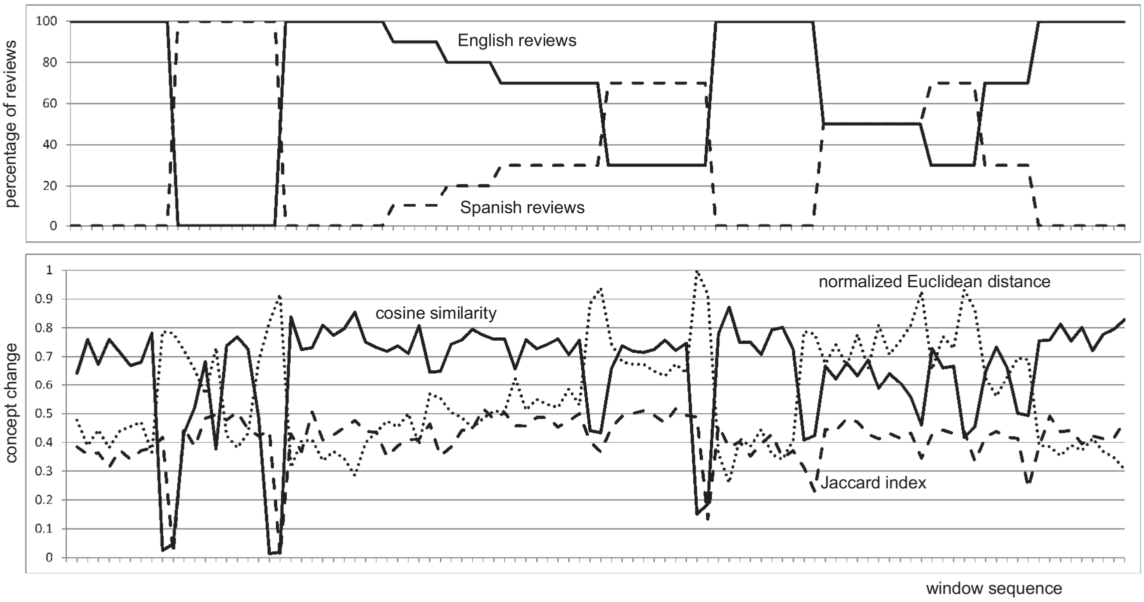

Fig. 5. The illustrations of the extent of changes between adjacent windows. The upper part shows the change degree percentage of a window contents. The lower part shows the influence of the change on the similarity expression between adjacent windows (a current and its previous one)

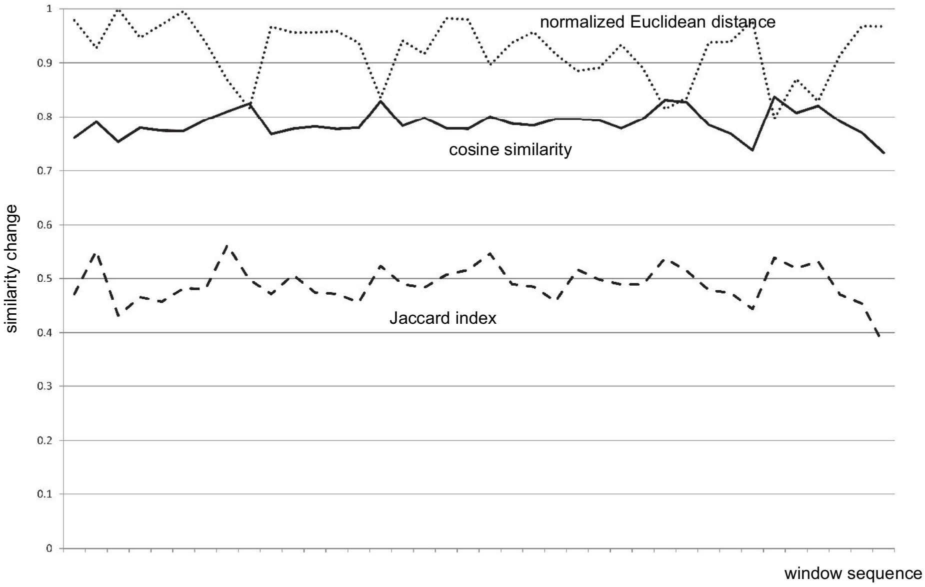

Fig. 6. The illustration of cases when relevant attributes in adjacent windows do not substantially change – the majority of significant words persists

The results of experiments demonstrated that also using a traditional machine-learning algorithm (*c5*), long data streams can be analyzed per partes when the main goal is monitoring the data and revealing significant changes.

VI. Conclusion

In this presented research, a possible way to reveal noticeable changes in a stream of short textual documents written freely in natural languages was suggested. The method is based on text mining. The data stream is not processed in one piece (batch); the data sequence is divided between subsequences represented by adjacent windows containing 10,000 items. For each window, its relevant features (words) are found by applying the popular *c5* decision tree to the window contents. Knowing a label for each text document, the tree can be generated via supervised machine learning. The words are represented numerically by their weighted frequency, *tf-idf*, but the results showed that using simple frequency or just binary representation would provide – in this case – the same result.

The tree returns relevant features. This way, each window provides a set of its significant words according to existing labels (here, two labels were available: a *positive* or *negative* evaluation of hotel service reviews).

Possible changes are measured as a distance between each set of revealed significant words given by adjacent couples of windows. The distances serve as a measure of (dis)similarity between the neighboring windows pairs. Three methods were used: Jaccard index, Euclidean distance, and cosine similarity. All the methods provided qualitatively the same results, detecting either no change in significant words, or a moderate one, or jump shifts. The conceptual difference between the window couples was therefore defined by their word contents. In this research, the changes were simulated by sudden or gradual mixture of two different languages (English and Spanish) used for writing the reviews.

Because the change of relevant words is here important, it is justified to expect that such a monitoring of changes in textual data streams would work also for one (ore more than two) language(s) when the contents regarding to a topic of a text document slowly or sharply changes. Such a method can be applied to analyzing, for example, blogs, discussion groups, customer feedback, and so like.

The concept-change degree evaluation depends on a user because it may be a matter of subjectivity or an application area: What is a sufficient amount of changes to consider it as significant? Different users can see it differently, or such a degree can be determined after evaluating window analyses.

This research now continues with looking for not only individual significant words (called also 1-grams) but also for important collocations (n-grams). The first results indicate that 3-grams can be expected to provide the most valuable information.

Acknowledgment

This investigation was partly supported by the research grant IGA of Mendel University in Brno No. 16/2015.

References

[1] C. C. Aggarwal and C. Zhai, Mining text data, Springer, New York, 2012.

[2] R. Feldman and J. Sanger, The text mining handbook: Advanced approaches in analyzing unstructured data, New York, Cambridge University Press, 2009.

[3] A. Banerjee and S. Basu, ”Topic models over text streams: A study of batch and online unsupervised learning”, *in Proc. SDM-2007 Conf.*, April 2007, pp. 437-442.

[4] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, ”Data stream mining: Practical approach”, J. Machine Learning Research (JMLR), Vol. 11, 2011, MIT Press, pp. 1601-1604.

[5] booking.com official website, Web: http://www.booking.com .

[6] J. R. Quinlan, Data mining tools See5 and C5.0. RuleQuest Research. Official website https://www.rulequest.com/see5-info.html , 2015.

[7] J. Žižka and A. Svoboda, ”Customers’ opinion mining from extensive amount of textual reviews in relation to induced knowledge growth”. J. Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis, in press.

[8] J. Žižka and F. Dařena, ”Parallel processing of very many textual customers’ reviews freely written down in natural languages”, *in Proc. IMMM-2012 Conf.*, Oct. 2012, pp. 147-153.

[9] J. Gama, I. Žliobaitė, M. Pechenizkiy, and A. Bouchachia, ”A survey on concept drift adaptation”, J. ACM Computing Surveys, Vol. 46, No. 4, 2014, pp. 1-37.

[10] J. R. Quinlan, *C4.5: Programs for machine learning*, New York, Morgan-Kaufmann, 1993.

[11] S. Hoeglinger and R. Pears, ”Use of Hoeffding trees in concept based data stream mining”, *in Proc. Information and Automation for Sustainability ICIAFS-2007*, Dec. 2007, pp. 57-62.

[12] A. Bifet and E. Frank, ”Sentiment knowledge discovery in Twitter streaming data”, *in Proc. Discovery Science DS-2010*, Oct. 6-8, 2010, pp. 1-15.

[13] F. Dařena, J. Žižka, and J. Přichystal, ”Clients freely written assessment as the source of automatically mined opinions”, J. Procedia Economics and Finance 12 (2014), Elsevier, pp. 103-110.

[14] J. Žižka and F. Dařena, ”Revealing prevailing semantic contents of clusters generated from untagged freely written text documents in natural languages”. *in Proc. TSD-2013 Conf.*, Sept. 2013, pp. 434-441.

[15] J. Žižka and F. Dařena, ”Automated mining of relevant n-grams in relation to predominant topics of text documents”, *in Proc. TSD-2015 Conf.*, Sept. 2015, pp. 461-469.

[16] V. Ganganwar, ”An overview of classification algorithms for imbalanced datasets”. International Journal of Emerging Technology and Advanced Engineering, Vol. 2, No. 4, 2012, pp. 42-47.

[17] J. Žižka and F. Dařena, ”Mining significant words from customer opinions written in different natural languages”, *in Proc. TSD-2011 Conf.*, Sept. 2011, pp. 211218.

[18] F. Sebastiani, ”Machine learning in automated text categorization”. J. ACM Computing Surveys, Vol. 34, No. 1, 2002, pp. 1-47.

[19] I. Chikalov, *Average time complexity of decision trees*. Intelligent Systems Reference Library, Vol. 21, Springer, 2011.

[20] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*, Pearson, 2005.

[21] G. Widmer and M. Kubat, ”Learning in the presence of concept drift and hidden contexts”, J. Machine Learning 23, 1, 1996, pp. 69101.