SuDoC: Semi-unsupervised Classification of Text Document Opinions Using a Few Labeled Examples and Clustering

František Dařena and Jan Žižka

Department of Informatics, FBE, Mendel University in Brno
Zemědělská 1, 613 00 Brno, Czech Republic
{frantisek.darena,jan.zizka}@mendelu.cz

Abstract. The presented novel procedure named *SuDoC* – or Semi-unsupervised Document Classification – provides an alternative method to standard clustering techniques when it is necessary to separate a very large set of textual instances into groups that represent the text-document semantics. Unlike the conventional clustering, SuDoC proceeds from an initial small set of typical specimen that can be created manually and which provides the necessary bias for generating appropriate classes. SuDoC starts with a higher number of generated clusters and – to avoid over-fitting – reiteratively decreases their quantity, increasing the resulting classification generality. The unlabeled instances are automatically labeled according to their similarity to the defined labeled samples, thus reaching higher classification accuracy in the future. The results of the presented strengthened clustering procedure are demonstrated using a real-world data set represented by hotel guests' unstructured reviews written in natural language.

Keywords: document labeling, clustering, small sample sets, text mining, natural language.

1 Introduction

Managing a very large set of relevant textual documents can provide valuable knowledge as many successful text-mining applications demonstrate [3,4,19]. A typical text-mining task is based on classification or prediction. Having a sufficient number of labeled instances, a user can employ them as training samples for adjusting specific parameters of a chosen classification algorithm: *supervised learning* [17]. Usually, the more textual documents representing individual classes are available, the better knowledge can be revealed, as the practice shows. Unfortunately, not always the respective labels that determine a document category are available. If a set of collected document samples is not too large, the supplementary information in the form of labels can be appended manually by a human expert. Naturally, for very extensive and potentially highly valuable collections of unlabeled documents (maybe hundreds of thousands or millions),

H.L. Larsen et al. (Eds.): FQAS 2013, LNAI 8132, pp. 625–636, 2013.
© Springer-Verlag Berlin Heidelberg 2013

the manual method is impracticable. In such a situation, the labeling can be automatically created with the help of clustering algorithms: *unsupervised learning* [17]. However, because a clustering method has less initial information available, the synthetic, machine-controlled division of the set of unlabeled documents between different categories (clusters) can be often imperfect resulting typically in a lower classification accuracy, purity, precision, and so like.

The *semi-supervised learning* procedure represents a trade-off between the supervised and unsupervised learning. It works initially with usually a small set of instances with known labels, which positively supports the supplementary labeling of a much larger remainder of unlabeled instances. The semi-supervised learning can employ several algorithms as *self-training, co-training, McLachlan, expectation-maximization, boosting, label propagation*, and others – an interested reader can find a good summary in [1].

This article suggests using a small set of labeled instances for improving the clustering process, too. However, because the presented method is not one of the typical semi-supervised learning approaches, here it is called *semi-unsupervised learning*, or *SuDoC – Semi-unsupervised Document Classification*, making reference to its similarity to the principles of the semi-supervised methods: Using a small set of beforehand labeled instances as bias in favor of improved automatic supplementary labeling. Here, the beforehand labeled instances play a role known as *training samples* and the following text uses this term.

At the beginning, a lot of unlabeled instances are biased (by a small number of training samples) to create a high number of clusters because such clusters are characterized in that they have higher purity. (An extreme – but here not used – case could be initiating clusters having just one instance: perfect purity, but no generality.) Then, reiteratively, the following steps decrease the quantity of clusters and propagate the labels through fewer larger clusters to still unlabeled instances, which eventually represent classes. The goal is therefore to create groups from a lot of instances, which provide improved classification functionality.

The method presented in this paper aims at providing an alternative possibility how to deal with one of the typical problems comprised by the missing labels for classification. In the following sections, the article describes the suggested clustering improvement procedure, the experiments carried out with a large real-world data collection from the Internet, and the results and their interpretation as well as a comparison with some selected typical classifiers.

2 Classification Based on a Small Number of Examples

In order to evaluate the performance of different classification methods, the set of all available instances – represented by positive and negative textual unstructured reviews of hotel service customers – was divided into two parts: a training and testing set. Labeled reviews from the training set were used for classification of the documents in the testing set. For the classification, several well-known standard supervised techniques were used. However, in this case, the number of reviews in the training set was very low (not more than 20 reviews) compared to the size of the testing set. Then, the results of those classifiers were compared to the suggested novel approach based on "strengthened" clustering described further.

Because all the reviews in the testing set were labeled too, it was possible to evaluate the quality of classification as well as the SuDoC process. There exist many different metrics applied to evaluation of a classifier performance. The presented experiments used *Accuracy*, *Precision*, *Recall*, and *F-measure* that are commonly accepted for estimating the classification performance evaluation measures [18]. In addition, the performed experiments were repeated 100 times in order to exclude the influence of randomness.

2.1 Training Samples Selection

From all textual reviews that were available (1,245 reviews), a small subset containing maximally 20 reviews was created using random selection from the whole assemblage of available documents. The reviews in the selection were then carefully manually labeled as negative or positive by the authors according to the real content from the human-like semantic point of view. When it was impossible to clearly determine whether a review was positive or negative, the document was simply excluded from the set of labeled samples – certain reviews contained both some positive and negative points of the used accommodation service, not being plainly distinctive. Thus, in the experiments, the subsets contained from 14 to 20 examples (in most cases typically 18 or 19). The sample sets, containing two classes, were relatively balanced, that is, they contained almost the same numbers of positive and negative review samples.

2.2 Using Common Supervised Techniques

To evaluate a possible contribution of SuDoC to processing of the data described in detail in the section 3, the first group of experiments assessed outputs of supervised machine-learning algorithms implemented in the popular data-mining system WEKA [10]. The various algorithms included: J48 [16], Naïve Bayes [12], Logistic Regression [5], Support Vector Machines [15], K-star [6], Instance based learning [2], and J-Rip [7]. During the experiments, default parameters of the mentioned classifiers were used.

2.3 SuDoC – Semi-unsupervised Document Classification

The presented SuDoC approach is grounded on the assumption that it is possible to label all mutually similar instances in a certain group (cluster) using the exploration of just a few of them. The more homogeneous the group is, the smaller amount of instances must be studied. In an ideal situation where the instances are perfectly (or very closely) similar (i.e., they belong to one class according to their sentiment or topic), it is sufficient to examine just one of them in order to label all remaining ones. In a situation that is not perfect (the group contains instances from different classes), the selection of an instance from a minority class might inevitably cause a high error rate of the classification. In order to prevent this situation, more instances as examples should be selected and the label that has the majority among the labels assigned to them can be used for the rest of the cluster, see Fig. 1. If the heterogeneity of the cluster in such a situation is not too high, it is possible to label all instances in the group with a user's acceptable accuracy.

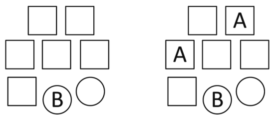

Fig. 1. Two identical clusters containing instances of two classes – class A (squares) and class B (circles); some of the instances are initially labeled (marked with a letter representing the assigned label of the document); remaining instances in the clusters are assigned to the same class that prevail within the randomly selected instances. Left – all instances are assigned to the class B with error 75% (6 from 8); right – all instances are assigned to the class A according the majority class of initially labeled instances with error 25% (2 from 8).

To be successful when using the approach demonstrated above, it is necessary to have *not too many* groups of instances with *sufficient homogeneity*. Such a state can be achieved through a process known as *clustering*. Clustering, as the most common form of unsupervised learning, enables automatic grouping of unlabeled instances into subsets called clusters according to the mutual (dis)similarity of the instances. The instances in each subset are more similar to each other than to instances in other subsets. During assigning the instances to the clusters, a particular clustering criterion function defined over the entire clustering solution is minimized or maximized.

The quality of the clustering solution is frequently measured by *purity*, *entropy*, *mutual information*, or *F-measure* [9]. Purity measures the extent to which each cluster contains instances from primarily one class. In a perfect clustering solution the clusters contain instances from only a single class, i.e., purity equals to 1 [20]. The smaller the clusters are, the higher their homogeneity generally is. In the situation when each cluster contains just one instance, the homogeneity is naturally perfect. This is, however, a quite useless solution lacking any generality as it is over-fitted just to the available instances while the goal is to categorize the instances that would appear in the future. Any solution containing fewer clusters with lower heterogeneity is thus better. It is therefore necessary to find a solution with a low number of clusters having a sufficient quality from a user view.

In the experiments, the clustering process used the software package *Cluto* version 2.1.2 [23]. This free software provides various different clustering methods working with several criterion functions and similarity measures, and it is very suitable for operating on very large datasets [13]. Cluto's criterion functions that are optimized during the clustering process operate either with *vector representation* (internal, external, and hybrid functions) of the objects to be clustered, or with *graph-based representation* (graph-based functions). Internal criterion functions are defined over the instances that are parts of each cluster and do not take into account the instances assigned to different clusters. External criterion functions derive the clustering solution from the difference between individual clusters. Various clustering criterion functions can be also combined to define a set of hybrid criterion functions that simultaneously optimize the individual criterion functions [20].

During the experiments, the following parameters of the clustering were set:

- similarity function: cosine similarity,
- clustering method: k-means (Cluto's specific variation), which iteratively adapts the initial randomly generated k cluster centroids' positions,
- criterion function optimized during clustering process: *H2* (hybrid).

Other clustering parameters of Cluto remained set to their default values, i.e., *Number of iterations*: 10, *Number of trials*: 10, *Cluster selection*: best, and *Row model*: none. The above parameterization was chosen based on the results of the previous experiments published in [22].

As it was mentioned above, sufficiently high quality (acceptable for a user) of clusters is essential for the success of classification based on a few *typical* examples. The relation between the number of clusters and the quality of the clusters was demonstrated in an experiment based on clustering documents represented by the below mentioned customer reviews. Table 1 contains information about the quality of clustering solutions for different numbers of clusters for the given specific data. The quality is measured by *Purity* which is defined as

$$Purity = \sum_{i=1}^{k} \frac{|C_i|}{|D|} \cdot \frac{1}{|C_i|} \max_j(|C_i|_{class=j}),$$

where k is the number of clusters, $|C_i|$ is the size o the i-th cluster, $|D|$ is the number of instances to be clustered, and $|C_i|_{class=j}$ is the number of instances of class j in cluster C_i.

Table 1. Quality of clustering solutions with different numbers of clusters measured by the *Purity* criterion

Number of clusters	1	2	20	50	100	200	300	500	750	1000
Purity	0.53	0.83	0.84	0.86	0.86	0.87	0.88	0.88	0.92	0.97

It is obvious that by labeling the documents in clusters according to a randomly selected document in each cluster, we might achieve about the 90% accuracy when there are 750 clusters created and when at least one document in each cluster is labeled. However, this number is too high for manual labeling. Having, for example, 20 clusters, the accuracy of the same process would be significantly lower (84% or less), also because the chance that a document of a minority class in a cluster might be randomly selected (this is a consequence of lower quality of the clustering solution). The following steps focus on achieving higher accuracy of such a classification with just a small number of specimen documents that need to be manually labeled in advance.

Such a small number, N_i, of documents that has to be manually labeled is the same as described in the section 2.1 in order to compare the presented procedure with commonly used classification techniques.

These initially labeled documents are used to label the remaining documents in the clusters they belong to. When the number of initially labeled documents, N_i, is low compared to the whole number of clusters (a low N_i is desired), we might achieve higher classification accuracy but not all documents will be classified. When N_i is close to the number of clusters, i.e., the number of clusters is low (having a low N_i), we achieve lower accuracy but all or almost all documents will be labeled, see Fig. 2 for a model example representing this situation.

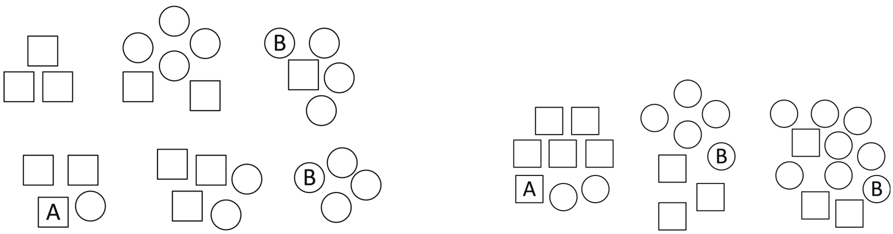

Fig. 2. Two clustering solutions of the same document collection. Squares represent documents of class A and circles documents of class B; documents marked with letters A or B are initially labeled documents. Remaining documents in the clusters are later labeled according to the three initially labeled ones. Left – error rate of such process is about 15% but not all documents are labeled (because three clusters do not contain any initially labeled documents); right – error rate is about 30% but all documents are labeled.

As told above, a higher number of clusters is better for achieving higher classification accuracy. However, the problem of having many unclassified documents must be eliminated. We therefore propose spreading the labels (assigned with higher accuracy in a clustering solution having a higher number of clusters, see Fig. 3) within a clustering solution having a lower number of clusters (see Fig. 4). Thanks to a few initially labeled documents, we can have much more labeled instances available, with a sufficient accuracy (Fig. 3). These labeled documents can be later used for labeling other unlabeled documents that are in other clusters at this moment. Thus, a different clustering solution is needed. In order to ultimately label all document instances, such a solution must have a smaller number of clusters as illustrated in Fig. 4.

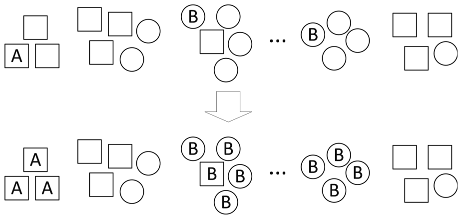

Fig. 3. Top – a few randomly selected examples are labeled. Bottom – labels of the examples are used to label the remaining instances in the clusters; some of the clusters contain instances without labels because the number of selected instances is much smaller than the number of clusters. However, the classification accuracy is higher.

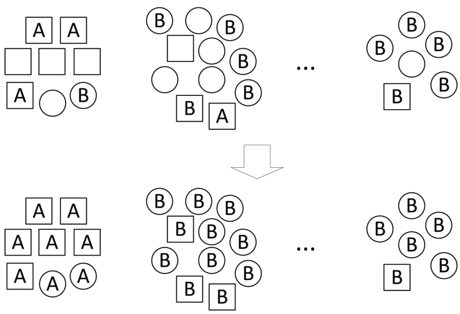

Fig. 4. A different clustering solution of the same collection as in Fig. 3. Top – distribution of the documents with labels that were assigned in the previous step. Bottom – the assigned labels are used to label remaining instances in the clusters. All or most of the clusters contain labeled instances.

The following pseudocode recapitulates the SuDoC algorithm:

```
/* manual labeling of document samples */
FOR EACH sample IN samples DO
   DISPLAY sample.text
   GET sample.label
END FOR

/* numbers of clusters in iterations are given as algorithm parameters */
FOR EACH nc IN number_of_clusters_in_iterations DO
    /* creating nc cluters from all documents, assigning a cluster
       number to every document */
    CREATE_CLUSTERS(all_documents, nc)

    /* counting the occurences of assigned labels in individual clusters */
    FOR EACH sample IN samples DO
        FIND d IN all_documents WHERE d = sample
        ADD sample.label TO cluster_labels[ d.cluster ]
    END FOR
```

```
    /* assigning the majority label to all documents in the clusters
       where at least one document has an assigned label */
    FOR EACH cluster IN cluster_labels
        FIND label with max. frequency IN cluster_labels[ cluster ]
        IF one label found THEN
            FOR EACH d IN all_documents DO
                IF d.cluster = cluster THEN
                    d.label = label
                END IF
            END FOR
        END IF
    END FOR
END FOR
```

3 Data Used in the Experiments

In order to verify the presented approach, several experiments with real-world data were carried out.

3.1 Data Characteristics

The text data used in the experiments was a subset from the data described in [21], containing customers' opinions written in many languages of several millions of hotel guests who – via the on-line Internet service – booked accommodations in many different hotels and countries all over the world. The subset used in our experiments contained 1,245 both positive and negative opinions related to one particular hotel. The data characteristics: *minimal review length* = 1 word, *maximal review length* = 262 words, *average review length* = 30.5 words, *standard deviation* = 35.7 words.

The reviews were always labeled as either positive or negative and this labeling was performed carefully. However, there were several entries that were originally categorized obviously wrongly as the consequence of their authors' errors. For some of the reviews, it was also not possible to determine the opinion polarity without knowing the context. For example, the review "Nothing!" was labeled as positive because it was an answer to a question: "What did you not like about the hotel?" However, this review might be perceived as negative when it would have been an answer to a question: "What did you like about the hotel?" Without knowing such a question (context), one could not decide whether the review was positive or negative.

The reviews were written only by people who made their reservation through the web and who really stayed in the hotel, thus based on their real experience. The reviews were often written quite formally, but most of them embodied all deficiencies typical for texts written in natural languages (i.e., mistyping, transposed letters, missing letters, grammar errors, and so like).

3.2 Data Preprocessing

To be able to use supervised and unsupervised machine learning techniques, the data must be transformed to a representation suitable for selected algorithms. In our experiments, the words in the documents were selected as meaningful units (terms) of the texts. A big advantage of such a word-based representation is its simplicity and the straightforward process of creation [11]. Each of the documents was therefore simply transformed into a bag-of-words, a sequence of words where the ordering was irrelevant. Every document was then represented by a numerical vector where individual dimensions were the words the values of which represented the weights of individual attributes (the words) of the text.

The procedure used the known *tf-idf* (Term Frequency-Inverse Document Frequency, see, e.g., [14]) weighting scheme that usually provided better results than a representation not employing global *idf* weights [22].

The quality of the document vector representation could be increased in several ways, e.g., by using n-grams, adding some semantics, removing very frequent or very infrequent words, eliminating stop words, stemming, but often with only marginal effects [8]. During the preprocessing phase, none of the mentioned techniques was performed.

4 Results

The following section summarizes the results of the experiments described in the previous sections. The non-trivial process of the SuDoC algorithm consists of two major steps, that is:

- preparing clustering solutions of the entire data to be labeled with different numbers of clusters, and
- spreading the labels from initially labeled document instances within at least two clustering solutions, starting with a clustering solution with a higher number of clusters and continuing with one or more clustering solutions having gradually fewer clusters.

The results of both of these tasks could be influenced by several parameter settings. The parameters of the clustering process were already examined in [22] and the settings that were found to provide the best clustering solutions were applied to the presented experiments.

In the second step, the number of clustering solutions used for spreading the known labels and the number of clusters in each of these solutions needed to be determined. In the initial experiments, it was revealed that using just two clustering solutions was quite sufficient. This finding enabled to achieve significantly better results when applying SuDoC than using the well known classifiers, with a lower number of computations than in the case of using three or more clustering solutions. Having more than 1,000 reviews to be automatically labeled, the best results were achieved when the first clustering solution contained a hundred of clusters and the second contained five clusters (100/5). When the numbers of clusters in the two clustering solutions were chosen differently, e.g., 100/10, or 200/10, the classification performance measures reached slightly worse values.

Table 2 contains averaged values of the chosen performance metrics of the experiments with different classifiers – commonly used classifiers on the top of the table, and SuDoC with tree different settings at the bottom (see also Fig. 5).

Table 2. Classification performance metrics for the used classifiers. The presented values are average values obtained from 100 experiments. *Acc* represents Accuracy, *Prec* Precision, *Rec* Recall, and *F* F-measure for the corresponding classes: + for positive reviews and − for negative ones. The numbers as 100/5 following SuDoC stand for the number of clusters in two used clustering solutions.

Classifier	Acc	$Prec^+$	$Prec^-$	Rec^+	Rec^-	F^+	F^-
J48	0.638	0.670	0.647	0.677	0.593	0.650	0.589
Naïve Bayes	0.694	0.699	0.715	0.762	0.619	0.720	0.648
Logistic Regression	0.719	0.736	0.722	0.747	0.688	0.733	0.694
Support Vector Machines	0.706	0.722	0.730	0.751	0.655	0.719	0.669
K-star	0.648	0.659	0.698	0.749	0.534	0.677	0.563
Instance based learning	0.659	0.700	0.704	0.710	0.602	0.668	0.590
J-Rip	0.594	0.652	0.619	0.603	0.583	0.627	0.562
SuDoC(100/5)	0.788	0.865	0.754	0.729	0.851	0.779	0.783
SuDoC(100/10)	0.758	0.823	0.736	0.706	0.810	0.742	0.757
SuDoC(200/10)	0.762	0.816	0.741	0.724	0.799	0.753	0.755

From the presented results, it is obvious that for the given specific task and processed data SuDoC significantly outperformed the commonly used classifiers.

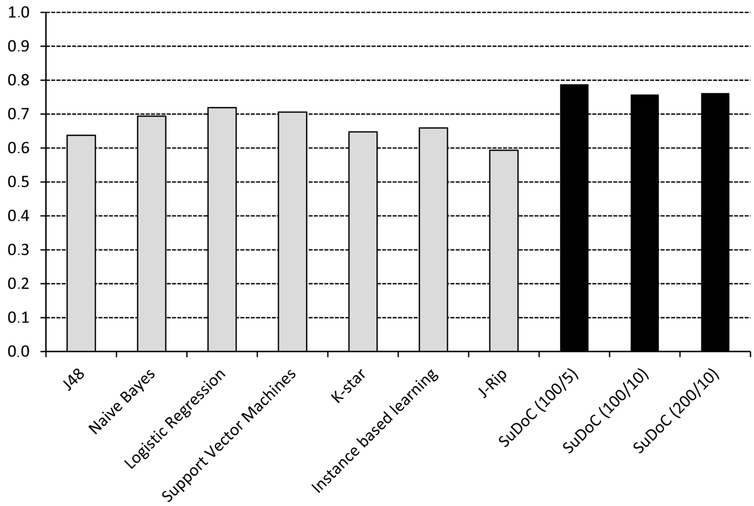

Fig. 5. Accuracy for the used classifiers and SuDoC algorithm

5 Conclusions

This paper presents a novel approach to labeling unknown document instances based on a small number of initially labeled examples. The described approach, called *SuDoC* (Semi-unsupervised Document Classification), can be used as an alternative to commonly used well-known classifiers, such as the Naïve Bayes classifier, Decision Trees, Support Vector Machines, and others. SuDoC's main idea is grounded on using a small number of specimen – a limited set of manually labeled instances representing considered classes.

This set is used for biasing the unlabeled instances so that they get automatically appropriate labels, thus creating classes supporting the future classification or prediction. The classes are generated reiteratively, from a larger number of smaller, less general clusters to a lower quantity of bigger, more general ones. Such a procedure demonstrated better results than applying traditional training of classification algorithms using the limited number of training samples.

The presented results, based on 100 experimental runs for each of the 10 algorithms, their initial conditions and settings, demonstrate that it is possible to achieve better values of the chosen classification performance metrics when using the SuDoC algorithm unlike the traditional clustering procedures.

The future work is going to focus on determining the number of used clustering solutions and the numbers of clusters in each of such solutions which are major aspects of the SuDoC procedure. This will include a large number of experimental runs with the data used in the presented experiments as well as with some different data, and a thorough analysis and comparison of the results. The process will also involve a large amount of manual labeling in order to arrive at representative outcomes. The SuDoC algorithm will be also used for processing documents in different natural languages.

Acknowledgments. This work was supported by the research grants of the Czech Ministry of Education VZ MSM No. 6215648904 and IGA of the Mendel University in Brno No. 4/2013.

References

1. Abney, S.P.: Semisupervised Learning for Computational Linguistics. Chapman & Hall/CRC (2008)
2. Aha, D., Kibler, D.: Instance-based learning algorithms. Machine Learning 6, 37–66 (1991)
3. Berry, M.W., Kogan, J. (eds.): Text Mining: Applications and Theory. John Wiley & Sons (2010)
4. Van Britsom, D., Bronselaer, A., De Tré, G.: Concept Identification in Constructing Multi-Document Summarizations. In: Greco, S., Bouchon-Meunier, B., Coletti, G., Fedrizzi, M., Matarazzo, B., Yager, R.R., et al. (eds.) IPMU 2012, Part II. CCIS, vol. 298, pp. 276–284. Springer, Heidelberg (2012)
5. le Cessie, S., van Houwelingen, J.C.: Ridge Estimators in Logistic Regression. Applied Statistics 41, 191–201 (1992)

6. Cleary, J.G., Trigg, L.E.: K*: An Instance-based Learner Using an Entropic Distance Measure. In: 12th International Conference on Machine Learning, pp. 108–114 (1995)
7. Cohen, W.W.: Fast Effective Rule Induction. In: Twelfth International Conference on Machine Learning, pp. 115–123 (1995)
8. Figueiredo, F., Rocha, L., Couto, T., Salles, T., Goncalves, M.A., Meira, W.: Word co-occurrence features for text classification. Information Systems 36, 843–858 (2011)
9. Ghosh, J., Strehl, A.: Similarity-Based Text Clustering: A Comparative Study. In: Grouping Multidimensional Data, pp. 73–97. Springer, Berlin (2006)
10. Hall, M., et al.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11, 10–18 (2009)
11. Joachims, T.: Learning to classify text using support vector machines. Kluwer Academic Publishers (2002)
12. John, G.H., Langley, P.: Estimating Continuous Distributions in Bayesian Classifiers. In: Eleventh Conference on Uncertainty in Artificial Intelligence, pp. 338–345 (1995)
13. Karypis, G.: Cluto: A Clustering Toolkit. Technical report, University of Minnesota (2003)
14. Nie, J.Y.: Cross-Language Information Retrieval. Synthesis Lectures on Human Language Technologies 3, 1–125 (2010)
15. Platt, J.: Fast Training of Support Vector Machines using Sequential Minimal Optimization. In: Schoelkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods – Support Vector Learning (1998)
16. Quinlan, R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo (1993)
17. Russel, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson Education, Upper Saddle River (2010)
18. Sokolova, M., Japkowicz, N., Szpakowicz, S.: Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. In: Sattar, A., Kang, B.-H. (eds.) AI 2006. LNCS (LNAI), vol. 4304, pp. 1015–1021. Springer, Heidelberg (2006)
19. Weiss, S.M., Indurkhya, N., Zhang, T., Damerau, F.J.: Text Mining: Predictive Methods for Analyzing Unstructured Information. Springer, New York (2010)
20. Zhao, Y., Karypis, K.: Criterion Functions for Document Clustering: Experiments and Analysis. Technical report, University of Minnesota (2003)
21. Žižka, J., Dařena, F.: Mining Significant Words from Customer Opinions Written in Different Natural Languages. In: Habernal, I., Matoušek, V. (eds.) TSD 2011. LNCS, vol. 6836, pp. 211–218. Springer, Heidelberg (2011)
22. Žižka, J., Burda, K., Dařena, F.: Mining Opinion-Clusters from Very Large Unstructured Real-World Textual Data. In: Ramsay, A., Agre, G. (eds.) AIMSA 2012. LNCS, vol. 7557, pp. 38–47. Springer, Heidelberg (2012)
23. `http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download/` (March 2013)