Wiadomości Statystyczne. The Polish Statistician, 2023, *68*(5), 40–57 https://doi.org/10.59139/ws.2023.05.3
Studia interdyscyplinarne. Wyzwania badawcze / Interdisciplinary studies. Research challenges
Zgłoszony/submitted: 20.12.2021, poprawiony/revised: 31.03.2023, zaakceptowany/accepted: 20.04.2023

New algorithm for determining the number of features for the effective sentiment-classification of text documents

Adam Idczak,[a] Jerzy Korzeniewski[b]

Abstract. Sentiment analysis of text documents is a very important part of contemporary text mining. The purpose of this article is to present a new technique of text sentiment analysis which can be used with any type of a document-sentiment-classification method. The proposed technique involves feature selection independently of a classifier, which reduces the size of the feature space. Its advantages include intuitiveness and computational non-complexity. The most important element of the proposed technique is a novel algorithm for the determination of the number of features to be selected sufficient for the effective classification. The algorithm is based on the analysis of the correlation between single features and document labels. A statistical approach, featuring a naive Bayes classifier and logistic regression, was employed to verify the usefulness of the proposed technique. They were applied to three document sets composed of 1,169 opinions of bank clients, obtained in 2020 from a Poland-based bank. The documents were written in Polish. The research demonstrated that reducing the number of terms over 10-fold by means of the proposed algorithm in most cases improves the effectiveness of classification.

Keywords: sentiment analysis, document sentiment classification, text mining, logistic regression, naive Bayes classifier, feature selection, correlation

JEL: C52, C81, M31

Nowy algorytm ustalania liczby zmiennych potrzebnych do klasyfikacji dokumentów tekstowych ze względu na ich wydźwięk emocjonalny

Streszczenie. Analiza sentymentu, czyli wydźwięku emocjonalnego, dokumentów tekstowych stanowi bardzo ważną część współczesnej eksploracji tekstu (ang. *text mining*). Celem artykułu jest przedstawienie nowej techniki analizy sentymentu tekstu, która może znaleźć zastosowanie w dowolnej metodzie klasyfikacji dokumentów ze względu na ich wydźwięk emocjonalny. Proponowana technika polega na niezależnym od klasyfikatora doborze cech, co skutkuje zmniejszeniem rozmiaru ich przestrzeni. Zaletami tej propozycji są intuicyjność i prostota obli-

[a] Uniwersytet Łódzki, Wydział Ekonomiczno-Socjologiczny, Polska / University of Lodz, Faculty of Economics and Sociology, Poland. ORCID: https://orcid.org/0000-0001-9676-2410. Autor korespondencyjny / Corresponding author: adam.idczak@uni.lodz.pl.

[b] Uniwersytet Łódzki, Wydział Ekonomiczno-Socjologiczny, Polska / University of Lodz, Faculty of Economics and Sociology, Poland. ORCID: https://orcid.org/0000-0001-6526-5921.
E-mail: jerzy.korzeniewski@uni.lodz.pl.

czeniowa. Zasadniczym elementem omawianej techniki jest nowatorski algorytm ustalania liczby terminów wystarczających do efektywnej klasyfikacji, który opiera się na analizie korelacji pomiędzy pojedynczymi cechami dokumentów a ich wydźwiękiem. W celu weryfikacji przydatności proponowanej techniki zastosowano podejście statystyczne. Wykorzystano dwie metody: naiwny klasyfikator Bayesa i regresję logistyczną. Za ich pomocą zbadano trzy zbiory dokumentów składające się z 1169 opinii klientów jednego z banków działających na terenie Polski uzyskanych w 2020 r. Dokumenty zostały napisane w języku polskim. Badanie pokazało, że kilkunastokrotne zmniejszenie liczby terminów przy zastosowaniu proponowanej techniki na ogół poprawia jakość klasyfikacji.

Słowa kluczowe: analiza sentymentu, klasyfikacja dokumentów ze względu na wydźwięk emocjonalny, eksploracja tekstu, regresja logistyczna, naiwny klasyfikator Bayesa, dobór cech, korelacja

1. Introduction

Text sentiment classification relates mainly to establishing the sentiment of the opinion expressed in a text. In this context, what proves very important is the text presentation, i.e. converting unstructured text into a machine-readable format using selected features. Therefore, feature selection plays a crucial role in text presentation. Existing literature on text sentiment typically uses a simple textual presentation, namely the bag-of-words (BOW), in which each document is represented by single terms along with their frequencies. The primary downside of the BOW is a huge number of features it produces, which very easily leads to the curse-of-dimensionality. Our study proposes a simple framework based on unigram models, which essentially consists in feature-filtering using distance-based correlation. The correlation is measured for each term-feature between the distances between text document sentiment labels and the distances between frequencies of the terms' occurrence, across all documents.

One can find a comprehensive overview of some techniques of sentiment analysis in Medhat et al. (2014). In some works, the SentiWordNet is used, which basically is a WordNet-based lexicon classifying each term as positive, negative or neutral. In Khan et al. (2011), the SentiWordNet was used to calculate the score and determine the polarity of either subjective or objective sentences from reviews and blog comments. The authors showed that their proposal slightly outperformed maximum-likelihood-based methods. Agarwal et al. (2011) carried out a sentiment analysis on Twitter data. They introduced polarity features, where the polarity was measured by means of the SentiWordNet. Kouloumpis et al. (2011) investigated the usefulness of linguistic features for the establishment of the sentiment of Twitter messages. The authors used a subjectivity lexicon to this end. Davies and Ghahramani (2011) presented a language-independent model for the sentiment analysis of short texts using emoticons as sentiment indicators. Their method slightly outperformed the naive Bayes classifier. Njølstad et al. (2014) proposed and evaluated four different feature categories composed of 26 article features for the

sentiment analysis. Then they used different machine-learning (ML) methods to train sentiment classifier of Norwegian on-line financial news. They achieved classification precision of 71%. Govindarajan (2013) proposed an ensemble classification method that used arcing classifier, naive Bayes and genetic algorithm. The evaluation of the performance of these classifiers was carried out by means of different performance-quality metrics on datasets of film reviews. However, the gain classifiers accuracy was very low.

Yazdani et al. (2017) concentrated on overcoming the limits of the BOW method by studying bigram models and using lexicon-based methods to capture the semantics of words. Iqbal et al. (2019) proposed an approach similar to that of Khan et al. (2011), focusing on maximum-likelihood and lexicon-based methods. The genetic algorithm was used for optimised feature selection. As regards feature space, Pintas et al. (2021) did an extensive review of the literature on this subject. The authors described over a hundred methods and algorithms along with a relatively detailed account of their merits and disadvantages. According to them, feature selection methods could be classified into three categories: filter, wrapper, and embedded methods. Filter methods are independent of the learning process and applicable prior to this process. Wrapper methods collaborate closely with the classifier trying to optimise the whole classification process. Embedded methods involve feature selection at the training stage. The important issues connected with feature selection are: measuring feature relevance, subset search and globalisation. The above-mentioned research demonstrates that the scientific achievements in this field enable the user to choose from amongst a plethora of different algorithms, classifiers, and optimisations. A full search of the whole feature space is impossible due to its high dimensionality.

For this reason, researchers try to develop efficient algorithms that would measure the relevance of single features and eliminate the irrelevant ones. However, the superiority of the new proposals (as claimed by their authors) over the classical methods of feature selection has been marginal. For example, Elakkiya and Selvakumar (2020) or Yassir et al. (2020) achieved the classification accuracy 1–2% higher than the accuracy obtained by means of one of the well-established methods. Moreover, most of the efficient methods are lexicon-based and operate only in English. Some are also very expensive and complicated in computational terms. In particular, as far as the selection of features is considered, no good method of establishing an adequate number of terms has been proposed so far.

The aim of the study is to propose a novel algorithm for the text sentiment analysis which can be used with any type of document sentiment classification method. The proposed technique determines the number of terms most important for the classification purposes and verifies the efficiency of two classifying methods in a reduced term space in a corpus of documents consisting of the opinions of bank clients.

2. Classification algorithms used

As mentioned before, we employed a statistical approach to assess the performance of the proposed technique. More specifically, two methods were used, i.e. naive Bayes classifier and logistic regression (see Idczak, 2021). These methods require the establishment of input data as a set of features, which are derived from a corpus and presented as frequencies of particular terms in particular documents. This type of representation is called unigram and might be presented as the following document-term matrix (DTM):

$$\mathbf{x} = \left[x_{ij}\right], \tag{1}$$

where:
$\mathbf{x}$ is the document-term matrix,
x_{ij} is the frequency of the j-th term occurrence in the i-th document,
$i = 1, \ldots, I$ (I is the total number of documents in a training set),
$j = 1, \ldots, J$ (J is the total number of terms in a training set).

The features or terms or words, i.e. the columns of matrix $\mathbf{x}$ will be denoted by $\boldsymbol{w}_j$, which is the j-th feature or word (j-th column of matrix $\mathbf{x}$). The true class labels of documents will be denoted by $\boldsymbol{w}_0$.

2.1. Naive Bayes

Bayes' rule (Domański & Pruska, 2000) for the document sentiment classification defines a conditional probability that document $\boldsymbol{x}_i$ belongs to class C_k:

$$P(C_k|\boldsymbol{x}_i) = \frac{p_k f(\boldsymbol{x}_i|C_k)}{\sum_{k=1}^{K} p_k f(\boldsymbol{x}_i|C_k)}, \tag{2}$$

where:
C_k is the k-th class, $k = 1, \ldots, K$,
$\boldsymbol{x}_i$ is the i-th document (i-th row of matrix $\mathbf{x}$) with features J,
p_k is the prior probability that the document belongs to class C_k,
$f(\boldsymbol{x}_i|C_k)$ is the probability of occurrence of document $\boldsymbol{x}_i$, providing it belongs to class C_k.

A naive Bayes classifier assigns document $\boldsymbol{x}_i$ to class C_k if the following equation is satisfied:

$$\mathrm{P}(C_k|\boldsymbol{x}_i) = \max_k \mathrm{P}(C_k|\boldsymbol{x}_i), \tag{3}$$

which is equivalent to:

$$\mathrm{P}(C_k|\boldsymbol{x}_i) = \max_k[p_k f(\boldsymbol{x}_i|C_k)]. \tag{4}$$

The above-mentioned classification rule assumes that $\boldsymbol{w}_j$ terms are independently distributed, given the k-th class:

$$f(\boldsymbol{x}_i|C_k) = \prod_{j=1}^{J} f\left(x_{ij}\middle|C_k\right). \tag{5}$$

In order to train a naive Bayes classifier, p_k will be calculated using the following relative-frequency estimation:

$$\hat{p}_k = \frac{n_k}{I}, \tag{6}$$

where n_k is the number of documents that belongs to the k-th class,

while $f(\boldsymbol{x}_i|C_k)$ will be calculated using the subsequent relative-frequency estimation:

$$\hat{p}\left(w_j = x_{ij}|C_k\right) = \frac{n_{ijk}}{n_{jk}}, \tag{7}$$

where:
n_{ijk} is the frequency of the i-th value of the j-th term in the k-th class,
n_{jk} is the frequency of the j-th term in the k-th class.

2.2. Logistic regression

Let us assume that C is the Bernoulli random variable:

$$C \sim Bernoulli(p), \tag{8}$$

that might take one of the two values:

$$C = \begin{cases} 0, \text{when the sentiment of a document is negative,} \\ 1, \text{when the sentiment of a document is positive.} \end{cases} \tag{9}$$

Then the logistic regression (Hosmer et al., 2013) can be written as follows:

$$p = p(C = 0|\boldsymbol{x}_i) = \frac{e^{\beta_0 + \boldsymbol{\beta}^t \boldsymbol{x}_i}}{1 + e^{\beta_0 + \boldsymbol{\beta}^t \boldsymbol{x}_i}}, \tag{10}$$

where β_0 is an intercept, and $\boldsymbol{\beta}$ is a vector of estimated parameters.

It is convenient to apply logit transformation on (10) to obtain some desirable properties of a linear model:

$$ln\left(\frac{p}{1-p}\right) = \beta_0 + \boldsymbol{\beta}^t \boldsymbol{x}_i. \tag{11}$$

As a result, the above-mentioned equation is linear in its parameters, so betas have a handy interpretation in terms of the odds ratio $\left(\frac{e^{\beta_0+\boldsymbol{\beta}^t x_{i'}}}{e^{\beta_0+\boldsymbol{\beta}^t x_i}}\right)$.[1] This means that if one element of $\boldsymbol{x}_i$, x_{ij} increases by one unit (ceteris paribus), the odds ratio will increase by e^{β_j}, i.e. the odds that a document has a negative sentiment (given the increased x_{ij}) increase (decrease) by $(e^{\beta_j} - 1) \cdot 100\%$.

$p(C = 0|\boldsymbol{x}_i)$ in (10) is the probability that document $\boldsymbol{x}_i$ has a negative sentiment, thus the probability that document $\boldsymbol{x}_i$ has a positive sentiment is calculated by the following equation:

$$p(C = 1|\boldsymbol{x}_i) = 1 - p(C = 0|\boldsymbol{x}_i). \tag{12}$$

Document $\boldsymbol{x}_i$ is classified as negative if the following equation is satisfied:

$$p(C = 0|\boldsymbol{x}_i) = max[p(C = 0|\boldsymbol{x}_i),\ p(C = 1|\boldsymbol{x}_i)]; \tag{13}$$

otherwise, the document is considered positive.

Parameters from equation (10) can be estimated by means of the maximum-likelihood method by maximising the following likelihood function:

$$\mathrm{L}(\boldsymbol{\beta}) = \prod_{i=1}^{I} p(C_1|\boldsymbol{x}_i)^{C_i} [1 - p(C_1|\boldsymbol{x}_i)]^{1-C_i}, \tag{14}$$

with respect to parameters β_0 and $\boldsymbol{\beta}$:

$$\widehat{\boldsymbol{\beta}} = \underset{\beta}{\operatorname{argmax}}\, \mathrm{L}(\boldsymbol{\beta}). \tag{15}$$

[1] $e^{\beta_0+\boldsymbol{\beta}^t x_{i'}}$ denotes the odds for feature $\boldsymbol{w}_j$ to be increased by one unit.

3. General description of the methodology

The approach adopted in this study involves feature selection, i.e. reducing the number of terms used in the document classification by selecting the most important ones. Thus, the method is independent of the classification method used subsequently. How to choose the most important terms or arrange all terms in the order of importance? We propose a technique which is connected with a distance-based correlation between features. The method is relatively uncomplicated and, what is most important, may be further used to determine the number of features to be selected. Within its framework, a term-priority list is created on the basis of the correlation between the terms and the document's sentiment. It is not easy to measure the correlation of this kind, therefore we tried to use a distance-based correlation coefficient. The higher the correlation between the document's sentiment distances (i.e. class label distances) and distances between the unigram (1) representation of a given term, the more important the term is. This approach proved very successful in cluster analysis with relation to distance-based correlation between sets of features (see Korzeniewski, 2012), therefore it should work in the classification of documents as well. If the 'jumps' in the document sentiment distances are positively correlated with the 'jumps' in the unigram representation of a given term, this situation resembles a positive correlation between two sets of features, and the importance of both sets for creating a possible cluster structure is emphasised. Formally, the distance-based correlation coefficient (*DBCorr*) between two sets of features A, B is given by the formula:

$$DBCorr(A,B,l) = \frac{\frac{1}{l}\sum_{t=1}^{l}(d_t^A d_t^B) - \bar{d}^A \bar{d}^B}{s^A s^B}, \tag{16}$$

where:

$1 \leq l \leq n$ denotes the number of document pairs drawn without replacement from all pairs of documents,

d_t^A, d_t^B denote distances for the t-th pair of documents coming from sets A and B, respectively, based on relevant features,

$\bar{d}^A$, $\bar{d}^B$, s^A, s^B denote arithmetic means and standard deviations computed from all l distances on both sets of features, respectively.

In the current structure of document classification, both sets of features, A and B, will be one-feature sets. Set A will consist of a feature classifying documents to one of the two classes, and set B will consist of one I-dimensional feature which is a unigram representation of a given term across all the documents in the training set. Let us establish that we will use formula (16) for $l = 50$ with the value of $DBCorr(A,B,l)$ being the arithmetic mean of 1,000 repetitions of drawing sets of documents consisting of 50 items. Thus in the further part of the text, the notation

$DBCorr(A, B)$ will be used instead of $DBCorr(A, B, l)$. We will use the Manhattan distance (a sum of absolute differences across all coordinates) on both features (or sets of features).

We propose the following formal description of the procedure of creating the feature-priority list:

- find $DBCorr(\boldsymbol{w}_0, \boldsymbol{w}_j)$ for each feature $\boldsymbol{w}_j$ present in the training set;
- rank all terms $\boldsymbol{w}_j$ according to the decreasing order of $DBCorr(\boldsymbol{w}_0, \boldsymbol{w}_j)$.

In order to present the newly-proposed algorithm for determining the number of features to be selected, at first the whole classification procedure should be run. The evaluation of the results should enable the formulation of the proposal. Therefore, our paper is further arranged as follows:

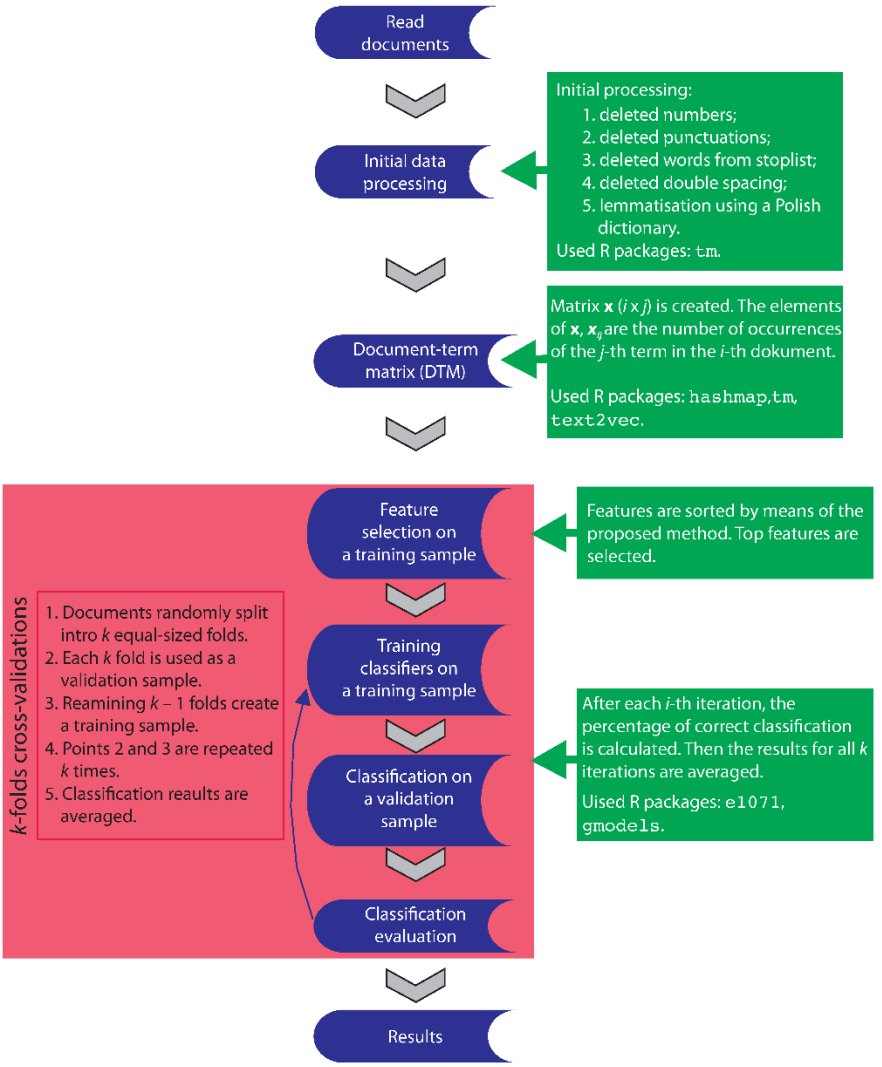
- presentation of the document classification procedure with respect to documents' sentiment;
- introduction of the new algorithm for determining the number of features (to be selected);
- assessment of the quality of the proposed algorithm by means of its application to the data sets used in the sentiment classification experiment.

4. Classification experiment

4.1. Experiment set-up

In order to assess the effectiveness of the proposed technique for the classification of a document sentiment, we conducted an experiment in line with the algorithm presented in Figure 1. All calculations were made by means of R software. First, the documents analysed were read into the memory and then initially processed, i.e. the unwanted numbers, punctuation marks and words were deleted. Lemmatisation was another, and a very important part of this step. This process groups the inflected forms of the word so that they can be analysed as a single item (word's lemma), e.g. *płakać* is lemma for *płakał, płakaliśmy, płacze*. It is especially important in the case of the Polish language, which is inflected. The lemmatisation was performed by means of functions from `tm` package in R and a Polish dictionary. This step might have a crucial impact on features (and their number) in the document-term matrix. For the purpose of this study, unigrams were considered. The DTM matrix was calculated by means of `hashmap`, `tm` and `tex2vec` packages. After the DTM was created, our novel technique was employed to sort features according to their relevance. Then the matrix was used for the purpose of 10-fold cross-validation, as shown in Figure 1, where a naive Bayes classifier and logistic regression were taught on a training sample which consisted of the most relevant features. The classification was assessed on a validation sample. This part of the algorithm was handled by `e1071` and `gmodels` packages. The classification was evaluated in terms of accuracy. A pseudocode is presented below as a supplement to Figure 1.

Figure 1. Algorithmic description of the experiment



 # Initial processing

Step 1. Delete numbers.
Step 2. Delete words from stoplist.
Step 3. Delete punctuation marks.
Step 4. Delete double spacing.
Step 5. Lemmatise each term using the Polish dictionary.
 # Creating document-term matrix
Step 6. Create document-term matrix **x** containing unigrams.
 # Creating k-fold validation structure
Step 7. Split all documents in k equal-sized folds. Each fold will be used as a validation set and the remaining k–1 folds will create a training set.

 # Ordering features with respect to their importance to documents' sentiment
Step 8. $x_{ij} \leftarrow frequency\ of\ term\ j\ in\ document\ i;$
$x_{i0} \leftarrow sentiment\ of\ document\ i; either\ 1\ or\ 0;$
Step 9. Draw dependently set of 50 documents from the training set.
$d_{\cdot}^{j} \leftarrow distance\ on\ feature\ j\ between\ a\ pair\ documents\ in\ the\ set;$
$d_{\cdot}^{0} \leftarrow distance\ between\ class\ labels\ for\ a\ pair\ of\ documents\ in\ the\ set;$
$DBCorr(\boldsymbol{w}_j, \boldsymbol{w}_0) \leftarrow \frac{\frac{1}{I}\sum_{t=1}^{I}(d_t^j d_t^0) - \bar{d}^j \bar{d}^0}{s^j s^0}$
Step 10. Repeat Step 9 1,000 times to find the average $\overline{DBCorr}(\boldsymbol{w}_j, \boldsymbol{w}_0)$ for each $j = 1, \ldots, J$.
Step 11. Arrange all J features in decreasing order of $\overline{DBCorr}(\boldsymbol{w}_j, \boldsymbol{w}_0)$
Step 12. Train classifier on the training set.
Step 13. Classify documents from the validation set. Find the percentage of correct classifications.
Step 14. Repeat Step 8 to Step 13 for each fold k. Find the average percentage of correct classifications.
Source: authors' work.

4.2. Data sets

The proposed method was verified on three datasets. The documents were client reviews on one of Poland-based banks. Each document was labelled with a positive or negative sentiment (positive or negative class). These labels were assigned manually by an opinion holder (a client) by choosing a happy- or a sad-face icon. Each dataset consisted of 302 features. There were 192 positive and 198 negative documents in the first dataset, 198 positive and 192 negative documents in the second dataset, and 188 positive and 201 negative documents in the third dataset.

4.3. Classification results

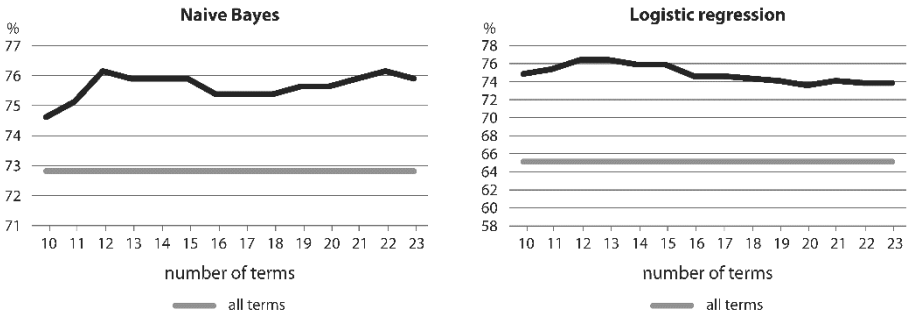
A novel feature-ordering method was applied to the three datasets comprising clients' reviews. The classification was performed by means of the naive Bayes classifier and the logistic regression. The results show that the proposed feature selection method for $k = 10, 11, \ldots, 23$ outperformed the classification based on all terms from all the three datasets (see figures 2–4), although there were some differences. The effectiveness of classification was assessed in terms of accuracy:

$$accuracy = \frac{TP + TN}{I}, \tag{17}$$

where:
TP is the number of documents with a positive sentiment, classified as positive,
TN is the number of documents with a negative sentiment, classified as negative,
I is the number of classified documents.

According to Figure 2, both the naive Bayes and the logistic regression on $k = 10, 11, \ldots, 23$ most relevant features from dataset 1 achieved better results (on average 75.64% for the naive Bayes and 74.85% for the logistic regression) than the classification based on all terms from this set (72.82% and 65.13%, respectively).

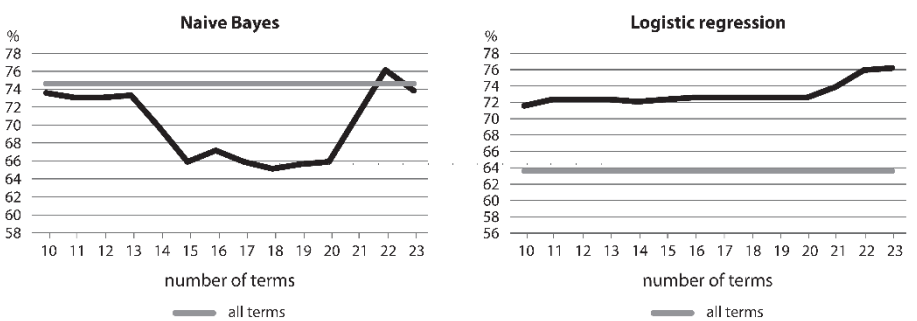
Figure 2. Accuracy of classification – dataset 1



Source: authors' calculation based on dataset 1.

The proposed feature selection method performed slightly poorer than the naive Bayes (on average 69.96% vs 74.62%) on dataset 2 (see Figure 3). It is worth mentioning that for $k = 22$, the adopted procedure still outperformed the classification based on all terms (76.15% vs 74.62%). On the other hand, the results of logistic regression were more consistent, i.e. k most relevant features yielded 72.97% vs 63.59% on average.

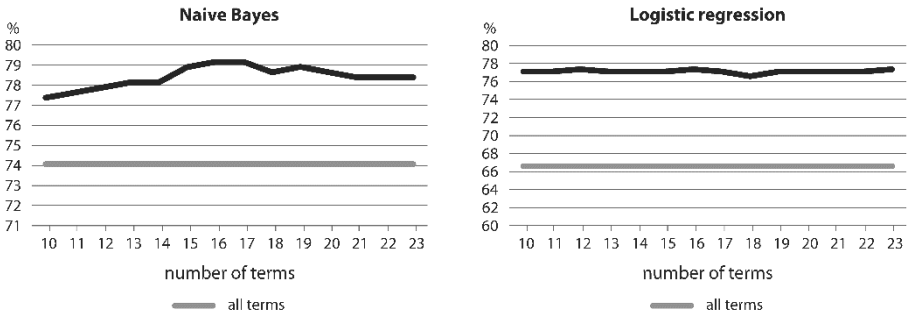
Figure 3. Accuracy of classification – dataset 2



Source: authors' calculation based on dataset 2.

The results obtained from dataset 3 (Figure 4) also confirm the usefulness of the method. In all the cases ($k = 10, 11, \ldots, 23$), both the naive Bayes and the logistic regression performed better than the classification based on all terms. Naive Bayes classified 78.41% of the documents correctly, while the all-terms approach did so in 74.06% cases. Logistic regression was also successful, having classified 77.13% of documents correctly, compared to 66.59% correct classifications by the all-terms approach.

Figure 4. Accuracy of classification – dataset 3



Source: authors' calculation based on dataset 3.

Detailed results of the classification are presented in Table 1. The highest accuracy scores for a given classifier are bolded.

Table 1. Detailed classification results on three datasets

Number of features (*k*)	Dataset 1		Dataset 2		Dataset 3	
	naive Bayes	logistic regression	naive Bayes	logistic regression	naive Bayes	logistic regression
10	74.62	74.87	73.59	71.54	77.37	77.11
11	75.13	75.38	73.08	72.31	77.62	77.11
12	**76.15**	**76.41**	73.08	72.31	77.88	**77.36**
13	75.90	**76.41**	73.33	72.31	78.14	77.11
14	75.90	75.90	69.74	72.05	78.14	77.11
15	75.90	75.90	65.90	72.31	78.90	77.11
16	75.38	74.62	67.18	72.56	**79.16**	**77.36**
17	75.38	74.62	65.90	72.56	**79.16**	77.11
18	75.38	74.36	65.13	72.56	78.64	76.59
19	75.64	74.10	65.64	72.56	78.91	77.11
20	75.64	73.59	65.90	72.56	78.65	77.11
21	75.90	74.10	71.03	73.85	78.39	77.11
22	**76.15**	73.85	**76.15**	75.90	78.39	77.11
23	75.90	73.85	73.85	**76.15**	78.39	**77.36**
All features	72.82	65.13	74.62	63.59	74.06	66.59

Source: authors' calculation based on datasets 1–3.

Table 1 shows that within dataset 1, the highest accuracy for the naive Bayes was obtained for $k = 12$ and $k = 22$ (76.15%), and it was higher by about 3.3 p.p. than the accuracy achieved by means of the all-terms classification. As for logistic regression, the best result was obtained for $k = 12$ and $k = 13$ (76.41%), and it was higher by approximately 11.3 p.p. than the result of the all-terms classification.

Within dataset 2, the highest accuracy for the naive Bayes was obtained for $k = 22$ (76.15%), which was higher by about 1.5 p.p. than in the case of the all-terms

classification. As for logistic regression, the highest accuracy was achieved for $k = 23$ (76.15%), which was higher by approximately 12.6 p.p. than the highest accuracy obtained by means of the all-terms classification.

As regards dataset 3, the highest accuracy for the naive Bayes was obtained for $k = 16$ and $k = 17$ (79.16%), which was higher by about 5.1 p.p. than the accuracy achieved by the all-terms classification. As for logistic regression, the highest accuracy was obtained for $k = 12, 16, 23$ (77.36%), and it was higher by approximately 10.8 p.p. than the accuracy achieved by the all-terms classification.

5. Algorithm for determining the number of features

How to find the adequate number k of features that should be selected? Let us assume that the criterion for the evaluation of the effectiveness of such an algorithm is based solely on the accuracy of the subsequent document classification. The already-verified fact that the classification based on a narrower number of selected features yielded better results than the one based on all features does not conclude this research. There are numerous other algorithms for reducing feature space in the document sentiment classification, and some of them might even give better results than our approach, but none of them is able to predict the adequate number of features to be selected. Distance-based correlation will be further used in such a way as to allow the prediction of the optimal number of features to be selected. The subject of determining the smallest possible number of features has not been investigated extensively in the literature so far, and to our best knowledge, no effective method to this end has yet been discovered.

As the use of single-feature sets is too weak to trace any connections between the number of features and the document sentiment labels, we propose computing distance-based correlations given by formula (16) for subsets A, consisting of two features, and B, consisting of one feature (document sentiment labels). Generally, investigating distance-based correlation for feature subsets consisting of several features does not make sense due to the curse of dimensionality. But in the current structure, there are only two features in one set and one feature in the other. Secondly, the features are already ordered in the decreasing order with respect to their one-to-one correlations with the sentiment labels. If there is an optimal number k of features, then, logically, the distance-based correlations between the 'better' (more meaningful) features positioned to the left of k should be much stronger than those of the features positioned to the right of k. Therefore, we propose to compute distance-based correlations for all the pairs of features from the set of $k-1$ features to the left of k and find the arithmetic mean $r1$ of these

correlations. In a similar manner, distance-based correlations for all pairs of features from the set of $k-1$ features to the right of k should be calculated, and the arithmetic mean $r2$ of these correlations should be computed.

Subsequently, we have to investigate the flow of $r1-r2$, i.e. the differences between the two correlations for $k = 5, 6, \ldots, 23$, and choose the k corresponding to the last local maximum. The algorithm formulated in this way needs determining the range of features arranged in a sequence from which the best candidate will be selected. In the case of our three data sets, the range $k = 5, 6, \ldots, 23$ was chosen because further features have very weak (below 0.005) distance-based correlation with document class labels.

Figure 5. The pseudocode of the algorithm for determining the optimal number of features

 # Selecting the optimal number of initial features from the set of all features arranged with respect to # their decreasing relevance to the document's sentiment

Step 1. Draw dependently a set of 50 documents from the training set.

$d_t \leftarrow distance\ on\ two\ features\ i, j\ between\ documents\ from\ pair\ t;$
$\bar{d} \leftarrow mean\ of\ distances\ d_t\ from\ all\ pairs\ t;$
$s \leftarrow standard\ deviation\ of\ distances\ d_t\ from\ all\ pairs\ t;$
$d_t^0 \leftarrow distance\ between\ class\ labels\ for\ pair\ t\ of\ documents;$
$\bar{d}^0 \leftarrow mean\ of\ distances\ d_t^0\ from\ all\ pairs\ t;$
$s^0 \leftarrow standard\ deviation\ of\ distances\ d_t^0\ from\ all\ pairs\ t;$

$$DBCorr(w_{ij}, \boldsymbol{w}_0) \leftarrow \frac{\frac{1}{l}\sum_{t=1}^{l}(d_t d_t^0) - \bar{d}\bar{d}^0}{s \cdot s^0}$$

Step 2. Repeat Step 1 1,000 times to find the average $\overline{DBCorr}(w_{ij}, \boldsymbol{w}_0)$.

Step 3. Find the 'left' mean $r1 = \frac{\sum_{i<j<k} \overline{DBCorr}(w_{ij}, \boldsymbol{w}_0)}{0.5 \cdot (k-1) \cdot (k-2)}$ for every $k = 11, 12, \ldots, 23$.

Step 4. Find the 'right' mean $r2 = \frac{\sum_{k<i<j\leq 2k} \overline{DBCorr}(w_{ij}, \boldsymbol{w}_0)}{0.5 \cdot (k-1) \cdot (k-2)}$ for every $k = 11, 12, \ldots, 23$.

Step 5. Select $k \in \{11, 12, \ldots, 23\}$ which corresponds to the last local maximum of $r1 - r2$.

Source: authors' work.

6. Assessment of the algorithm

The new algorithm was be applied to the three data sets that had already been classified. In Table 2, we present the detailed differences between mean correlations and single correlations for the most meaningful features. It is much easier, however, to investigate these numbers in graphical form (we took this into account and Figure 5 consists of three graphs).

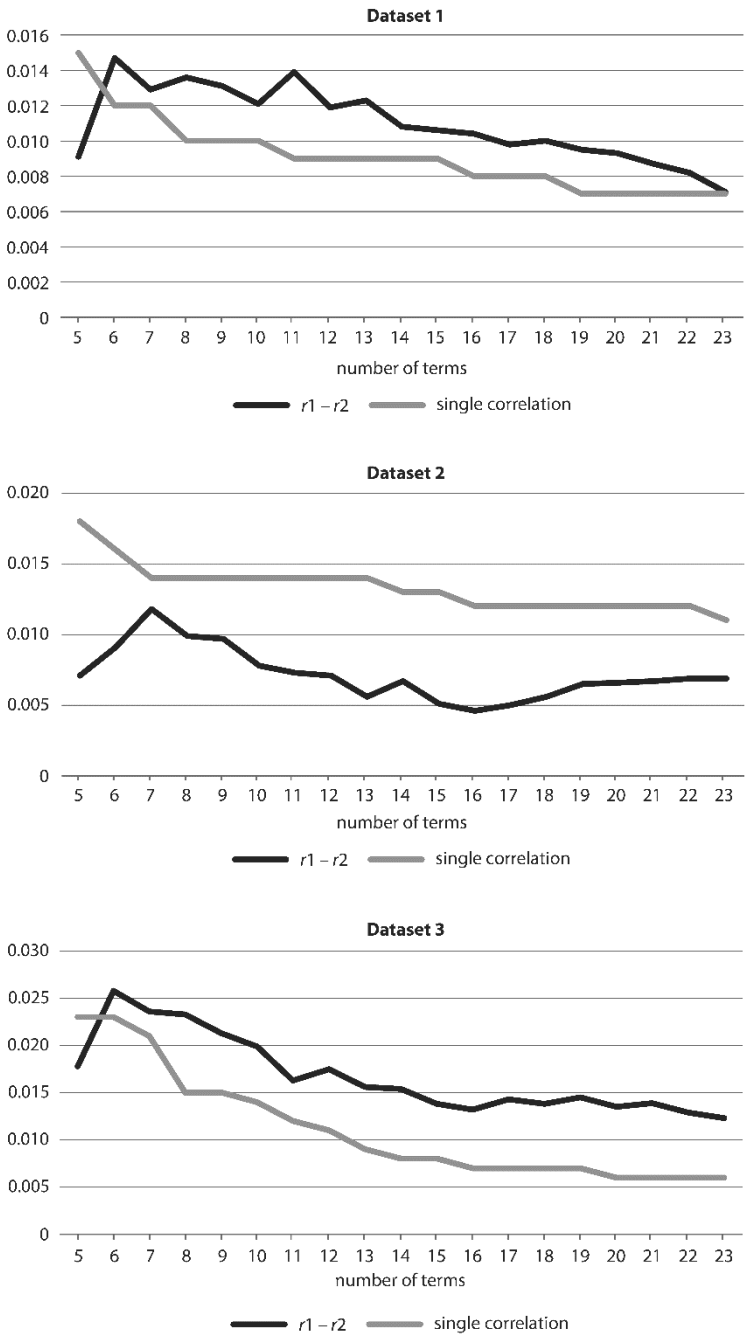
Table 2. Differences between means of correlations and single-feature correlations

Number of features (k)	Dataset 1		Dataset 2		Dataset 3	
	$r1 - r2$	single correlations	$r1 - r2$	single correlations	$r1 - r2$	single correlations
5	0.0091	0.015	0.0071	0.018	0.0178	0.023
6	0.0147	0.012	0.0091	0.016	0.0258	0.023
7	0.0129	0.012	0.0118	0.014	0.0236	0.021
8	0.0136	0.010	0.0099	0.014	0.0233	0.015
9	0.0131	0.010	0.0097	0.014	0.0213	0.015
10	0.0121	0.010	0.0078	0.014	0.0199	0.014
11	0.0139	0.009	0.0073	0.014	0.0163	0.012
12	0.0119	0.009	0.0071	0.014	0.0175	0.011
13	0.0123	0.009	0.0056	0.014	0.0156	0.009
14	0.0108	0.009	0.0067	0.013	0.0154	0.008
15	0.0106	0.009	0.0051	0.013	0.0138	0.008
16	0.0104	0.008	0.0046	0.012	0.0132	0.007
17	0.0098	0.008	0.0050	0.012	0.0143	0.007
18	0.0100	0.008	0.0056	0.012	0.0138	0.007
19	0.0095	0.007	0.0065	0.012	0.0145	0.007
20	0.0093	0.007	0.0066	0.012	0.0135	0.006
21	0.0087	0.007	0.0067	0.012	0.0139	0.006
22	0.0082	0.007	0.0069	0.012	0.0129	0.006
23	0.0071	0.007	0.0069	0.011	0.0123	0.006

Source: authors' calculations based on datasets 1–3.

The first striking observation is that the only set with single-feature correlations higher than differences $r1 - r2$ is dataset 2. This is not very surprising given that (as e.g. the naive Bayes classifier suggests) this set might prove quite difficult for classification (due to the uncertainty as to what number of features should be chosen). For dataset 1, the algorithm performed well, because it was clear that the last local maximum corresponded to $k = 18$. The choice of 18 was probably not the best one; 13, 14 or even 22 would be better in view of the numbers from Table 1. However, the classification accuracy for the most relevant initial features of $k = 18$ was higher than the classification accuracy for all the features. As regards dataset 2, the results were slightly more difficult to interpret, because the first region of the local maxima started with $k = 7$ and ended with $k = 14$ or $k = 15$. Later on, however, i.e. for $k \geq 22$, the numbers started growing again. Thus, the assumption that $k = 22$ or $k = 23$ runs in line with the algorithm formulation. Numbers from Table 1 confirm the above. For example, in the case of the naive Bayes classifier, selecting 22 was one of few options that guaranteed relatively effective classification compared to the performance of all the features. For the logistic regression likewise – the best possible choice was 22 or 23. For dataset 3, the algorithm selected $k = 21$, which, by no means being the best choice, was anyway better than the result for all the features in the light of numbers from Table 1 for both classifiers.

Figure 6. Comparison of differences between the means and individual feature correlations



Source: authors' calculation based on datasets 1–3.

7. Conclusions

In this paper, we propose a simple technique for the text sentiment classification based on unigram models. Essentially, it consists in feature-filtering by means of distance-based correlation. The correlation is measured for each term-feature $\boldsymbol{w}_i$ between the distances between text document sentiment labels and the distances between the frequencies of occurrence of terms across all documents. It is possible to order the terms, so that the impact of the curse of dimensionality becomes softened. This is because it is sufficient to find the correlation for two groups of terms positioned prior and posterior to the currently-assessed term-feature $\boldsymbol{w}_i$. Thus, the proposed procedure is computationally non-complex. Its other advantages include the fact that it enables users to customise the algorithm and choose any classifier, and that it is lexicon-free and easy to use.

The experimental results based on three datasets of clients' reviews show that the proposed method yields better results than the standard full bag-of-word approach. Moreover, the proposed distance-based correlation algorithm for ordering features according to their significance for determining the document sentiment allows the application of distance-based correlation also to the choice of the best number of initial (most meaningful) features that would ensure the effective classification. This algorithm consists in comparing distance-based correlation for two-feature subsets positioned to the left and to the right of any single feature that already has a place in the sequence of features.

The only limitation of this algorithm are situations in which the corpus of documents and the resulting set of features are so large that distance-based correlations between single features and the documents sentiment labels are too weak (< 0.005) to make reasonable ordering of features possible.

Our technique may be used to upgrade any text classifier with respect to text sentiment. We believe the fact that our approach uses distance-based correlations between terms and text-sentiment labels opens it for further development. More specifically, it might be applied to larger text corpora and sets of terms, providing that that one carefully uses the number of terms included in the set of terms for the correlation assessment. Ever-faster algorithms become increasingly desirable in modern societies, where time efficiency is crucial for the effective performance of the growing amount of online work (both clerical and analytical). Such work is often based on the recognition of the sentiment of text documents.

References

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011). Sentiment Analysis of Twitter Data. W: *LSM '11: Proceedings of the Workshop on Languages in Social Media* (s. 30–38). Association for Computational Linguistics.

Davies, A., & Ghahramani, Z. (2011). Language-independent Bayesian sentiment mining of Twitter. W: *The fifth SNAKDD Workshop 2011 on Social Network Mining and Analysis* (s. 99–106).

Domański, C., & Pruska, K. (2000). *Nieklasyczne metody statystyczne*. Polskie Wydawnictwo Ekonomiczne.

Elakkiya, E., Selvakumar, S. (2020). GAMEFEST: Genetic Algorithmic Multi Evaluation measure based FEature Selection Technique for social network spam detection. *Multimed Tools and Application*, *79*(11–12), 7193–7225. https://doi.org/10.1007/s11042-019-08334-1.

Govindarajan, M. (2013). Sentiment Analysis of Movie Reviews using Hybrid Method of Naive Bayes and Genetic Algorithm. *International Journal of Advanced Computer Research*, *3*(4), 139–145. https://accentsjournals.org/PaperDirectory/Journal/IJACR/2013/12/21.pdf.

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied Logistic Regression* (3rd ed.). John Wiley & Sons. https://doi.org/10.1002/9781118548387.

Idczak, A. P. (2021). Sentiment Classification of Bank Clients' Reviews Written in the Polish Language. *Acta Universitatis Lodziensis. Folia Oeconomica*, (2), 43–56. https://doi.org/10.18778/0208-6018.353.03.

Iqbal, F., Hashmi, J. M., Fung, B. C. M., Batool, R., Khattak, A. M., Aleem, S., & Hung, P. C. K. (2019). A Hybrid Framework for Sentiment Analysis Using Genetic Algorithm Based Feature Reduction. *IEEE Access, 7*, 14637–14652. http://doi.org/10.1109/ACCESS.2019.2892852.

Khan, A., Baharudin, B., & Khan, K. (2011). Sentiment Classification Using Sentence-level Lexical Based Semantic Orientation of Online Reviews. *Trends in Applied Sciences Research*, *6*(10), 1141–1157. https://doi.org/10.3923/tasr.2011.1141.1157.

Korzeniewski, J. (2012). *Metody selekcji zmiennych w analizie skupień. Nowe procedury.* Wydawnictwo Uniwersytetu Łódzkiego. http://dx.doi.org/10.18778/7525-695-6.

Kouloumpis, E., Wilson, T., & Moore, J. (2011). Twitter Sentiment Analysis: The Good the Bad and the OMG!. *Proceedings of the Sixteenth International AAAI Conference on Web and Social Media*, *5*(1), 538–541. https://doi.org/10.1609/icwsm.v5i1.14185.

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, *5*(4), 1093–1113. https://doi.org/10.1016/j.asej.2014.04.011.

Njølstad, P. C. S., Høysæter, L. S., Wei, W., & Gulla, J. A. (2014). Evaluating Feature Sets and Classifiers for Sentiment Analysis of Financial News. W: *WI-IAT '14: Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)* (p. 71–78). IEEE. https://doi.org/10.1109/WI-IAT.2014.82.

Pintas, J. T., Fernandes, L. A. F., & Garcia, A. C. B. (2021). Feature selection methods for text classification: a systematic literature review. *Artificial Intelligence Review*, *54*(8), 6149–6200. https://doi.org/10.1007/s10462-021-09970-6.

Yassir, A. H., Mohammed, A. A., Alkhazraji, A. A. J., Hameed, M. E., Talib, M. S., & Ali, M. F. (2020). Sentimental classification analysis of polarity multi-view textual data using data mining techniques. *International Journal of Electrical & Computer Engineering* (2088–8708), *10*(5), 5526–5533. http://doi.org/10.11591/ijece.v10i5.pp5526-5534.

Yazdani, S. F., Murad, M. A. A., Sharef, N. M., Singh, Y. P., & Latiff, A. R. A. (2017). Sentiment Classification of Financial News Using Statistical Features. *International Journal of Pattern Recognition and Artificial Intelligence, 31*(3), 1–34. https://doi.org/10.1142/S0218001417500069.