Int. J. Appl. Math. Comput. Sci., 2018, Vol. 28, No. 4, 787–801
DOI: 10.2478/amcs-2018-0060

A CASE STUDY IN TEXT MINING OF DISCUSSION FORUM POSTS: CLASSIFICATION WITH BAG OF WORDS AND GLOBAL VECTORS

PAWEŁ CICHOSZ [a]

[a] Institute of Computer Science
Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland
e-mail: p.cichosz@elka.pw.edu.pl

Despite the rapid growth of other types of social media, Internet discussion forums remain a highly popular communication channel and a useful source of text data for analyzing user interests and sentiments. Being suited to richer, deeper, and longer discussions than microblogging services, they particularly well reflect topics of long-term, persisting involvement and areas of specialized knowledge or experience. Discovering and characterizing such topics and areas by text mining algorithms is therefore an interesting and useful research direction. This work presents a case study in which selected classification algorithms are applied to posts from a Polish discussion forum devoted to psychoactive substances received from home-grown plants, such as hashish or marijuana. The utility of two different vector text representations is examined: the simple bag of words representation and the more refined embedded global vectors one. While the former is found to work well for the multinomial naive Bayes algorithm, the latter turns out more useful for other classification algorithms: logistic regression, SVMs, and random forests. The obtained results suggest that post-classification can be applied for measuring publication intensity of particular topics and, in the case of forums related to psychoactive substances, for monitoring the risk of drug-related crime.

Keywords: text mining, discussion forums, text representation, document classification, word embedding.

1. Introduction

Despite the rapid growth of other types of social media, Internet discussion forums remain a highly popular communication and knowledge exchange channel. It is common to see discussion forums dedicated to several specific domains, such as art and artists, sports, collectible items, cars, electronic devices, operating systems, and programming languages, to name only a few of the most typical examples. They usually contain richer, deeper, and longer discussions than microblogging services, such as Twitter or Facebook. Unlike regular blogs, they include posts from numerous authors with vastly varying levels of activity, writing styles and skills, as well as proficiency in the area to which the forum is devoted. This makes discussion forums both an interesting and challenging source of data for text mining (Marra *et al.*, 2004; Lui *et al.*, 2007; Said and Wanas, 2011; Holtz *et al.*, 2012).

1.1. Motivation. Internet discussion forums can be used for analyzing user interests and sentiments, particularly associated with topics of long-term, persisting involvement and areas of specialized knowledge or experience. Discovering and characterizing such topics and areas by text mining algorithms can serve various applications specific to the domains of particular forums. Some obvious examples include the following:

- mining forums devoted to certain types of products (e.g., cars, smartphones) to discover requested features, usage patterns, common complaints, recommended products, solutions to typical problems, etc.;
- mining forums devoted to certain types of hobbies (e.g., cycling, photography) to discover associated equipment needs, interest differences between beginner and advanced users, etc.;
- mining forums devoted to certain actually or potentially illegal activities (e.g., soccer hooliganism, racist violence, street racing, drug production and distribution or use) to discover their types and monitor the risk of possible illegal behavior, etc.

To make such applications possible, appropriate algorithms for various underlying text mining tasks are necessary, which would be capable of producing good results despite all possible data imperfections inherent for discussion forums. This work focuses on the text classification task: using a set of documents with assigned class labels to create a model that can predict classes of arbitrary documents (Manning *et al*., 2008). It can be useful for recognizing discussion topics and measuring their publication intensity.

There is a huge amount of prior work on text classification (e.g., McCallum and Nigam, 1998; Joachims, 1998; Radovanović and Ivanović, 2008; Rousseau *et al*., 2015; Dařena and Žižka, 2017). This article is supposed to extend the current state of knowledge by adding some novel contributions, summarized below.

1. Besides the most common bag of words text representation, the more refined global vectors representation (GloVe) (Pennington *et al*., 2014) based on word embeddings is additionally employed, which makes it easy to control the dimensionality and apply arbitrary general-purpose classification algorithms. While text representations based on word embeddings are becoming popular (Goldberg and Levy, 2014; Lau and Baldwin, 2016), there are still not so many demonstrations of their utility for text classification, particularly for the GloVe representation. According to Pennington *et al.* (2014) it outperformed the more widely known *word2vec* algorithm of Mikolov *et al.* (2013a) in several word analogy and similarity tasks, and it appears the most promising representative of the family of word embeddings.

2. Using the GloVe representation makes it possible to examine the performance of diverse general-purpose classification algorithms, some of which are not very commonly applied to text data. In particular, there do not appear to be any previously published studies examining the performance of the SVM and random forest models with the GloVe representation. Not being limited to text data, these algorithms could easily use additional attributes not based on text content, e.g., derived from publication time, authorship, or presence of multimedia, which may be desirable for some applications.

3. The effects of mutual information-based term selection is investigated to verify whether it can make the high dimensionality of the bag of words representation comparable to the reduced dimensionality of the GloVe representation without a significant loss of predictive performance, and whether it can yield any improvements for the latter.

4. The discussion forum used for this work is in Polish, which has not been the subject of so many text mining studies and makes some of text preprocessing operations more complex than for an English text. There are no previously published demonstrations of Polish text classification with the GloVe representation.

5. The discussion forum used for this work is devoted to psychoactive substances received from home-grown plants, such as hashish or marijuana. The obtained results suggest the possibility of applying a text classification approach to monitoring the risk of drug-related crime.

1.2. Data. The experimental demonstrations presented in this article use a collection of posts retrieved from a Polish discussion forum devoted to psychoactive substances received from home-grown plants, such as hashish or marijuana. All posts published between January 2014 and September 2016 are used, partitioned into two disjoint subsets as follows:

training set: from January 2014 do June 2015 (used for deriving text representation and for model creation),

test set: from July 2015 to August 2016 (used for model evaluation),

which contain 122463 and 80576 posts, respectively.

2. Text representation

The first issue to be resolved when applying modeling algorithms from the fields of machine learning and statistics to text data is transforming the analyzed document corpus into a representation that can be handled by these algorithms. This is usually a vector representation in which each document is assigned values of a fixed, common set of attributes (Dumais *et al*., 1998; Aggarwal and Zhai, 2012; Szymański, 2014). A document is then represented by a vector of its attribute values, with the number of vector elements being the same for all documents. More complex non-vector representations are occasionally applied that may better preserve the semantic structure of the analyzed text (Rousseau *et al*., 2015), but they require dedicated modeling algorithms, which are beyond the scope of this article.

2.1. Bag of words. The simplest vector text representation that remains the most common in text mining applications is the *bag of words* (BOW) representation (McCallum and Nigam, 1998; Joachims, 1998; Aggarwal and Zhai, 2012; Szymański, 2014), in which attributes directly correspond to words or, in a slightly more general setting, n-*grams*—word sequences of length n (usually $n \leq 3$). Words or n-grams used for this representation are called *terms*. All occurrences of the same term in a document are treated in the same way, regardless of their position and surrounding terms, which makes this representation perfectly order- and context-insensitive (apart from partial sensitivity resulting from using n-grams for $n > 1$). This clearly limits its capability of preserving semantic information, offering simplicity, ease of use, and direct attribute interpretability in return.

The BOW representation can be used in different variations, depending on how attributes corresponding to terms are exactly defined (Manning *et al.*, 2008). The most common *term frequency* variation is used for this work, with the value of attribute $a_t(x)$ for term t and document x defined as

$$a_t(x) = \mathrm{TF}_t(x), \tag{1}$$

where $\mathrm{TF}_t(x)$ is the number of occurrences of term t in document x. The matrix of all such attribute values for a particular set of documents is referred to as the *document-term matrix*. It is essential to ensure that the set of attributes of the bag of words representation is fixed and common for all documents. This can be achieved by using a fixed common vocabulary. While a modified TF-IDF variant, weighting terms based on their specificity represented by *inverse document frequency*, is useful for some applications (Manning *et al.*, 2008), it did not bring any improvements for text classification algorithms used in this work.

The order- and context-insensitivity of the bag of words representation are likely to cause a loss of semantic information contained in analyzed documents. This is only partially improved by using n-grams for $n > 1$, which may cause other problems: occurrence frequencies for 2-grams or 3-grams often become extremely small and most of them occur only in few documents, whereas their total number may be very large, resulting in a very high-dimensional and extremely sparse representation. Despite its limitations, the BOW representation remains effective in many applications, in which raw term occurrence statistics constitute a sufficient description of text content. Attribute values are simple and efficient to calculate, although using them may not always be so efficient unless some form of dimensionality reduction is applied. The dimensionality of the bag of words representation clearly depends on the size of the analyzed text corpus. This is confirmed by empirical observations, indicating a sublinear functional relationship between the number of distinct words in a collection of documents and the total length of these documents, referred to as Heaps's law (Heaps, 1978). In practice, after some basic frequency-based term filtering, it is typically between several hundred and several thousand (or more if n-grams with $n > 1$ are used).

2.2. Global vectors. Limitations of the bag of words representation can be overcome by more refined context-sensitive approaches. These include representations based on *word embeddings*, such as *word2vec* (Mikolov *et al.*, 2013a) which is obtained by training a two-layer neural network to associate words with their occurrence contexts. This produces word vectors (vector representations of words), but a related *doc2vec* algorithm can be used to obtain document vectors as well (Le and Mikolov, 2014).

This article uses another representation based on word embeddings, known as *global vectors*, or *GloVe* (Pennington *et al.*, 2014). It can be considered a simpler and potentially more efficient alternative to *word2vec* that achieves similar effects using a different approach, based on term co-occurrence statistics. In the experiments reported by Pennington *et al.* (2014) it outperformed *word2vec* with respect to both accuracy and computational expense in several word analogy and similarity tasks. It is also easier to apply due to a smaller number of adjustable parameters and less sensitivity to the exact parameter setup.

The algorithm uses a text corpus to create a co-occurrence matrix Γ such that $\Gamma[t, \kappa]$ is the frequency of term t appearing in the context of term κ, for some pre-established context window size. The matrix can be used to estimate the occurrence probability of term t in the context of term κ as follows:

$$P(t|\kappa) = \frac{\Gamma[t, \kappa]}{\sum_{t'} \Gamma[t', \kappa]}. \tag{2}$$

The matrix is then factorized using stochastic gradient descent to determine term vectors such that the dot product of vectors for any two terms t and κ approximates $\log P(t|\kappa)$. The dimensionality of this representation can be controlled and is typically orders of magnitude less than the dimensionality of the bag of words representation.

More precisely, the GloVe algorithm assigns two vectors to each term t: the word vector $\mathbf{w}_t$ and the context vector $\tilde{\mathbf{w}}_t$, such that the dot product $\mathbf{w}_t \bullet \tilde{\mathbf{w}}_\kappa$ approximates $\log P(t|\kappa)$ or, equivalently,

$$\mathbf{w}_t \bullet \tilde{\mathbf{w}}_\kappa + b_t + \tilde{b}_\kappa \approx \log \Gamma[t, \kappa], \tag{3}$$

where b_t and $\tilde{b}_\kappa$ are bias terms for $\mathbf{w}_t$ and $\tilde{\mathbf{w}}_\kappa$, respectively. Word and context vectors are estimated using the *Adagrad* algorithm (Duchi *et al.*, 2011), which is a variant of stochastic gradient descent with step-size adaptation, applied to the following weighted least squares objective:

$$\sum_{t_1 . t_2} f(\Gamma[t_1, t_2])\big(\mathbf{w}_{t_1} \bullet \tilde{\mathbf{w}}_{t_2} + b_{t_1} + \tilde{b}_{t_2} - \log \Gamma[t_1, t_2]\big)^2. \quad (4)$$

The weighting function f supposed to assign higher weights to more frequent term co-occurrences (but without overweighting the most frequent ones) is defined as

$$f(v) = \begin{cases} \left(\dfrac{v}{v_{\max}}\right)^{\alpha} & \text{if } v < v_{\max}, \\ 1 & \text{otherwise,} \end{cases} \quad (5)$$

where $v_{\max}$ is the cutoff value for co-occurrence counts and the α exponent controls the sensitivity of weights to increased co-occurrence counts.

It can be easily seen that the GloVe representation captures some semantic information based on word context. Indeed, the logarithm of the occurrence probability ratio of two terms t_1 and t_2 in the context of the same term κ, which can be considered a measure of their semantic divergence, relates to the difference of their word vectors as follows:

$$\begin{aligned} \log \frac{P(t_1|\kappa)}{P(t_2|\kappa)} &= \log P(t_1|\kappa) - \log P(t_2|\kappa) \\ &= (\mathbf{w}_{t_1} - \mathbf{w}_{t_2}) \bullet \tilde{\mathbf{w}}_{\kappa}. \end{aligned} \quad (6)$$

For symmetric co-occurrence matrices, word vectors $\mathbf{w}$ and context vectors $\tilde{\mathbf{w}}$ are roughly the same, with minor differences only due to random initialization. This is the case for symmetric context windows, extending to both sides of the target term. Asymmetric context windows, extending only to the left, can sometimes work better, but symmetric context windows are used for this work, adequate for the free word order of the Polish language. Following the recommendation of Pennington *et al.* (2014), the sums of word and context vectors are used as ultimate term vectors to reduce noise and the risk of overfitting:

$$\mathbf{a}(t) = \mathbf{w}_t + \tilde{\mathbf{w}}_t. \quad (7)$$

Word embeddings in general and global vectors in particular can be considered a potentially superior alternative to other vector text representations which achieve reduced dimensionality by transforming the bag of words document-term matrix. These include latent semantic analysis (LSA) (Dumais, 2005; Moldovan *et al.*, 2005; Aswani Kumar and Srinivas, 2006) and latent Dirichlet allocation (LDA) (Blei *et al.*, 2003). Word embedding methods are less computationally demanding and believed to be more effective in identifying semantic relationships than those earlier approaches (Mikolov *et al.*, 2013b).

Like *word2vec* (and unlike *doc2vec*), the GloVe representation produces term vectors, rather than document vectors, which are actually needed to create document classification models. One simple and possibly imperfect approach to obtain GloVe document vectors is to perform weighted summation of term vectors for all terms occurring in a document, with term frequencies used as weights. The vector for document x would be then calculated as

$$\mathbf{a}(x) = \sum_{t \in x} \mathbf{a}(t) \mathrm{TF}_t(x). \quad (8)$$

This solution, which can be considered a simple example of compositional semantics models (Mitchell and Lapata, 2010; Yessenalina and Cardie, 2011), is adopted for this work.

3. Text classification

Text classification, also referred to as text categorization, consists in using a set of class-labeled documents from some domain to create a model providing class predictions for arbitrary documents from the same domain (Sebastiani, 2002). According to traditional machine learning terminology, the unknown true mapping of documents to classes is referred to as the *concept* and can be considered a function $c : X \to \mathcal{C}$, where X denotes the domain and $\mathcal{C}$ is a finite set of classes. True classes are known for a subset of the domain $T \subset X$ called the training set. A model created based on the training set can be similarly considered a function $h : X \to \mathcal{C}$, supposed to approximate the concept c. A probabilistic classification scenario is also often adopted in which the model is supposed to produce predictions of class probabilities rather than class labels.

3.1. Naive Bayes.

A particularly simple conceptually and implementationally as well as computationally efficient classification algorithm that is often successfully applied to text classification is the *naive Bayes classifier*. It predicts the class probability given attribute values,

$$P(c = d \mid a_1 = v_1, a_2 = v_2, \ldots, a_n = v_n), \quad (9)$$

which is referred to as the posterior probability of class d for a vector of attribute values $v_1, v_2, \ldots, v_n$.

3.1.1. Basic general-purpose algorithm.

According to the Bayes theorem (Bayes, 1763) the posterior class probability can be calculated as follows:

$$\frac{P(c = d) P(a_1 = v_1, \ldots, a_n = v_n \mid c = d)}{P(a_1 = v_1, \ldots, a_n = v_n)}. \quad (10)$$

The denominator can be considered a class-independent normalizing constant. The prior class probability appearing in the numerator can be directly estimated from the training set. Only the conditional joint probability of attribute values given the class deserves more interest, since it cannot be reliably estimated directly for realistically-sized datasets. This is where the "naivety" of the algorithms comes into place, which consists in adopting the usually unsatisfied conditional attribute independence assumption given the class and calculating the conditional joint probability as

$$\prod_{i=1}^{n} P(a_i = v_i | c = d). \tag{11}$$

Conditional attribute value probabilities given the class for single attributes $P(a_i = v_i|c = d)$ can be estimated from the training set directly, often using Laplace smoothing or Cestnik m-estimation (Cestnik, 1990) to avoid issues related to zero probabilities. Numeric attributes are either discretized or assumed to be distributed normally, with density function values used instead of probabilities in the calculation.

The "naive" independence assumption results in calculated probabilities being incorrect, but it does not necessarily limit the predictive utility of the algorithm (Domingos and Pazzani, 1997; Hand and Yu, 2001). It remains particularly popular in text mining applications, where its capability of incorporating numerous attributes to class predictions without the risk of overfitting is particularly desirable.

3.1.2. Text-specific algorithm.

Technically, the naive Bayes classifier can be directly applied to text data with any vector representation. However, for the term frequency bag of words representation the calculation of the conditional attribute value probability given the class can be modified. For this representation the value of attribute a_t corresponding to term t is the occurrence count of the term in the document. Attributes are therefore numeric and the standard general purpose algorithm would handle them either by discretization or by using normal density functions. A much better alternative, however, is to use the multinomial distribution (McCallum and Nigam, 1998; Lewis, 1998):

$$\begin{aligned} P(a_{t_1} = v_1, v_2, \dots, a_{t_n} = v_n | c = d) \\ = \Big(\sum_{i=1}^{n} v_{t_i}\Big)! \prod_{i=1}^{n} \frac{P_1(t_i|d)^{v_i}}{v_i!}, \end{aligned} \tag{12}$$

where $V = \{t_1, t_2, \dots, t_n\}$ is the vocabulary and $v_1, v_2, \dots, v_n$ are the corresponding occurrence counts of vocabulary terms in the document being classified (playing the role of attribute values). This represents the probability of each term t_i appearing v_i times in a document of class d, calculated based on the probability of a single occurrence of term t_i:

$$P_1(t_i|d) = \frac{\sum_{x \in T_{c=d}} a_{t_i}(x)}{\sum_{x \in T_{c=d}} \sum_{j=1}^{n} a_{t_j}(x)} \tag{13}$$

for each $i = 1, 2, \dots, n$. The algorithm applying this form of probability calculations is known as the *multinomial naive Bayes classifier*.

3.2. Logistic regression.

Logistic regression is an instantiation of generalized linear models which adopts a composite model representation function, with an inner linear model and an outer logit transformation (Hilbe, 2009). The inner linear model is defined as follows:

$$g(x) = \mathbf{w} \bullet \mathbf{a}(x) + b, \tag{14}$$

where $\bullet$ is the dot product operator, $\mathbf{w}$ is a vector of model parameters $w_1, w_2, \dots, w_n$, and b is an additional intercept parameter. The outer logit transformation produces probability predictions:

$$P(1|x) = \frac{e^{g(x)}}{e^{g(x)} + 1}. \tag{15}$$

This assumes a binary probabilistic classification scenario with the $\{0, 1\}$ set of classes, for which the model predicts the probability of class 1 and the probability of class 0 can be obtained as its 1's complement. Training a logistic regression model consists in finding parameters $w_1, w_2, \dots, w_n$ and b which maximize the log-likelihood of training set classes:

$$\sum_{x \in T} \big(c(x) \ln P(1|x) + (1 - c(x)) \ln(1 - P(1|x))\big). \tag{16}$$

Due to the probabilistic objective function used for parameter estimation, logistic regression can generate well-calibrated probability predictions and is often the classification algorithm of choice when this is required. It is easy to apply and not overly prone to overfitting unless used for high-dimensional data, which is unfortunately often the case in text classification applications, at least with the bag of words representation. With that being said, the algorithm can be used with text data in arbitrary vector representations without any adjustments.

3.3. Support vector machines.

Support vector machines (SVMs), which often belong to the most effective general-purpose classification algorithms (e.g., Hamel, 2009; Cichosz, 2015; Bilski and Wojciechowski, 2016), can be viewed as a considerably strengthened version of a basic linear-threshold classifier with the following enhancements (Cortes and Vapnik, 1995; Platt, 1998):

margin maximization: the location of the decision boundary (separating hyperplane) is optimized with respect to the classification margin;

soft margin: incorrectly separated instances are permitted;

kernel trick: complex nonlinear relationships can be represented by representation transformation using kernel functions.

The SVM algorithm assumes a binary classification scenario with two classes, denoted by -1 and 1 for convenience, since using class labels as numbers simplifies the form of the objective function and constraints of the underlying optimization task. Class predictions are generated using a standard linear-threshold rule:

$$h(x) = \mathrm{sgn}(\mathbf{w} \bullet \mathbf{a}(x) + b), \tag{17}$$

with $\mathbf{w}$ and b being model parameters, obtained by solving the following quadratic programming problem:

minimize

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_x \xi_x \tag{18}$$

subject to

$$(\forall x \in T) \quad c(x)(\mathbf{w} \bullet \mathbf{a}(x) + b) \geq 1 - \xi_x, \tag{19}$$

$$(\forall x \in T) \quad \xi_x \geq 0. \tag{20}$$

The first term of the objective function is responsible for classification margin maximization, i.e., placing the decision boundary so as to maximize the distance from the closest correctly separated instances. The second term represents penalty for constraint violations, whose magnitude is controlled by the cost parameter C. Constraint violations are permitted by introducing slack variables ξ_x for each training instance x.

The presented primal form of the optimization problem permits easy interpretation, but transforming it to a dual form by applying Lagrange multipliers provides substantial advantages (Cristianini and Shawe-Taylor, 2000; Schölkopf and Smola, 2001). A particularly important property of the dual form is that it only uses attribute value vectors within dot products, both during model creation and during prediction. This makes it possible to apply the kernel trick—an implicit representation transformation. Instead of dot products of the original attribute value vectors, kernel function values $K(x_1, x_2)$ are used, which represent dot products of enhanced attribute value vectors $\mathbf{a}'(x_1) \bullet \mathbf{a}'(x_2)$. This achieves the effect of transforming attribute value vectors without actually applying the transformation. The special case in which $K(x_1, x_2) = \mathbf{a}(x_1) \bullet \mathbf{a}(x_2)$, referred to as the *linear kernel*, does not transform the original representation. The most popular type of nonlinear kernel functions is the *radial kernel*:

$$K(x_1, x_2) = e^{-\gamma\|\mathbf{a}(x_1) - \mathbf{a}(x_2)\|^2}, \tag{21}$$

where $\gamma > 0$ is an adjustable parameter.

Instead of binary linear-threshold SVM predictions it may be often more convenient to use probabilistic predictions, like for naive Bayes and logistic regression. This is possible by applying a logistic transformation to the signed distance of classified instances from the decision boundary, with parameters adjusted for maximum likelihood (Platt, 2000).

A noteworthy property of the SVM is the insensitivity of model quality to data dimensionality, which—unlike for many other algorithms—does not increase the risk of overfitting because model complexity is related to the number of instances close to the decision boundary rather than to the number of attributes. This is definitely a potential advantage in text classification applications, where high dimensionality is to be expected, at least with the bag of words representation (Joachims, 1998; Joachims, 2002). That being said, the SVM algorithm is not necessarily quick and easy to be successfully applied, as it tends to be sensitive to parameter settings (in particular, the cost parameter, the kernel type, and kernel function parameters) and is computationally expensive for large datasets.

3.4. Random forest.

Random forests belong to popular ensemble modeling algorithms which achieve improved predictive performance by combining multiple diverse models for the same domain (Dietterich, 2000). They usually yield excellent predictive performance with little or no need for parameter tuning and low risk of overfitting (e.g., Cichosz, 2015; Siwek and Osowski, 2016). A random forest is an ensemble model represented by a set of unpruned decision trees, grown based on multiple bootstrap samples drawn with replacement from the training set, with randomized split selection (Breiman, 2001). It can be considered an enhanced form of bagging (Breiman, 1996), which additionally stimulates the diversity of individual models in the ensemble by randomizing the decision tree growing algorithm used to create them.

Random forest growing consists in growing multiple decision trees, each based on a bootstrap sample from the training set (usually of the same size as the original training set), by using an essentially standard decision tree growing algorithm (Breiman *et al.*, 1984; Quinlan, 1986). Since the expected improvement of the resulting model ensemble over a single model is contingent upon sufficient diversity of the individual models in the ensemble (Breiman, 1996; Dietterich, 2000), the following modifications are applied to stimulate the diversity of decision trees in a random forest:

- large maximally fitted trees are grown (with splitting continued until reaching a uniform class, exhausting the set of instances, or exhausting the set of possible splits);

- whenever a split has to be selected for a tree node, a small subset of available attributes is selected randomly and only those attributes are considered for candidate splits.

Individual trees built that way are likely to be overfitted, but nevertheless no pruning is applied to them. With the random internal attribute selection this gives them many opportunities to differ, though, and their individual overfitting is effectively canceled out if they are used as an ensemble.

Random forest prediction is achieved by simple unweighted voting of individual trees from the model. Vote distribution can be also used to obtain class probability predictions. With sufficiently many diversified trees (typically hundreds), this simple voting mechanism usually makes random forests extremely accurate and resistant to overfitting. As a matter of fact, in many cases they belong to the most accurate classification models that can be achieved.

Given the advantages of random forests, it may be surprising to see not so many examples of their application to text classification. This may be related to concerns about the suitability of the algorithm for the bag of words representation, which is usually extremely high-dimensional and sparse, with possibly many irrelevant terms. That being said, some successful results have been reported (Rios and Zha, 2004; Koprinska *et al*., 2007; Xue and Li, 2015) and a modified versions of the algorithm adjusted to text data have been proposed (Xu *et al*., 2012; Wu *et al*., 2014).

3.5. Term selection for classification.

One way of reducing the dimensionality of the bag of words representation is the selection of the most predictively useful terms for the classification task being considered. In principle, this could be performed using any general-purpose attribute selection method (Liu and Motoda, 1998; Guyon and Elisseeff, 2003; Liu *et al*., 2010; Cichosz, 2015), but the high dimensionality of the bag of words representation makes computationally intensive attribute selection algorithms hardly applicable and favors relatively simple techniques. One particularly popular approach is to rank terms with respect to the mutual information between term occurrence and class labels (Yang and Pedersen, 1997; Forman, 2003).

The mutual information for term t and concept c, in this context also referred to as the information gain of term t with respect to concept c, can be calculated on a document set D as follows:

$$\begin{aligned}I(t,c) = &\sum_{d\in\mathcal{C}} P(t,c=d)\log\frac{P(t,c=d)}{P(t)P(c=d)}\\ &+\sum_{d\in\mathcal{C}} P(\neg t,c=d)\log\frac{P(\neg t,c=d)}{P(\neg t)P(c=d)},\end{aligned} \tag{22}$$

where

- $P(t,c=d)$ and $P(\neg t,c=d)$ are the probabilities that a randomly chosen document respectively contains and does not contain term t, and is of class d, estimated on the training set:

$$P(t,c=d) = \frac{|T_t,c=d|}{|T|}, \tag{23}$$

$$P(\neg t,c=d) = \frac{|T_{\neg t},c=d|}{|T|}, \tag{24}$$

where $T_{t,c=d}$ and $T_{\neg t,c=d}$ are the subsets of training documents of class d respectively containing and not containing term t;

- $P(t)$ and $P(\neg t)$ are the probabilities of term t respectively appearing and not appearing in a randomly chosen document, estimated on the training set:

$$P(t) = \frac{|T_t|}{|T|}, \tag{25}$$

$$P(\neg t) = \frac{|T_{\neg t}|}{|T|}, \tag{26}$$

where T_t and $T_{\neg t}$ are the subsets of training documents respectively containing and not containing term t.

4. Experimental study

Algorithms presented in the previous section were applied to the classification of discussion forum posts, using the two previously described text representation methods. The experiments are supposed to compare the utility of particular algorithms and text representations, as well as investigate the effects of term selection.

4.1. Experiment setup.

The process used to transform discussion forum posts to vector representations as well as to create and evaluate classification models is summarized below.

4.1.1. Text representation.

The bag of words and GloVe document representations for experiments presented in this article were created using the `text2vec` R package (Selivanov, 2016). The same text preprocessing was applied for the two representations to identify the common vocabulary:

- removing numbers;
- converting to lowercase;
- spelling correction and stemming using *Hunspell* via the `hunspell` R package interface (Oooms, 2016) with the Polish *LibreOffice* dictionary;
- removing Polish stop words;
- filtering terms according to the following criteria:
  - no less than 100 and no more than 10000 occurrences in the training set,
  - occurring in no less than 0.2% and no more than 80% documents from the training set.

This produced a vocabulary of 2155 words, out of more than $300,000$ distinct words appearing in original unprocessed training documents. Only documents containing at least a single vocabulary term were left in the training and test sets, which reduced their size to 119084 and 77798 documents, respectively.

The Polish language poses some extra challenges for spelling correction due the the presence of diacritical characters in the alphabet. It is not uncommon to see posts with some or all of diacritics missing, which is likely to mislead spelling correction. Occasional occurrences of English words may be confused with misspelled Polish words. Dictionary-based stemming is also less reliable than for English due to much more complex inflection. More refined spell checkers and morphological analyzers dedicated to the Polish language, word sense disambiguation, as well as detecting foreign language terms and phrases, might improve the results to some extent. Examining these possibilities of enhanced natural language processing is postponed for future work, though. Only one simple tweak was applied to improve the performance of *Hunspell* spelling correction—enforcing the preference for replacement words that only add missing diacritics.

The attributes of the BOW representation for the training and test sets were obtained as the corresponding term frequencies for all vocabulary terms.

Unlike for the English language, there are no Polish GloVe term vectors available that would be pre-trained on large text corpora, such as Wikipedia articles or Internet site contents collected by web crawlers. The term vectors of the GloVe representation were therefore determined based on the term co-occurrence matrix for the training set, using the same vocabulary, with the following parameter settings:

context window size: 5,

word vector dimensionality: 50,

weighting function cutoff $v_{\max}$*:* 10,

weighting function exponent α*:* $\frac{3}{4}$,

maximum gradient descent step size: 0.1.

They are mostly based on the results and suggestions of Pennington *et al.* (2014), who found relatively small improvement for context windows longer than 5, suggested low sensitivity to the setting of the weighting function cutoff level, and recommended a weighting function exponent of $3/4$. Only minimal parameter space search was performed, limited to the setting of word vector dimensionality, but no improvement was found for values above 50.

The training set term vectors were then used to obtain document vectors for the training and test sets by weighted summation, as described before, using term frequencies for the training and test sets, respectively, as weights.

It is worthwhile to stress that "attribute definitions" for both the bag of words and GloVe representations (i.e., the vocabulary and term vectors) are entirely determined using the training set only and then applied to calculate attribute values for the training and the test set. This guarantees there is no optimistic bias in subsequent model evaluation that could result from using the test set for any processing with impact on model creation.

4.1.2. Text classification. The most interesting setup for a discussion forum post-classification task would be to have human-assigned class labels for the training set, representing some concepts of interest (e.g., related or unrelated to a specific topic, interesting vs. uninteresting, spam vs. ham, suspicious vs. innocent, etc.). In the lack of such human-assigned labels, the discussion forum structure will be used as a substitute. The forum used in this article has the following set of top-level branches (translated to English):

- *Marijuana*,
- *Cannabis Indica Strains*,
- *Growing*,
- *Users' Plants*,
- *Free Seeds*,
- *Supplies*,
- *Other*.

For each of them a binary classification task is defined, with posts in a given branch considered positive and all other posts considered negative. There is a subset of posts not assigned to any of these branches (about 30% of the training set and about 15% of the test set), which are considered negative for all branches. Table 1 presents the training set class distributions. For the most part of this

Table 1. Class distribution for the top-level forum branches in the training set.

Branch	Class distribution	
	negative	positive
Marijuana	0.92	0.08
Cannabis Indica Strains	0.97	0.03
Growing	0.63	0.37
Users' Plants	0.96	0.04
Free Seeds	0.90	0.10
Supplies	0.99	0.01
Other	0.91	0.09

experimental study, we focus on the *Marijuana* branch, believed to be most directly associated with drug-related crime.

The most common classification quality measures such as the misclassification error or classification accuracy are not very useful whenever classes are unbalanced or likely to have different predictability. This is why a quality measure sensitive to misclassification distribution is required. In the experiments reported in this section, classification quality is visualized using ROC curves, presenting possible tradeoff points between the true positive rate and the false positive rate (Egan, 1975; Fawcett, 2006), and summarized using the area under the ROC curve (AUC).

The following algorithm and text representation configurations are used in the experiments:

MNB-BOW: the multinomial naive Bayes classifier with the term frequency bag of words representation—a custom implementation in R developed for this article;

NB-BOW, NB-GloVe: the standard naive Bayes classifier with the term frequency bag of words and GloVe representations using normal density functions for term frequency attributes—the implementation provided by the `e1071` R package (Meyer *et al.*, 2015);

LR-BOW, LR-GloVe: logistic regression with the term frequency bag of words and GloVe representations—the implementation provided by the standard `glm` R function (R Development Core Team, 2016);

SVM-BOW, SVM-GloVe: the SVM algorithm with the term frequency bag of words and GloVe representations—the implementation provided by the `e1071` R package (Meyer *et al.*, 2015);

RF-BOW, RF-GloVe: the random forest algorithm with the term frequency bag of words and GloVe representations—the implementation provided by the `randomForest` R package (Liaw and Wiener, 2002).

For the multinomial naive Bayes algorithm, Laplace smoothing was applied in term occurrence probability estimates. The standard naive Bayes algorithm has no parameters to set. For the logistic regression and SVM algorithms, parameters controlling the underlying optimization process were left at default values. The SVM parameters specifying the optimization problem were set as follows:

constraint violation cost C: 1,

kernel type: radial,

kernel parameter γ: the inverse of the input dimensionality,

class weights: 10 for class 1 (the minority class), 1 for class 0.

Except for the latter, these are default settings. The linear kernel achieved slightly worse results. Limited search indicated no improvement resulting from modifying the cost parameter or kernel parameters.

For the random forest algorithm, the following setup was used:

tree count: 500,

random attribute subset size: the square root of the total number of available attributes,

stratified bootstrap sample size: the number of instances of class 1 (the minority class).

Except for the latter, these are default settings. This is a safe choice given the relatively low sensitivity of the algorithm to parameter values.

It is worthwhile to notice that the parameter setups for the SVM and the random forest algorithm include settings responsible for properly handling unbalanced classes (ensuring sufficient sensitivity to the minority class). This is achieved by specifying class weights for SVM (assigning a higher weight to the minority class when calculating the constraint violation penalty term in the optimization objective) and by specifying stratified bootstrap sample size for the random forest algorithm (drawing the maximum possible number of minority class instances and the same number of majority class instances). These settings were verified to indeed improve model quality. No form of class rebalancing is necessary for the naive Bayes and logistic regression algorithms, since any class weights or priors would only shift the default class probability cutoff point used for predicted class label assignment. This would serve no useful purpose given the fact that the ROC analysis used for predictive performance evaluation is based on predicted class probabilities instead of class labels anyway.

4.2. Results. The presented results make it possible to assess the predictability of forum branches, compare different algorithms and text representations, and verify the effects of term selection.

4.2.1. Predictability of forum branches. The multinomial naive Bayes algorithm was applied to all top-level forum branches. The obtained ROC curves are presented in Fig. 1. The level of prediction quality varies across forum branches, but with nearly all AUC values above 0.75 and some reaching 0.9 can be definitely considered sufficient for most common practical needs. The *Supplies* branch turned out particularly easy to recognize, with the *Free Seeds*, *Cannabis Indica Strains*, and *Marijuana* branches not far behind. The hard predictability observed for the *Users' Plants* branch may be due to the limited textual contents of posts, which contain mostly photos.

4.2.2. Algorithm and representation comparison. The remaining experiments in this section focus on the *Marijuana* class. Figure 2 presents the ROC curves for all the previously listed algorithms with the BOW representation. It can be immediately seen that, while the multinomial naive Bayes classifier achieves the best predictive performance, the logistic regression, SVM, and random forest algorithms are not far behind. They all managed to successfully cope with the high-dimensional sparse bag of words representation. What is particularly striking is the poor performance of the standard naive Bayes algorithm, handling term frequency attributes as regular numeric attributes using normal density functions, which turned clearly worst in the competition.

The ROC curves obtained with the GloVe representation are presented in Fig. 3. When using word embeddings, the SVM and random forest algorithms reach roughly the same performance level that was observed for the multinomial naive Bayes with the BOW representation and can be considered a viable alternative for the latter. The considerably reduced dimensionality and context-sensitivity of document vectors make it possible to successfully discriminate between classes, even if the performance level of multinomial naive Bayes remains unbeaten. This may be actually the best prediction quality possible with the data anyway. The logistic regression algorithm achieves the same prediction quality as with bag of words, and the standard naive Bayes algorithm becomes even worse than before. While the experiments only investigate classification quality and not computational expense, it is worthwhile to mention that the GloVe representation reduced the overall computation time of model creation and evaluation in comparison to bag of words by a factor of more than 12 for the SVM and more than 350 for the random forest.

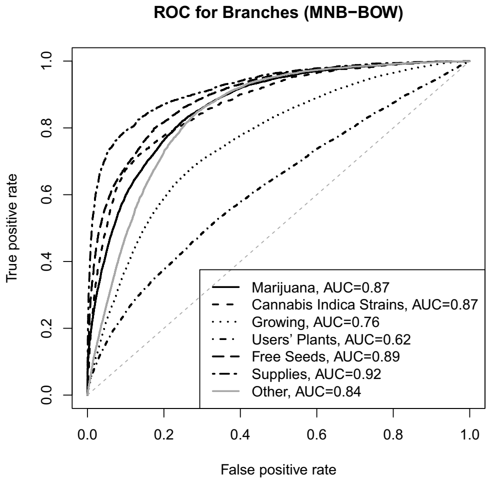

Fig. 1. Multinomial naive Bayes ROC curves for forum branches.

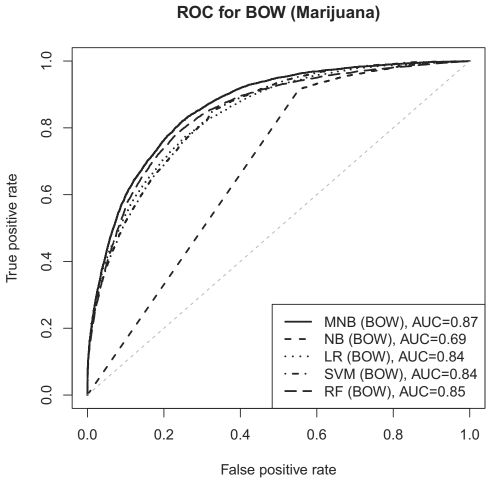

Fig. 2. ROC curves of all algorithms using the bag of words representation for the Marijuana branch.

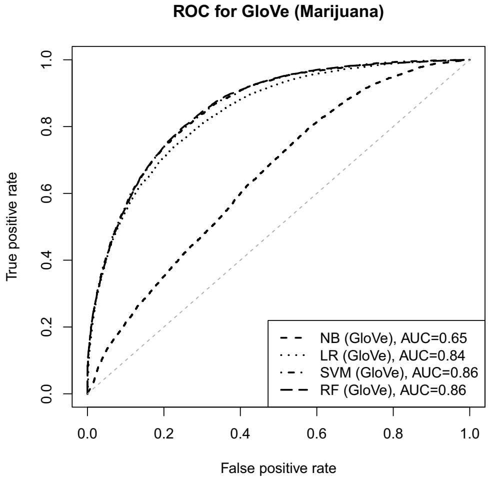

Fig. 3. ROC curves of all algorithms using the GloVe representation for the Marijuana branch.

4.2.3. Term selection. To examine the effect of vocabulary size on the predictive performance using the bag of words and GloVe representations, term selection was performed on the training set, using the mutual information criterion to select subsets of 50, 100, 500, and 1000 top terms. In the case of the bag of words representation, only attributes corresponding to selected terms were used. For the GloVe representation, term selection was applied in two variants:

before word embedding: the co-occurrence matrix and term vectors are determined using the reduced vocabulary with selected terms only, and then term vectors corresponding to these terms are used to calculate document vectors;

after word embedding: the co-occurrence matrix and term vectors are determined using the original vocabulary, and then only word vectors corresponding to selected terms are used to calculate document vectors.

For each of the two representations the best algorithms identified previously were used, i.e., the multinomial naive Bayes and random forest algorithms, respectively, to see whether and how model quality changes with the term selection applied. The ROC curves are presented in Fig. 4.

Moderate term selection turned out not to degrade the performance of the multinomial naive Bayes classifier, but did not lead to any improvement, either. Using top 1000 or 500 most useful terms, the algorithm delivers nearly the same classification quality as with the full vocabulary of more than 2000 terms. More aggressive term selection degrades its predictive performance, moderately for 200 terms and more severely for 100 and 50 terms. In the case of the GloVe representation, any term selection appears to be harmful, applied either before or after word embedding. It is noteworthy, though, that this representation makes it possible to reach about the same performance level as multinomial naive Bayes with 500 or more attributes using only 50 attributes.

5. Conclusion

The results presented in this article enhance the state of knowledge about the practical utility of classification algorithms for text data. They also encourage practical applications to discussion forum and other social media mining. That being said, several issues arise that are worth investigating in the future.

5.1. Major findings. The major findings of this work are summarized below. While they are strictly speaking valid only for the particular application domain and dataset used for the reported experiments, at least some

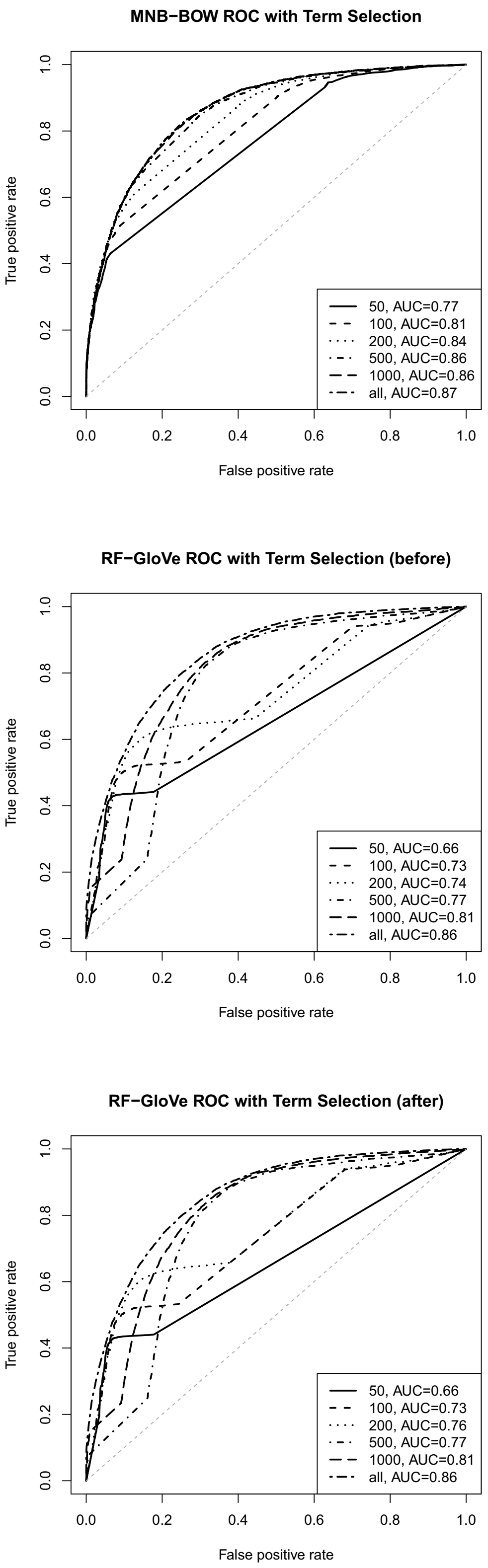

Fig. 4. ROC curves with term selection for the Marijuana branch.

of them are likely to hold for other application domains and datasets.

1. The GloVe text representation based on word embedding makes it possible to identify meaningful relationships between terms occurring in discussion forum posts.

2. Discussion forum posts can be reliably classified to classes corresponding to forum branches based on their textual contents, and a similar level of classification quality may be possible using other human-assigned class labels.

3. While it is hard to beat the predictive performance of the multinomial naive Bayes classifier with the bag of words representation, the SVM and random forest algorithms can reach the same quality level when used with the GloVe representation.

4. The GloVe representation is a more effective method of dimensionality reduction than term selection, making it possible to reach the classification performance level of bag of words with 500 or more terms using as little as 50 attributes.

5. The combination of the GloVe representation and the random forest algorithm appears to be a particularly useful approach to text classification due to its good prediction quality, reduced dimensionality without term selection, low sensitivity to parameter settings, and low risk of overfitting, as well as the capability to directly incorporate additional non-textual attributes.

5.2. Practical utility. The primary potential practical utility of the reported research is related to automatic monitoring of discussion forums and other social media channels. One reasonable usage scenario would be classification to pre-defined topics and measuring publication intensity of particular topics in time. While classification models for this scenario would make their predictions primarily based on post textual content, it may be sometimes desirable or even necessary to include other attributes, characterizing the publication time, the publication method, the author, or the usage of non-textual content (such as mathematical formulae, images, and audio or video clips). This can be easily achieved by combining the GloVe text representation with additional non-text attributes and applying general-purpose classification algorithms. Unlike algorithms dedicated to text and the bag of words representation (such as the multinomial naive Bayes classifier), they can easily and transparently work using both text and non-text attributes.

When applied to a discussion forum devoted to psychoactive substances, document classification algorithms can be useful for monitoring the risk of drug-related crime. This could be achieved by classification to classes corresponding to crime types or risk levels and measuring the publication intensity of posts in these classes. Clearly, human-assigned class labels for training posts would be needed to make this application scenario possible.

5.3. Open issues and future work. The scope and depth of research presented in this article is unavoidably limited. The selection of algorithms used, while believed to be reasonable, could be definitely wider, including both more algorithms dedicated to text and general-purpose ones. No extensive parameter tuning was performed, which might have prevented some algorithms from achieving their top performance. It may be the case, in particular, for the SVM algorithm, known to often be sensitive to parameter settings. More importantly, the effects of varying the settings of the GloVe algorithm, such as context window size and word vector dimensionality, were not investigated. While preliminary runs with other settings were performed and found not to yield any improvements, this is far from being a systematic study. Exploring the performance of other algorithms and parameter setups may be therefore one direction for future research, not necessarily very exciting, but definitely useful.

Text representation is at the heart of the case study contributed by this article. It would be therefore interesting to thoroughly examine possible enhancements to the process used to derive both the bag of words and GloVe representations used for this work. These may include more refined natural language processing techniques used for foreign phrase detection, spelling correction, and stemming, a more adequate list of stop words, incorporating a thesaurus (both general and domain-specific) to combine synonymous terms, including selected particularly meaningful bigrams or trigrams in the vocabulary, more carefully adjusted term filtering criteria, and better term selection techniques. It may be also worthwhile to consider more appropriate methods of combining GloVe term vectors into document vectors, as well as comparing the utility of the GloVe representation with that of *doc2vec*, latent semantic analysis, latent Dirichlet allocation, and other vector text representations (Szymański, 2014).

Other interesting directions of future work are related to the application domain addressed by this article. They include incorporating additional non-text attributes to forum post representation, which may improve the quality of classification models, and using human-assigned class labels more precisely representing discussion topics.

Acknowledgment

The work described in this article was supported by the National Center for Research and Development under the grant no. DOB-BIO7/05/02/2015. The dataset used for the experiments was made available by the Scott Tiger company.

References

Aggarwal, C.C. and Zhai, C.-X. (Eds.) (2012). *Mining Text Data*, Springer, New York, NY.

Aswani Kumar, C. and Srinivas, S. (2006). Latent semantic indexing using eigenvalue analysis for efficient information retrieval, *International Journal of Applied Mathematics and Computer Science* **16**(4): 551–558.

Bayes, T. (1763). An essay towards solving a problem in the doctrine of chances, *Philosophical Transactions of the Royal Society of London* **53**: 370–418.

Bilski, A. and Wojciechowski, J. (2016). Automatic parametric fault detection in complex analog systems based on a method of minimum node selection, *International Journal of Applied Mathematics and Computer Science* **26**(3): 655–668, DOI: 10.1515/amcs-2016-0045.

Blei, D.M., Ng, A.Y. and Jordan, M.I. (2003). Latent Dirichlet allocation, *Journal of Machine Learning Research* **3**: 993–1022.

Breiman, L. (1996). Bagging predictors, *Machine Learning* **24**(2): 123–140.

Breiman, L. (2001). Random forests, *Machine Learning* **45**(1): 5–32.

Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. (1984). *Classification and Regression Trees*, Chapman and Hall, New York, NY.

Cestnik, B. (1990). Estimating probabilities: A crucial task in machine learning, *Proceedings of the 9th European Conference on Artificial Intelligence (ECAI-90), Stockholm, Sweden*, pp. 147–149.

Cichosz, P. (2015). *Data Mining Algorithms: Explained Using R*, Wiley, Chichester.

Cortes, C. and Vapnik, V.N. (1995). Support-vector networks, *Machine Learning* **20**(3): 273–297.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, New York, NY.

Dařena, F. and Žižka, J. (2017). Ensembles of classifiers for parallel categorization of large number of text documents expressing opinions, *Journal of Applied Economic Sciences* **12**(1): 25–35.

Dietterich, T.G. (2000). Ensemble methods in machine learning, *Proceedings of the 1st International Workshop on Multiple Classifier Systems, Cagliari, Italy*, pp. 1–15.

Domingos, P. and Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss, *Machine Learning* **29**(2–3): 103–137.

Duchi, J., Hazan, E. and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization, *Journal of Machine Learning Research* **12**: 2121–2159.

Dumais, S.T. (2005). Latent semantic analysis, *Annual Review of Information Science and Technology* **38**(1): 188–229.

Dumais, S.T., Platt, J.C., Heckerman, D. and Sahami, M. (1998). Inductive learning algorithms and representations for text categorization, *Proceedings of the 7th International Conference on Information and Knowledge Management (CIKM-98), Bethesda, MD, USA*, pp. 148–155.

Egan, J.P. (1975). *Signal Detection Theory and ROC Analysis*, Academic Press, New York, NY.

Fawcett, T. (2006). An introduction to ROC analysis, *Pattern Recognition Letters* **27**(8): 861–874.

Forman, G. (2003). An extensive empirical study of feature selection measures for text classification, *Journal of Machine Learning Research* **3**: 1289–1305.

Goldberg, Y. and Levy, O. (2014). word2vec Explained: Deriving Mikolov et al.'s negative sampling word-embedding method, *arXiv:* 1402.3722.

Guyon, I.M. and Elisseeff, A. (2003). An introduction to variable and feature selection, *Journal of Machine Learning Research* **3**: 1157–1182.

Hamel, L.H. (2009). *Knowledge Discovery with Support Vector Machines*, Wiley, New York, NY.

Hand, D.J. and Yu, K. (2001). Idiot's Bayes—not so stupid after all?, *International Statistical Review* **69**(3): 385–399.

Heaps, H.S. (1978). *Information Retrieval: Computational and Theoretical Aspects*, Academic Press, New York, NY.

Hilbe, J.M. (2009). *Logistic Regression Models*, Chapman and Hall, New York, NY.

Holtz, P., Kronberger, N. and Wagner, W. (2012). Analyzing Internet forums: A practical guide, *Journal of Media Psychology* **24**(2): 55–66.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features, *Proceedings of the 10th European Conference on Machine Learning (ECML-98), Chemnitz, Germany*, pp. 137–142.

Joachims, T. (2002). *Learning to Classify Text by Support Vector Machines: Methods, Theory, and Algorithms*, Springer, New York, NY.

Koprinska, I., Poon, J., Clark, J. and Chan, J. (2007). Learning to classify e-mail, *Information Sciences: An International Journal* **177**(10): 2167–2187.

Lau, J.H. and Baldwin, T. (2016). An empirical evaluation of doc2vec with practical insights into document embedding generation, *Proceedings of the 1st Workshop on Representation Learning for NLP, Berlin, Germany*, pp. 78–86.

Le, Q.V. and Mikolov, T. (2014). Distributed representations of sentences and documents, *Proceedings of the 31st International Conference on Machine Learning (ICML-14), Beijing, China*, pp. 1188–1196.

Lewis, D.D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval, *Proceedings of the Tenth European Conference on Machine Learning (ECML-98), Chemnitz, Germany*, pp. 4–15.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomForest, *R News* **2**(3): 18–22, `http://CRAN.R-project.org/doc/Rnews/`.

Liu, H. and Motoda, H. (1998). *Feature Selection for Knowledge Discovery and Data Mining*, Springer, New York, NY.

Liu, H., Motoda, H., Setiono, R. and Zhao, Z. (2010). Feature selection: An ever-evolving frontier in data mining, *Proceedings of the 4th Workshop on Feature Selection in Data Mining (FSDM-10), Hyderabad, India*, pp. 4–13.

Lui, A. K.-F., Li, S.C. and Choy, S.O. (2007). An evaluation of automatic text categorization in online discussion analysis, *Proceedings of the 7th IEEE International Conference on Advanced Learning Technologies (ICALT-2007), Niigata, Japan*, pp. 205–209.

Manning, C.D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*, Cambridge University Press, Cambridge.

Marra, R.M., Moore, J.L. and Klimczak, A.K. (2004). Content analysis of online discussion forums: A comparative analysis of protocols, *Educational Technology Research and Development* **52**(2): 23–40.

McCallum, A. and Nigam, K. (1998). A comparison of event models for naive Bayes text classification, *Proceedings of the AAAI/ICML-98 Workshop on Learning for Text Categorization, Madison, WI, USA*, pp. 41–48.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A. and Leisch, F. (2015). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*, R package version 1.6-7, `https://CRAN.R-project.org/package=e1071`.

Mikolov, T., Chen, K., Corrado, G.S. and Dean, J. (2013a). Efficient estimation of word representations in vector space, *arXiv:*1301.3781.

Mikolov, T., Le, Q.V. and Sutskever, I. (2013b). Exploiting similarities among languages for machine translation, *arXiv:*1309.4168.

Mitchell, J. and Lapata, M. (2010). Composition in distributional models of semantics, *Cognitive Science* **34**(8): 1388–1429.

Moldovan, A., Boţ, R.I. and Wanka, G. (2005). Latent semantic indexing for patent documents, *International Journal of Applied Mathematics and Computer Science* **15**(4): 551–560.

Ooms, J. (2016). *hunspell: Morphological Analysis and Spell Checker for R*, R package version 2.3, `https://CRAN.R-project.org/package=hunspell`.

Pennington, J., Socher, R. and Manning, C.D. (2014). GloVe: Global vectors for word representation, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP-14), Doha, Qatar*, pp. 1532–1543.

Platt, J.C. (1998). Fast training of support vector machines using sequential minimal optimization, *in* B. Schölkopf *et al.* (Eds.), *Advances in Kernel Methods: Support Vector Learning*, MIT Press, Cambridge, MA, pp.185–208.

Platt, J.C. (2000). Probabilistic outputs for support vector machines and comparison to regularized likelihood methods, *in* A.J. Smola *et al.* (Eds.), *Advances in Large Margin Classifiers*, MIT Press, Cambridge, MA, pp. 61–74.

Quinlan, J.R. (1986). Induction of decision trees, *Machine Learning* **1**: 81–106.

R Development Core Team (2016). *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, `http://www.R-project.org`.

Radovanović, M. and Ivanović, M. (2008). Text mining: Approaches and applications, *Novi Sad Journal of Mathematics* **38**(3): 227–234.

Rios, G. and Zha, H. (2004). Exploring support vector machines and random forests for spam detection, *Proceedings of the 1st International Conference on Email and Anti Spam (CEAS-04), Mountain View, CA, USA*, pp. 398–403.

Rousseau, F., Kiagias, E. and Vazirgiannis, M. (2015). Text categorization as a graph classification problem, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics and the 6th International Joint Conference on Natural Language Processing (ACL-IJCNLP-15), Beijing, China*, pp. 1702–1712.

Said, D. and Wanas, N. (2011). Clustering posts in online discussion forum threads, *International Journal of Computer Science and Information Technology* **3**(2): 1–14.

Schölkopf, B. and Smola, A.J. (2001). *Learning with Kernels*, MIT Press, Cambridge, MA.

Sebastiani, F. (2002). Machine learning in automated text categorization, *ACM Computing Surveys* **34**(1): 1–47.

Selivanov, D. (2016). *text2vec: Modern Text Mining Framework for R*, R package version 0.4.0, `https://CRAN.R-project.org/package=text2vec`.

Siwek, K. and Osowski, S. (2016). Data mining methods for prediction of air pollution, *International Journal of Applied Mathematics and Computer Science* **26**(2): 467–478, DOI: 10.1515/amcs-2016-0033.

Szymański, J. (2014). Comparative analysis of text representation methods using classification, *Cybernetics and Systems* **45**(2): 180–199.

Wu, Q., Ye, Y., Zhang, H., Ng, M.K. and Ho, S.-H. (2014). ForesTexter: An efficient random forest algorithm for imbalanced text categorization, *Knowledge-Based Systems* **67**: 105–116.

Xu, B., Guo, X., Ye, Y. and Cheng, J. (2012). An improved random forest classifier for text categorization, *Journal of Computers* **7**(12): 2913–2920.

Xue, D. and Li, F. (2015). Research of text categorization model based on random forests, *2015 IEEE International Conference on Computational Intelligence and Communication Technology (CICT-15), Ghaziabad, India*, pp. 173–176.

Yang, Y. and Pedersen, J. (1997). A comparative study on feature selection in text categorization, *Proceedings of the 14th International Conference on Machine Learning (ICML-97), Nashville, TN, USA*, pp. 412–420.

Yessenalina, A. and Cardie, C. (2011). Compositional matrix-space models for sentiment analysis, *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP-11), Edinburgh, UK*, pp. 172–182.

Paweł Cichosz received his MSc and PhD degrees in computer science from the Warsaw University of Technology in 1994 and 1998, respectively. He is currently an assistant professor at the Institute of Computer Science there. His areas of research interests include machine learning, data mining, and artificial intelligence. He has also practical experience in applied data science projects.

Received: 22 September 2017
Revised: 18 January 2018
Re-revised: 10 May 2018
Accepted: 2 June 2018