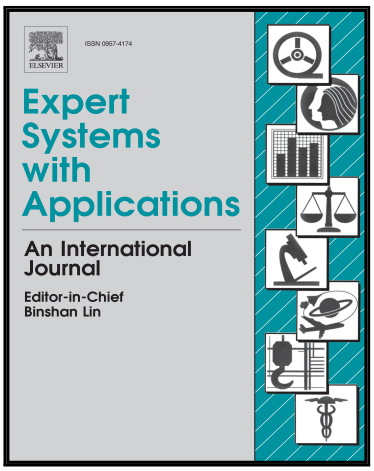
Accepted Manuscript



Text classification method based on Self-Training and LDA topic models

Miha Pavlinek, Vili Podgorelec

PII: S0957-4174(17)30166-5
DOI: 10.1016/j.eswa.2017.03.020
Reference: ESWA 11175

To appear in: *Expert Systems With Applications*

Received date: 26 August 2016
Revised date: 7 March 2017
Accepted date: 8 March 2017

Please cite this article as: Miha Pavlinek, Vili Podgorelec, Text classification method based on Self-Training and LDA topic models, *Expert Systems With Applications* (2017), doi: 10.1016/j.eswa.2017.03.020

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

ACCEPTED MANUSCRIPT

Highlights

- A novel text classification method for learning from very small labeled set.
- The method uses a text representation based on the LDA topic model.
- Self-training is used to enlarge labeled set from unlabeled instances.
- A model for setting methods' parameters for any document collection is proposed.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Text classification method based on Self-Training and LDA topic models

Miha Pavlinek*, Vili Podgorelec

Faculty of Electrical Engineering and Computer Science, Institute of Informatics, University of Maribor, Slovenia.

Abstract

Supervised text classification methods are efficient when they can learn with reasonably sized labeled sets. On the other hand, when only a small set of labeled documents is available, semi-supervised methods become more appropriate. These methods are based on comparing distributions between labeled and unlabeled instances, therefore it is important to focus on the representation and its discrimination abilities. In this paper we present the ST LDA method for text classification in a semi-supervised manner with representations based on topic models. The proposed method comprises a semi-supervised text classification algorithm based on self-training and a model, which determines parameter settings for any new document collection. Self-training is used to enlarge the small initial labeled set with the help of information from unlabeled data. We investigate how topic-based representation affects prediction accuracy by performing NBMN and SVM classification algorithms on an enlarged labeled set and then compare the results with the same method on a typical TF-IDF representation. We also compare ST LDA with supervised classification methods and other well-known semi-supervised methods. Experiments were conducted on 11 very small initial labeled sets sampled from six publicly available document collections. The results show that our ST LDA method, when used in combination with NBMN, performed significantly better in terms of classification accuracy than other comparable methods and variations. In this manner, the ST LDA method proved to be a competitive classification method for different text collections when only a small set of labeled instances is available. As such, the proposed ST LDA method may well help to improve text classification tasks, which are essential in many advanced expert and intelligent systems, especially in the case of a scarcity of labeled texts.

Keywords: classification, topic modeling, LDA, semi-supervised learning, self-training

*Corresponding author. Tel: +386 22207420

Email addresses: miha.pavlinek@um.si (Miha Pavlinek), vili.podgorelec@um.si (Vili Podgorelec)

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Dear editor(s) and reviewers,

we would like to thank you for your efort and helpful comments. Please find the complete Point-to-Point responses bellow:

Reviewer 1, Comment 1:

"The article has been significantly improved according to my comments, and I believe that the manuscript can be accepted in this round. One minor comment is that the highlight seems to be limited to 80 characters according to the regulation. Authors are suggested to provide a concise highlight for readers."

Thank you for noticing our mistake regarding Highlights. Now there are only the core findings included in a shortened form according to Elsevier's instructions. Highlights are now written as follows:

"
- A novel text classification method for learning from very small labeled set.
- The method uses a text representation based on the LDA topic model.
- Self-training is used to enlarge labeled set from unlabeled instances.
- A model for setting methods' parameters for any document collection is proposed. "

Reviewer 2, Comment 1:

"Authors must completely adhere to the three specifications of mandatory highlights below:

(1) Only include 3 to 5 highlights. Minimum number is 3, and maximum number is 5.

(2) There should be a maximum of 85 characters, including spaces, per highlight. Please kindly read this guideline carefully - the guideline does not say there should be a maximum of 85 words per highlight. For example, the word "impact" consists of 6 characters; the word "significance" consists of 12 characters. Remove any acronyms from Highlights.

(3) Only the core results of the paper should be covered. For details with examples, see http://www.elsevier.com/highlights."

Thank you for noticing our mistake regarding Highlights. Now there are only the core findings included in a shortened form according to Elsevier's instructions. Highlights are now written as follows:

"
- A novel text classification method for learning from very small labeled set.

ACCEPTED MANUSCRIPT

- The method uses a text representation based on the LDA topic model.
- Self-training is used to enlarge labeled set from unlabeled instances.
- A model for setting methods' parameters for any document collection is proposed. "

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

1. Introduction

Today's information-oriented society is inundated and overwhelmed with data and information. The trend of unstructured and semi-structured content is growing so rapidly that, without any automated support, it is becoming unmanageable for humans to make proper use of it. Part of the automation of text processing can be provided through text classification, which deals with the problem of assigning labels from predefined categories to text documents. In order for classification to be effective, enough labeled data must be available to train a successful classifier. Large training sets do not assure just better generalization, but also provide better accuracy. However, in reality, besides much unlabeled content, sometimes we only have a few labeled instances. This phenomenon is typical in many fields such as speech recognition, sensor data analysis or text mining (Chapelle et al., 2010). Of course, we can label instances manually, but labeling is considered to be a difficult and very time-consuming task (Ko & Seo, 2009). As an alternative, we can apply Semi-Supervised Learning (SSL) methods, where unlabeled data can be utilized to aid classification. By taking advantage of additional information, SSL methods enlarge the initial labeled set, which can then be used as an input to traditional machine learning algorithms. SSL can be useful especially in so-called cold-start problems where the system has just been created, or where we want to initiate a classification task on existing but unclassified data (Ocepek et al., 2015), e.g. by classifying a bunch of existing emails into newly created folders, determining medical diagnosis from existing medical records, or in a recommender system, where new user profiles are hard to define from just a few known activities and the popularity of new items cannot be set until some user ratings or tags are given (Xie et al., 2016).

Since many SSL methods are based on measuring similarities between labeled and unlabeled data, the representation of documents is crucial. Indeed, the representation of a text as unstructured content is even more important than choosing the right machine learning algorithm and tuning its parameters (Joachims, 1998). While structured content can be represented uniformly with feature vectors, unstructured content can be represented in different ways. In text mining tasks we commonly use the Vector Space Model (VSM) representation, where features are based on words as independent units, and values are provided through different weighting schemes, such as Term Frequency – Inverse Document Frequency (TF-IDF). However, the drawback of such a representation is that the order of words and their semantic meaning are ignored (Sriurai, 2011). Furthermore, such word vectors are sparse and very high-dimensional; therefore, it is impossible to use just any machine learning algorithm on them seamlessly (Beyer et al., 1999). For dimensionality reduction, feature selection techniques like Information Gain or Chi Square can be used, but sparseness still remains. On the other hand, we can use topic models, which consider context, reduce the number of features and compact the representation of text (Colace et al., 2014). In this way, we can represent each document in a latent topic space instead of a word space.

ACCEPTED MANUSCRIPT

In recent studies it has been shown that document similarity measures are more efficient when based on LDA than when they are based on the bag-of-words with TF-IDF representation (He & Zheng, 2015) (Xie & Xing, 2013). In this context, the semantic similarity between two texts was also a matter of examination (Niraula et al., 2013). The results showed that topic models outperform typical representations in a supervised setting when the proportion of training data is very small (Lu et al., 2011) (Blei et al., 2003).

In this paper we investigated the idea of using topic models to represent text in order to improve performance in a semi-supervised setting for very small initial labeled sets (with only up to 10 instances per class). Many different semi-supervised methods and frameworks have been developed so far (Zhang et al., 2015) (Triguero et al., 2015a) (Colace et al., 2014) (Guzmán-Cabrera et al., 2009), and many classification methods using topic models have been proposed recently (Venugopalan & Rai, 2015) (Rubin et al., 2012) (Sriurai, 2011) (Zhou et al., 2009). Meanwhile, we found only two approaches where the LDA model was modified to be used in semi-supervised classification methods. These are named as a semi-supervised LDA but with different abbreviations: ssLDA (Wang et al., 2012) and sLDA (Fu et al., 2015). They both use LDA and introduce a semi-supervised approach by using labels inside the Gibbs sampling process. ST LDA, on the other hand, uses LDA topic model to extract features, which are then passed forward to a self-training algorithm. In comparison with ssLDA and sLDA, our proposed ST LDA method (text classification method based on self-training and LDA topic models) was designed especially for very small labeled data sets. The sLDA differs in fixing the number of topics with number of classes, which makes it impossible to identify other hidden topics that can reveal unknown relationship with classes. Unlike sLDA and ST LDA, ssLDA can be used to classify documents over multiple different labels.

To the best of our knowledge, this is the first work of performing self-training on text with topic based representation. The contributions of this study are as follows:

- We proposed a novel text classification method, ST LDA, which is designed to deal with situations where there is only a small set of labeled instances available to learn from. The ST LDA method is based on a self-training algorithm, which is used to enlarge an initially small set of labeled instances with the use of information from unlabeled content. The representation of documents is based on LDA topic models.
- We designed a model for the analysis of optimal parameter values and performed extensive computational tests to determine them. The derived model can be used to tune the parameters for any new text collection.

We conducted experiments on a diverse set of 11 very small initial labeled sets sampled from six publicly available document collections with retaining ratios of 0.1% and 0.5% of instances. They demonstrate the strength of the proposed ST LDA method and show that it outperforms supervised classifiers as well as state-of-the-art semi-supervised methods.

ACCEPTED MANUSCRIPT

The remainder of this paper is organized as follows. Section 2 outlines the theoretical background where semi-supervised learning and topic modeling are summed up. Section 3 presents the proposed ST LDA method for text classification from constrained labeled sets. Section 4 describes the performed experiments, presents the obtained results and their analysis. Section 5 presents discussion. Finally, section 6 concludes the paper and suggests some future work.

2. Background

In this chapter we present semi-supervised learning and topic modeling as essential concepts in our method.

2.1. Semi-Supervised Learning

The Semi-Supervised Learning (SSL) paradigm is an extension of supervised learning with the principle of inclusion of unlabeled instances, which are used as background knowledge. When using SSL for a classification task, the entire approach is usually denoted as a Semi-Supervised Classification (SSC) (Chapelle et al., 2010), where unlabeled data can be understood as part of the entire training set. In general, SSC is intended to learn a better classifier by including unlabeled instances rather than using only a labeled set. This approach can be performed in two different ways: as transductive learning, where labeled data is used to build a classifier which is then used on unlabeled data; and inductive learning, where the evaluation is carried out on a validation set (a part of data that is excluded from the available training data before the semi-supervised training phase) (Chen & Wang, 2011).

There are many SSL methods available, such as self-training (Yarowsky, 1995), co-training (Mitchell & Blum, 1998), Expectation Maximization with Naïve Bayes (Nigam et al., 2000), graph-based algorithms and semi-supervised SVM (Zhu & Goldberg, 2009). The most simple and adaptable between them is the self-training method which was also used as the foundation for our proposed method.

Self-training is a wrapper method for SSL (Triguero et al., 2015b) where, in the first place, a classifier is usually trained with an underlying classifier on a small number of labeled instances D^l. Then the learnt classification model is used to classify unlabeled instances D^u and, according to the gained class predictions, the most reliable instances with their predicted labels are moved to the labeled set. Reliability is determined with a confidence measure that can be customized. The entire process is then repeated until all the instances from D^u are moved to D^l or certain stopping criteria are met, which can be defined by the maximum number of iterations or any other limitation rule. Since too many mislabeled instances can have a negative effect on the further learning process, especially in early iterations, it is very important how many instances are moved within a single iteration. This is determined with the throttling parameter. In self-training it often turns out that the most reliable instances are classified

ACCEPTED MANUSCRIPT

predominantly only in certain categories, which leads to imbalanced data sets. This can be overcome with balancing techniques, such as adding additional instances from the category or removing the least reliable instances from the labeled set.

2.2. Topic modeling

Representing a document by relying only on its particular words is not good enough for the efficient comparison of documents. Namely, two documents in the same context can use different vocabularies that contain words with similar meanings. For this purpose, we need to represent the documents within a common semantic space. The most essential technique for such a representation is Latent Semantic Analysis (LSA) which transforms text with Singular Value Decomposition (SVD) and represents it as latent concepts in a low dimensional semantic space (Deerwester et al., 1990). Its follower Probabilistic Latent Semantic Analysis (PLSA) is a basic topic model which improves interpretation with topics as multinomial word distributions (Hofmann, 1999). As PLSA is based on the maximum likelihood estimation for given documents and is, therefore, susceptible to overfitting, Latent Dirichlet Allocation (LDA) was proposed as an improved topic model, which introduces Dirichlet prior and provides a fully generative model (Blei et al., 2003). With LDA each document is represented as a multinomial distribution of topics where topics can be seen as higher level concepts that are analogous to clusters. It is based on the assumption that each document in a collection is created from several latent topics, where each topic is presented with a mixture of words. In this way, we can describe the creation of documents with a two-step process:

1. For each document $m \in M$ sample a topic proportions θ_m from Dirichlet distribution $Dir(\alpha)$.
2. For each word placeholder n in the document m
   a. Choose a topic $z_{m,n}$ randomly according to the sampled topic proportions θ_m.
   b. Choose a word $w_{m,n}$ randomly from the multinomial distribution ϕ_k of the previously chosen topic $z_{m,n}$.

In the above-mentioned process, the parameters α and β are vectors of hyperparameters which determine the Dirichlet prior on θ as a set of topic distributions for all documents and on ϕ as a set of word distributions in all topics. Typically, symmetric Dirichlet priors are used, where $\alpha_1 = \alpha_2 = \alpha_K = \alpha$, which define how probability distribution is concentrated into a single point. The entire process is depicted in Figure 1.

Although LDA is a relatively simple model, exact inference of its hidden variables and parameters is intractable. Therefore, a number of algorithms are available to get approximate estimates of model parameters ranging from variational EM (Blei et al., 2003) to expectation propagation (Minka & Lafferty, 2002) and Gibbs sampling. In our study we used Gibbs sampling, which is based on the Markov chain Monte Carlo (MCMC) and was initially present as

ACCEPTED MANUSCRIPT

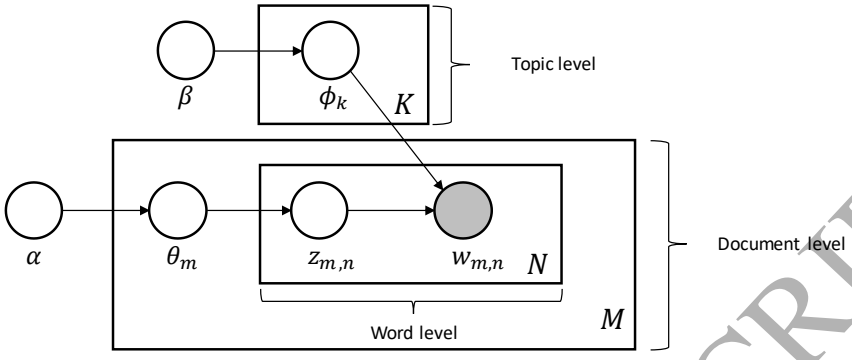


Figure 1: Latent Dirichlet Allocation model

related with LDA in (Griffiths & Steyvers, 2004). As an input it takes α and β parameters, the number of topics and, since it is calculating estimates for θ and ϕ in iterations, it also needs the number of iterations. Following related research (Steyvers & Griffiths, 2007) (Lu et al., 2011), parameters α and β can be fixed to reasonable defaults, such that $\alpha = 50/K$ and $\beta = 0.01$. The other two parameters importantly determine the quality of a topic model, but their optimal values are difficult to define.

3. ST LDA method

In this section we describe the proposed text classification method, denoted as Self-Training with Latent Dirichlet Allocation (ST LDA), which is trained in a semi-supervised fashion. For this purpose, we used the self-training method while representing the documents with topic distributions based on the LDA topic model. As an input, the ST LDA method takes a few labeled documents, which form an initial labeled set, and many more unlabeled documents; the unlabeled documents can be composed from various sources with a similar topic distribution as in the initial labeled set. As the output, the method produces a classification model which is built with a supervised classifier on a final labeled set – the initial labeled set enlarged with labeled instances from an unlabeled set. Thus, ST LDA can be denoted as an inductive classification method. The ST LDA method consists of two major parts: a self-training algorithm for enlarging the initial labeled set, and a model for setting of the algorithm's parameters. The parameter estimation model is included in the ST LDA method in order to provide a fully automated text classification method without the need of manually tuning parameters for each new document collection.

ST LDA begins with the model for determining the parameters over an arbitrary text collection. Calculated parameters are then, in addition with the initial (very small) labeled and (much larger) unlabeled set, sent to the self-training algorithm, which is the central part of the method. All the documents

ACCEPTED MANUSCRIPT

are first represented with a topic distribution resulting from the LDA topic model, after which the self-training method is performed. The output from the self-training algorithm is an enlarged labeled set, which is applicable for training purposes in any supervised classification method, and the learned classification model can then be used to classify other unknown (unlabeled) instances. The entire method is presented in Figure 2.

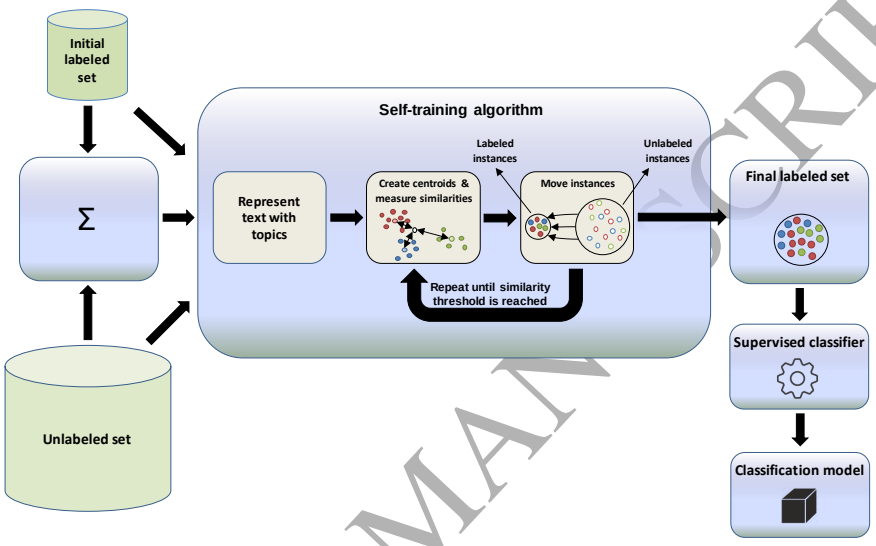


Figure 2: The overview of the proposed ST LDA method

3.1. Enlarging a labeled set with the self-training method

We developed a self-training algorithm, which consists of two phases and is responsible for enlarging the initial labeled set. The goal of the initial phase is to achieve a common topic based representation from both labeled and unlabeled data. Therefore, all instances are first combined into one set on which topic modeling is performed. Gibbs sampling is used to build the LDA model and each instance is then represented with LDA topic distributions. At this point, the labels from labeled data are not taken into consideration.

In the second phase, unlabeled data are moved iteratively into a labeled set until the predefined threshold is reached. Only the most reliable unlabeled instances are moved into the labeled set. For this purpose, we defined our own semantic similarity measure based on topic distributions and cosine similarity measure. Since computing the distance between unlabeled and labeled instances in a training set is time consuming, we proposed making centroids for each class and using them to measure distances similar to the centroid-based document classification (Han & Karypis, 2000). Each iteration is performed in two steps:

ACCEPTED MANUSCRIPT

a. For each category we created centroid vectors, as averages of labeled instances from the given category. Then we computed the cosine distance between unlabeled instances and centroid vectors. Here, the cosine distance was defined as a complement measure to cosine similarity as shown in Eq. (1). This measure is also commonly used in text mining tasks (Turney & Pantel, 2010).

$$d_{cos}(x, y) = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \quad (1)$$

b. Unlabeled instances are sorted by reliability, which is defined by the difference between distances from the two nearest centroids. The higher number denotes an instance which is much closer to the nearest centroid than to the next one. The most reliable unlabeled instances are then moved into a labeled set where they are labeled according to its nearest centroid. By choosing unlabeled instances we also consider class (im)balance so that, when possible, a labeled set is distributed uniformly. For this purpose, imbalance ratios are first calculated for all classes. Each class ratio (r) is then subtracted from the throttling value (R) so that mostly $R - r$ instances can be moved to a particular class.

This process finishes when, for each unlabeled instance, the difference between distances from the two nearest centroids is smaller than the similarity threshold, which is determined in advance. With the throttling parameter, we also predefine the number of moved instances in one iteration. As a result of this phase we get the final labeled set. The exact algorithm is presented in Algorithm 1

3.2. Designing a model for parameter estimation

To make our approach more useful we wanted to define a complete method which, in addition to the algorithmic part, proposes the parameter values as well. Instead of trying to optimize a method on a specific data set, we designed a general parameter estimation model based on 6 data sets, which could serve for the configuration of the algorithm for any given text collection.

Parameter setting or tuning is usually performed manually using the cross-validation technique on a training set and optionally a validation set (Siersdorfer & Weikum, 2005) (Wen et al., 2015). On the other hand, parameter values can also be defined using grid search, where every possible combination of parameter values from the given subset is evaluated separately. In our case, we need to set four parameters, so we could have afforded numerous experiment runs with various combinations of parameters. However, we combined grid search with the ANOVA statistical test as proposed by (Castillo et al., 2012) to determine the influences of particular parameters.

All the experiments within this scope have been performed on the predefined (original) training sets, which have been split into smaller training and validation sets at a ratio of 70:30. Then we sampled 0.1% and 0.5% of labeled

ACCEPTED MANUSCRIPT

Algorithm 1 Proposed self-training algorithm

Input:

D^l - labeled documents

D^u - unlabeled documents

- $D = D^u \cup D^l; D^u \gg D^l$
- for each category in D there is at least one instance in D^l

ST - similarity threshold

R - throttling

T - number of topics

GI - number of gibbs iterations

Output:

final labeled set $\widehat{D^l}$

Initialization:

Representing all documents in D with topic distributions based on parameters T and GI

Self-training:

while $\varepsilon > ST$ **do**

I. From categories of labeled documents D^l create centroids $C = c_1, \ldots, c_m$ with labels l_{c_i} as defined by each category.

II. For each $d_k^{(u)}$ measure cosine distances to all of the centroids $c_i \in C; d_{\cos}(d_k^{(u)}, c_i)$.

III. For each $d_k^{(u)}$ calculate the difference between two minimum distances and order unlabeled instances by the obtained differences from the highest to the lowest value.

$$\bar{C} = \{c_x \in C \mid \exists c_y \in C : d_{\cos}(d_k^{(u)}, c_x) \geq d_{\cos}(d_k^{(u)}, c_y)\}, C \subset \bar{C}$$

$$c_{min_1} = argmin_{c_i \in C}(d_{\cos}(d_k^{(u)}, c_i))$$

$$c_{min_2} = argmin_{c_i \in \bar{C}}(d_{\cos}(d_k^{(u)}, c_i))$$

$$dif_k = d_{\cos}(d_k^{(u)}, c_{min_2}) - d_{\cos}(d_k^{(u)}, c_{min_1})$$

IV. Define value for $\varepsilon; \varepsilon = max(\{dif_1, ..., dif_n\})$

V. If $(\varepsilon > ST)$

(1) Calculate the ratio (r) between the number of instances in categories in D^l and for each category select $R - r$ instances with lowest dif_k from D^u.

(2) Move selected instances from D^u to D^l, where $l(d_k^{(u)}) = l_{C_{min_1}}$ for each $d_k^{(u)}$.

end while

ACCEPTED MANUSCRIPT

instances with stratified sampling from these smaller training sets; all the non-sampled instances were used as the unlabeled set. In this way, we prepared 11 experimental benchmarks, each containing a small initial labeled set and a larger unlabeled set for training, and a validation set for testing. In this manner, the algorithm has been run with every possible combination of parameter values on training sets and the results obtained on the validation sets were considered.

All the ranges of the parameters' values were chosen in accordance with preliminaries and common practices in related works. For the similarity threshold we used values ranging from 0.1 to 0.9, in increments of 0.1. For the throttling parameter, values between 10 and 200 were used in steps of 20 (in the first phase) or 10 (in the second phase). In order to measure the impact of the quality of the topic model we used topic numbers ranging from 20 to 200 in steps of 20 and Gibbs sampling iterations with the values 500, 1,000 and 1,500. The entire designed theoretical model is depicted in Figure 3.

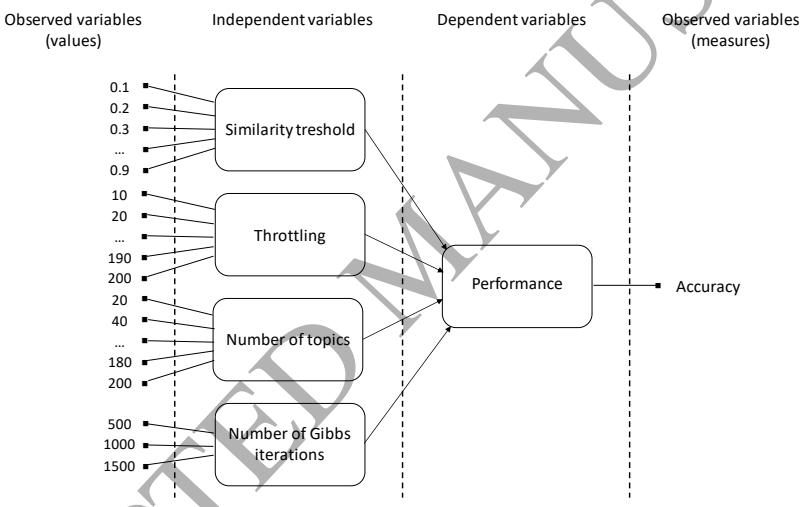


Figure 3: The theoretical model for the analysis of optimal parameter values

In this way, we examined 29,700 different combinations in the first phase (for determining which parameters had the greatest influence on the results) and 2,200 in the second phase (for determining the values of specific parameters) concurrently. With such a brute-force approach we first determined which parameters had the greater influence on the performance of the algorithm and then established a general formula to set each parameter. For this purpose, we used the classification accuracy as the evaluation metric and the Naïve Bayes Multinomial algorithm as the supervised classification method due to its training speed.

Similarity threshold

Due to time complexity we limited the first part of our experiments to throt-

ACCEPTED MANUSCRIPT

tling parameters with values in a range from 20 to 200 in increments of 20. With this limitation we performed 29,700 runs for all value combinations. For this part we applied the ANOVA statistical test to determine whether the influence of a change in a particular parameter value was significant or not, as suggested in (Castillo et al., 2012). To obtain the ANOVA table we used the SPSS statistical tool. This can be seen in Table 1, where each factor is represented by the sum of squares, the degrees of freedom, the mean square, the statistical F value and the significance level. We treated each parameter as an independent factor and examined their influences on the performance of the method as a dependent variable.

Table 1: ANOVA table for the accuracy as response to different parameters

Parameter	Sum of Squares	df	Mean Square	F	Sig.
Similarity threshold	1094432.060	8	136804.008	831.142	.000
Throttling	29420.010	9	3268.890	19.860	.000
Topic num.	29049.792	9	3227.755	19.610	.000
Gibbs iteration num.	1066.243	2	533.121	3.239	.039

The results show that a change in either parameter influences the performance significantly and with regard to the F-statistics values, the similarity threshold is by far the most sensitive between them. Obviously, the least important parameter is the number of Gibbs iterations, while throttling and the number of topics have a very similar influence. According to these findings, we first defined the similarity threshold. Based on descriptive statistics, with metrics like minimum, maximum, median, mean and standard deviation, we noticed that a similarity threshold with a value of 0.1 gives the best results (Table 2). In this manner, we fixed the value of the similarity threshold at 0.1 in all subsequent experiments.

Table 2: Descriptive statistics for different similarity threshold values

Sim. threshold	N	Minimum	Maximum	Median	Mean	Std. Deviation
0.1	3300	13.23988	83.90037	71.8712	68.21732	12.12702
0.2	3300	13.23988	83.11057	71.58145	67.61355	12.01514
0.3	3300	13.23988	81.34872	71.19421	66.87777	12.17785
0.4	3300	13.23988	80.58317	70.6302	66.09059	12.33377
0.5	3300	13.23988	79.64763	69.82354	65.10343	12.5341
0.6	3300	13.23988	79.70838	68.6088	63.67441	12.86578
0.7	3300	13.23988	78.85784	66.16039	61.33573	13.25327
0.8	3300	11.37072	77.339	60.50699	56.34348	13.97217
0.9	3300	10.98131	76.06318	49.94055	48.5882	14.59706
Total	29700	10.98131	83.90037	68.1218	62.64939	14.25825

Number of Gibbs iterations

In continuation, we preserved only the results with a similarity threshold of 0.1 and with the ANOVA test reexamined the influence of the remaining

ACCEPTED MANUSCRIPT

parameters on the performance under these new conditions - the results are presented in Table 3.

Table 3: ANOVA table for accuracy as a response to different parameters with preserving a similarity threshold at 0.1

Parameter	Sum of Squares	df	Mean Square	F	Sig.
Throttling	2507.034	9	278.559	1.899	.048
Topic num.	6780.730	9	753.414	5.181	.000
Gibbs iteration num.	287.786	2	143.893	0.978	.376

Although more iterations in Gibbs sampling generally improves the quality and stability of the topic model (Griffiths & Steyvers, 2004), in our setting it did not influence performance significantly. On the other hand, this number importantly contributes to time complexity. Therefore, it was reasonable to fix it to the lowest value from the range. In this manner, we fixed the number of Gibbs iterations to 500. At this point we extended the throttling parameter ranging from 10 to 200.

Number of topics

Determining the optimal topic number has been the focus of many previous studies (Greene et al., 2014) (Niraula et al., 2013) (Lu et al., 2011) (Wang et al., 2014) and is still an open issue. The problem lies in the fact that the optimal number of topics varies for different collections, where a larger number provides fine grained topics that are very specific. On the other hand, with a smaller number of topics we get topics that are too general. To this end, there are some predictive metrics on a theoretical basis, among which we examined four.

First we tried to define the number of topics with a cross-validation technique, which is used commonly for determining parameter values for specific collections in runtime. This way the training set is first split into complementary subsets and then, iteratively, every single part is used for validating the model that is trained from the other parts. Anyhow, this approach was not suitable for our problem, because there were only a few labeled instances of each class and, as we split the trainining sets, too few instances (or even none) remained for training purposes.

We also tried two metrics suggested by (Cao et al., 2009) and (Arun et al., 2010) and compared them with perplexity or held-out likelihood (Blei et al., 2003). As the values were not distributed normally, we compared them with a nonparametric Friedman statistical test, which was carried out considering the average accuracy results for the number of topics proposed by each metric on all data sets with all throttling values. The results of the Friedman test showed no significant difference between them ($\chi^2(2) = 2.476; sig. = 0.290$). Therefore, we decided to use perplexity which is, according to literature, the most common metric to determine the optimal number of topics.

Perplexity reflects the ability of generalization of the model of the unknown instances where a lower value represents a higher generality. This metric is often

ACCEPTED MANUSCRIPT

used to evaluate the quality of a language model and operates on the principle of supervised learning. Perplexity can be calculated with the following equation:

$$per(D_{test}) = \exp\left\{-\frac{\sum_{d=1}^{M} \log p(w_d)}{\sum_{d=1}^{M} N_d}\right\}, \tag{2}$$

where D_{test} represents the held out data set, M denotes the number of documents in a collection, w_d is describes words and N_d is the number of words in a given document d. In this manner, the number of topics is determined by calculating perplexity for topic numbers between 20 and 200 (or more if perplexity for the highest topic number has the lowest value) where the number of topics for the lowest perplexity is taken.

Throttling

To define a general way of determining the throttling parameter, we were trying to find dependencies between throttling and other properties within 11 initially prepared labeled sets, with the retaining ratios (RR) of 0.1% and 0.5%. First, we obtained optimal throttling values (OV), which gave us the best results for different combinations in the performed experiments. The similarity threshold, as the parameter with the highest influence on the results, and the number of Gibbs iterations, as the parameter without a significant influence on the results (after fixing the similarity threshold) was fixed as described above. On the other hand, the number of topics, where the optimal number of topics was different for each individual document set, varied from 20 to 200 in steps of 20.

For each data set we chose some factors that we assumed could have some influence on the results. In this way we chose Kullback-Liebler (KL) divergence, Jensen-Shannon (JS) divergence and average cosine similarity between the initial labeled set and the unlabeled set. The cosine distance is the typical similarity measure between two texts. KL divergence is an asymmetric difference measure between two probability distributions (Eq. (3)), and JS divergence (Eq. (4)) is its symmetric derivation where M is the average of distributions P and Q (Eq. (5)). All these measures were obtained as averages across different numbers of topics (from 20 to 200). Other included factors were the number of instances in the initial labeled set (#LI), the number of instances in the unlabeled set (#UI), the vocabulary size (VS) for the training set, the number of classes (#C) and the imbalance ratio (IR) between the majority class and the minority class in the initial labeled set (Eq. (6)). All these values are presented in Table 4.

$$D_{KL}(p||q) = \sum_i p(i) \log(\frac{p(i)}{q(i)}) \tag{3}$$

$$D_{JS} = \frac{1}{2} D_{KL}(P||M) + \frac{1}{2} D_{KL}(Q||M) \tag{4}$$

$$M = \frac{1}{2}(P + Q) \tag{5}$$

ACCEPTED MANUSCRIPT

$$IR = \frac{\#\ of\ instances\ in\ majority\ class}{\#\ of\ instances\ in\ minority\ class} \tag{6}$$

Table 4: Data sets properties and their values as used in the model

	RR	OV	KL	JS	COS	VS	#C	IR	#LI	#UI
20 Newsgroups	0.1%	40	10.097	0.2615	0.0566	20129	20	1.00	20	7885
	0.5%	30	0.5510	0.1490	0.0564	20129	20	4.00	40	7865
Reuters R8	0.1%	80	19.832	0.4613	0.0400	5953	8	2.00	10	3829
	0.5%	40	12.851	0.3289	0.0427	5953	8	7.00	19	3820
Reuters R52	0.5%	30	14.304	0.3537	0.0331	6675	52	1.00	52	4520
Google snippets	0.1%	100	25.110	0.5484	0.0356	6396	8	1.00	8	7034
	0.5%	60	10.631	0.2751	0.0365	6396	8	9.00	35	7007
WebKB	0.1%	110	22.877	0.5277	0.0721	5521	4	1.00	4	1958
	0.5%	70	15.241	0.3616	0.0698	5521	4	5.00	10	1952
Ohscal	0.1%	30	16.680	0.4170	0.0487	8331	10	1.00	10	5459
	0.5%	60	0.7980	0.2157	0.0552	8331	10	5.00	27	5442

To define a formula for determining the throttling parameter, we built a regression model using SPSS and its automatic linear modeling function with stepwise regression. The obtained regression model determined that, with average KL divergence between labeled and unlabeled instances and the number of classes, we could predict throttling value significantly with a 60.4% of variance where $p < 0.0137$, which is less than 0.05. According to the coefficients table (Table 5) we can calculate the throttling parameter value for each document set with the following formula:

$$R = KL * 29.531 - \#C * 0.793 + 26.801 \tag{7}$$

Table 5: Coefficients table with unstandardized coefficients (B), coefficient's standard error, statistic tests (t) and observed significance levels (Sig.)

	B	Std. Error	t	Sig.
Intercept	26.801	17.447	1.536	0.163
KL	29.531	9.571	3.085	0.015
#C	-0.793	0.427	-1.857	0.100

4. Experiments

In this section we present an evaluation of the proposed ST LDA method, which is based on a number of experiments. We compared our method with other comparable methods and variations such as supervised learning with Support Vector Machine (SVM) and Naïve Bayes Multinomial (NBMN), Self-Training based on BOW representation with a TF-IDF weighting scheme, a

ACCEPTED MANUSCRIPT

combination of Expectation-Maximization and Naïve Bayes classifier(EM-NB) and its augmented version, EM-NB with components (EMC-NB).

4.1. Experimental setup

To perform a comparative analysis, six fully labeled training sets (presented below) were reduced to initial labeled sets with retaining ratios of 0.1% and 0.5% of instances, where the retaining ratio denotes the proportion of instances in the training set; all the remaining instances were used to form unlabeled sets. The initial labeled sets were created with stratified random sampling for each category in a given data set. In one case (Reuters R52 data set) the two created initial labeled sets were equal and, therefore, only one was used. In this manner, we created 11 pairs of (very small) labeled and (much larger) unlabeled sets of instances. The proposed ST LDA method was then used to prepare 11 final labeled sets, which were trained with two supervised classification methods (SVM and NB). The trained classifiers were finally evaluated on the test sets.

The self-training algorithm was implemented in the Java programming language using WEKA (Hall et al., 2009), an open source machine learning environment, from which we also used NBMN (Naïve Bayes Multinomial) to train a classifier from the final labeled set. Similarly, LibSVM (Chang & Lin, 2011) implementation was used to train a SVM classifier with a linear kernel on the final labeled set. To build the LDA topic models we used the MALLET toolkit (McCallum, 2002).

4.2. Data sets

We performed experiments over six data sets: 20 Newsgroups, Reuters R8, Reuters R52, WebKB with four classes, Ohscal and Google snippets. For each data set we performed some preprocessing steps where the words were converted to lower case, stop words and the words that were shorter than three characters were removed and all the remaining words were stemmed with Porter Stemmer. The most notable characteristics, like the number of documents, the number of classes and the number of terms after preprocessing steps, are presented in Table 6.

20 Newsgroups consists of approximately 20,000 documents, which are distributed nearly equally across 20 categories. These documents were gathered from UseNet, where each category represents a newsgroup. Categories were organized into a hierarchical structure, where the main categories are computers, recreation and entertainment, science, talk, social, miscellaneous and a specific alt category. We used the bydate version, where documents are first sorted by date then divided into train and test sets with a 60:40 split.

Reuters R8 and R52 sets are derived from an essential Reuters-21578 data set with 90 categories. Both sets are single labeled with a ModApte split, which assumes that there are at least two instances in each category, one for training and one for the test set, with the overall ratio around 70:30. Reuters R8 consists of a set of 7,674 documents which is divided into 8 categories, while Reuters R52 contains 9,100 instances in 52 categories.

ACCEPTED MANUSCRIPT

Ohscal is a subset of the well-known Ohsumed data set, which consist of titles and abstracts from medical journals. Since Ohsumed include multi-labeled documents, documents in Ohscal are labeled with a single category. The latter contains 9,121 instances from the 10 different fields of medicine that are used as categories. There is no predefined split for this set, thus we partitioned documents randomly such that 70% was used as training data and the remaining 30% was used as test data.

In WebKB collection we can find texts from the websites of university computer science departments, where pages are divided into seven categories: student, faculty, staff, course, project, department and other. In our experiments we used its variant, which only covers the first four categories with a 66:33 split.

The Google snippets collection was built as a part of the research on classifying short texts (Phan et al., 2008). Labeled train and test data were created by inserting phrases from eight different domains into a search engine and using the retrieved snippets. The collection contains 10,060 train instances and 2,280 test instances. In our case it was used to examine how our model performed on a short text, which is a typical form of microblogs, posts and discussions in social networks and other rapidly growing social web content (Rao et al., 2016).

Table 6: Basic characteristics of the used original data sets

	# Documents	# Categories	# Terms
20 Newsgroups	18,846	20	33,489
Reuters R8	7,674	8	7,808
Reuters R52	9,100	52	8,937
Google snippets	12,340	8	7,039
WebKB	4,199	4	7,719
Ohscal	9,121	10	10,036

4.3. Experimental results

First, we wanted to evaluate the proposed parameter estimation model. In this manner, we compared the results obtained with ST LDA, where 1) all the parameters were set using the proposed parameter estimation model, and 2) all the parameters were set in accordance with the performed grid search. For the grid search, we used data from all 31,900 experimental runs (29,700 from the first and 2,200 from the second phase) performed in the model design phase and took parameters that provided the best result. Although a grid search approach is extremely time-consuming (the time needed for performing a grid search on a single data set can be expressed in days), the results of the grid search parameters showed no significant difference in terms of prediction accuracy (Figure 4), which was confirmed with a Wilcoxon signed-rank test ($Z = -0.164; sig. = 0.884$). Moreover, due to algorithms' non-deterministic behavior, parameters calculated from our model in several cases give even better results. Finally, the use of extremely tuned parameters in learning algorithms can easily produce over-fitted classifiers, which perform poorly on unseen data.

ACCEPTED MANUSCRIPT

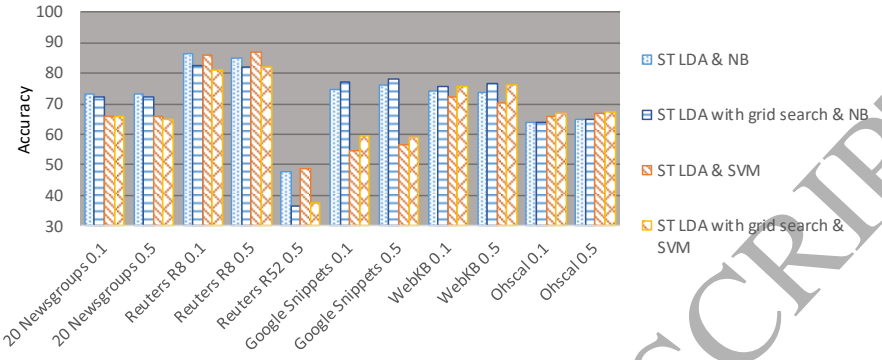


Figure 4: Results compared between ST LDA with parameters obtained (1) from proposed estimation model and (2) from grid search

We then compared the proposed ST LDA method (using the proposed parameter estimation model to set all the parameters) with the following 5 methods and variants:

- *Baseline:* In order to be convinced of the advisability of the Semi-Supervised Learning approach we used the supervised setting as a baseline, where classifiers were trained on initial labeled sets with NBMN and SVM classification algorithms.
- *ST TF-IDF*: Next, we compared a similar method to ST LDA where, instead of LDA based representation, the input text was represented with a bag of words and TF-IDF weighting, denoted as ST TF-IDF.
- *EM-NB*: Expectation Maximization (EM) is a statistical method for estimating parameters with incomplete data in two steps: the Expectation step (E-step) and the Maximization step (M-step). In (Nigam et al., 2000) authors propose EM where Naïve Bayes classifier is used to estimate parameters from labeled instances and then to assign probabilistically-weighted class labels to unlabeled instances. In a second iteration new labeled instances are considered to re-estimate parameters and so it iterates until the results converge. Unlike ST LDA, it is fully deterministic, but a great drawback to its value is that it converges to local optima.
- *EMC-NB*: In the same work the authors also presented an extension which violates the assumption that there are as many mixture components in data as the targeted classes. They proposed dividing each class into many components wherein the number of them has to be determined in advance. With this idea, in the initial phase instances are labeled with components

ACCEPTED MANUSCRIPT

that are derived from the original class and whose probabilities are distributed randomly. EM-NB is then performed on this representation and, in the final phase, the components with all probabilities are merged into base classes again. Choosing the number of mixture components is an important drawback of this method. The authors suggest leave-one-out cross validation method, which (1) is very time consuming and (2) cannot be taken into consideration by very small sets where only one labeled instance can be present in a category.

Baseline, ST TF-IDF and ST LDA methods were evaluated with SVM and NBMN classification methods. Expectation maximization with Naïve Bayes (EM-NB) and its extension with components (EMC-NB) were included in the comparison as a gold standard for semi-supervised classification methods.

For ST LDA the average and standard deviation over 10 runs for each of the 11 initial data sets is reported. Except EM-NB and EMC-NB, all other methods were trained with Naïve Bayes and SVM classifiers on final labeled sets and evaluated on original test sets. The EMC-NB was run with 3, 6, 9 and 12 components and only the best result was used for comparison. Similar to model design, classification accuracy was used as an evaluation measure. The final results are presented in Table 7, where the highest accuracies are highlighted in bold text.

Table 7: Classification accuracy results of Baseline, ST LDA, ST TF-IDF, EM-NB and EMC-NB over multiple initial labeled sets

	RR	Baseline		ST LDA		ST TF-IDF		EM-NB	EMC-NB
		NB	SVM	NB	SVM	NB	SVM		
20 Newsgroups	0.1%	26.65	26.58	73.39 ± 0.54	65.73 ± 1.03	57.51	52.55	50.98	50.92
	0.5%	33.86	28.79	73.27 ± 0.72	65.81 ± 0.83	64.67	60.14	55.14	50.96
Reuters R8	0.1%	41.21	26.27	86.42 ± 0.82	85.98 ± 3.23	14.71	16.22	32.34	35.54
	0.5%	83.69	80.04	85.02 ± 0.98	86.64 ± 1.25	59.94	62.31	74.60	**88.31**
Reuters R52	0.5%	31.50	36.10	47.71 ± 7.07	49 ± 6.24	20.02	19.59	24.77	45.37
Google snippets	0.1%	16.45	13.25	74.51 ± 1.57	54.53 ± 3.66	69.12	51.71	37.32	42.02
	0.5%	30.79	23.86	76 ± 0.85	56.59 ± 2.21	72.94	58.46	72.50	69.65
WebKB	0.1%	54.37	41.91	73.98 ± 0.52	72.3 ± 1.38	67.84	67.84	72.85	73.78
	0.5%	62.39	60.10	73.92 ± 0.6	70.22 ± 1.36	66.48	64.11	72.35	**74.36**
Ohscal	0.1%	30.61	30.37	64.11 ± 0.76	65.77 ± 0.83	49.30	49.78	50.04	48.04
	0.5%	45.00	36.58	64.7 ± 0.51	66.71 ± 0.65	43.12	45.72	52.52	54.91

The results indicate that the proposed ST LDA method outperforms other methods in 9 out of 11 initial labeled sets. We tested our observations using the nonparametric statistical tests as suggested by Demšar (Demšar, 2006), using an alpha level of $\alpha = 0.05$. Therefore, we initially provided ranks for all methods according to their results (Table 8). The differences between average ranks were tested using the Friedman test, where the results provided with Naïve Bayes and SVM were tested separately. For both cases, the results indicated statistically

ACCEPTED MANUSCRIPT

significant differences between the compared methods over 11 initial labeled sets ($\chi_{NB}{}^2(3) = 19.909, p < 0.0001$) ($\chi_{SVM}{}^2(3) = 19.473, p < 0.0001$).

Table 8: Basic characteristics of the used original data sets

	Average rank	
	NB	SVM
Baseline	1.85	1.62
ST LDA	4.85	4.38
ST TF-IDF	2.23	2.23
EM-NB	2.69	3.08
EMC-NB	3.38	3.69

Since the overall Friedman test showed significant differences, we applied the Wilcoxon signed-rank test additionally on the accuracy results to compare ST LDA with other methods in a pairwise manner. Again, we compared methods separately on the accuracy results gained with Naïve Bayes and SVM on the final labeled sets. The results revealed that ST LDA performed significantly better than all other methods with the exception of the comparison between ST LDA with SVM classifier and EMC-NB (Table 9). The statistical significance of NBMN remained even after we conducted a Holm-Bonferroni correction and adjusted alpha levels ($0.003 < 0.0125; 0.003 < 0.0166; 0.003 < 0.025; 0.021 < 0.05$). However, after a Holm-Bonferroni adjustment, the ST LDA method with SVM as a supervised classifier did not provide significantly better results than the other compared methods ($0.003 < 0.0125; 0.004 < 0.0166; 0.05 > 0.025; 0.075 > 0.05$).

Table 9: Results of the Wilcoxon signed-rank test

	NBMN			*SVM*		
Comparison	*R+*	*R-*	*p-value*	*R+*	*R-*	*p-value*
ST LDA vs. Baseline	11	0	**0.003**	11	0	**0.003**
ST LDA vs. ST TF-IDF	11	0	**0.003**	10	1	**0.004**
ST LDA vs. EM-NB	11	0	**0.003**	8	3	0.050
ST LDA vs. EMC-NB	9	2	**0.021**	7	4	0.075

5. Discussion

While designing the parameter estimation model, the results implied an interesting hint. Similarity threshold, the most influential parameter, gives better results at lower values, which is contrary to our expectations. We believe that in the early stages, due to the smaller number of instances, knowledge is too modest to achieve higher accuracies. Furthermore, in particularly in the early stages, adding only a few instances in each iteration may lead to over-fitting.

ACCEPTED MANUSCRIPT

In comparison with other methods, ST LDA seems very stable when we vary the retaining ratio. The results indicate that it works well especially on small initial labeled sets where the differences to results of other methods are much greater than in the case of larger initial labeled sets. It can also be noted that ST LDA was outperformed by other methods (EMC-NB) on sets, where baseline methods (NB and SVM) performed relatively well. At the same time, these are also the smallest sets by the number of documents. This may indicate that ST LDA should be the method of choice especially when the baseline results are poor or there are a lot of documents in the collection.

As an indicator that the proposed method works well on small initial labeled sets, we can highlight the classification results on a 20 Newsgroups data set, which were presented within experiments with ssLDA (Wang et al., 2012). Those results showed that our method provides much better accuracy on small initial labeled sets than ssLDA on larger initial labeled sets. On the other hand, the main drawback of ST LDA is the time needed to build a classification model, where most of the execution time is consumed on a self-training algorithm.

6. Conclusion and Future Work

We showed that the ST LDA method can improve classification results on very small labeled sets. The method was tested on a variety of data sets with different imbalanced ratios and with different ratios of initial labeled sets. It was discovered that ST LDA with an NBMN classifier outperforms other compared methods and variations. This was additionally confirmed with the nonparametric statistical tests.

In addition to the algorithmic part, the method also included a model to determine the algorithm's input values based on given text collections. With the study on parameters we examined how much of an impact a particular parameter has on a method's accuracy. It was found that similarity threshold is the most important factor, followed by throttling and number of topics. Although we expected that the number of Gibbs iterations would also have a significant effect on classification results, this was not the case. We showed that for each collection, the throttling parameter value can be generalized with the regression model. With these findings we defined a model that can be used to calculate parameter values for any text collection.

Furthermore, we showed that compact text representation based on an LDA topic model can improve classification performance in a semi-supervised setting. The main reason why bag of words representation performs worse in such a setting, lies in data sparseness. Self-training is based on a comparison between labeled and unlabeled instances, which is less efficient in higher dimensions where differences between similar instances are getting less informative.

Nevertheless, there are also many possibilities for improvements, which will be addressed in our future work. First, we will investigate the representation based on topic models within other well defined semi-supervised learning methods. We will also extend our method with components that will be able to

ACCEPTED MANUSCRIPT

suggest unlabeled instances that are the most appropriate for labeling to prepare a better initial labeled set. Also, in terms of performing classification on a final labeled set we should consider classifiers described by authors like Gu et al. (Gu & Sheng, 2016), or Karakatič et al. (Karakatič & Podgorelec, 2016) and explore their impact on the final results.

Acknowledgments

The authors would like to acknowledge financial support from the Slovenian Research Agency (research core funding No. P2-0057).

References

Arun, R., Suresh, V., Madhavan, C. E. V., & Murty, M. N. (2010). On finding the natural number of topics with Latent Dirichlet Allocation: Some observations. In Mohamed J. Zaki, J. X. Yu, B. Ravindran, & V. Pudi (Eds.), *Advances in Knowledge Discovery and Data Mining* (pp. 391–402). Springer Berlin Heidelberg.

Beyer, K., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999). When is "nearest neighbor" meaningful? *Database Theory—ICDT'99*, (pp. 217–235).

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

Cao, J., Xia, T., Li, J., Zhang, Y., & Tang, S. (2009). A density-based method for adaptive LDA model selection. *Neurocomputing*, *72*, 1775–1781.

Castillo, P. A., Arenas, M. G., Rico, N., Mora, A. M., García-Sánchez, P., Laredo, J. L. J., & Merelo, J. J. (2012). Determining the significance and relative importance of parameters of a simulated quenching algorithm using statistical tools. *Applied Intelligence*, *37*, 239–254.

Chang, C.-c., & Lin, C.-j. (2011). LIBSVM : A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2*, 1–39.

Chapelle, O., Schölkopf, B., & Zien, A. (2010). *Semi-supervised learning*. (1st ed.). The MIT press.

Chen, K., & Wang, S. (2011). Semi-supervised learning via regularized boosting working on multiple semi-supervised assumptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*, 129–143.

Colace, F., De Santo, M., Greco, L., & Napoletano, P. (2014). Text classification using a few labeled examples. *Computers in Human Behavior*, *30*, 689–697.

ACCEPTED MANUSCRIPT

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, *41*, 391–407.

Demšar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *The Journal of Machine Learning Research*, *7*, 1–30.

Fu, Y., Yan, M., Zhang, X., Xu, L., Yang, D., & Kymer, J. D. (2015). Automated classification of software change messages by semi-supervised Latent Dirichlet Allocation. *Information and Software Technology*, *57*, 369–377.

Greene, D., O'Callaghan, D., & Cunningham, P. (2014). How Many Topics? Stability Analysis for Topic Models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 498–513). Springer Berlin Heidelberg.

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*, *101*, 5228–5235.

Gu, B., & Sheng, V. S. (2016). A Robust Regularization Path Algorithm for ν-Support Vector Classification. *IEEE Transactions on Neural Networks and Learning Systems*, (pp. 1–8).

Guzmán-Cabrera, R., Montes-Y-Gómez, M., Rosso, P., & Villaseñor-Pineda, L. (2009). Using the Web as corpus for self-training text categorization. *Information Retrieval*, *12*, 400–415.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *SIGKDD Explorations*, *11*, 10–18.

Han, E.-h., & Karypis, G. (2000). Centroid-Based Document Classification : Analysis & Experimental Results. *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, (pp. 424–431).

He, M., & Zheng, W. (2015). Using probabilistic topic models for document similarity computation. In *Computer Science and Applications: Proceedings of the 2014 Asia-Pacific Conference on Computer Science and Applications (CSAC 2014)* (p. 303). Shanghai, China: CRC Press.

Hofmann, T. (1999). Probabilistic Latent Semantic Analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence* (pp. 289–296). Morgan Kaufman.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning* (pp. 137–142). Springer Berlin Heidelberg.

Karakatič, S., & Podgorelec, V. (2016). Improved classification with allocation method and multiple classifiers. *Information Fusion*, *31*, 26–42.

ACCEPTED MANUSCRIPT

Ko, Y., & Seo, J. (2009). Text classification from unlabeled documents with bootstrapping and feature projection techniques. *Information Processing and Management*, *45*, 70–83.

Lu, Y., Mei, Q., & Zhai, C. (2011). Investigating task performance of probabilistic topic models: An empirical study of PLSA and LDA. *Information Retrieval*, *14*, 178–203.

McCallum, A. K. (2002). MALLET: A Machine Learning for Language Toolkit.

Minka, T., & Lafferty, J. (2002). Expectation-Propagation for the Generative Aspect Model. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence* (pp. 352–359). Morgan Kaufmann Publishers Inc.

Mitchell, T., & Blum, A. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory* (pp. 92–100). ACM.

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. (2000). Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, *39*, 103–134.

Niraula, N., Banjade, R., Ştefănescu, D., & Rus, V. (2013). Experiments with Semantic Similarity Measures based on LDA and LSA. In *Statistical Language and Speech Processing* (pp. 188–199). Springer Berlin Heidelberg.

Ocepek, U., Rugelj, J., & Bosnić, Z. (2015). Improving matrix factorization recommendations for examples in cold start. *Expert Systems with Applications*, *42*, 6784–6794.

Phan, X.-H., Nguyen, L.-M., & Horiguchi, S. (2008). Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-scale Data Collections. In *Proceeding of the 17th international conference on World Wide Web* (pp. 91–100). ACM.

Rao, Y., Xie, H., Li, J., Jin, F., Wang, F. L., & Li, Q. (2016). Social emotion classification of short text via topic-level maximum entropy model. *Information and Management*, *53*, 978–986.

Rubin, T. N., Chambers, A., Smyth, P., & Steyvers, M. (2012). Statistical topic models for multi-label document classification. *Machine Learning*, *88*, 157–208.

Siersdorfer, S., & Weikum, G. (2005). Automated retraining methods for document classification and their parameter tuning. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (pp. 478–486). Springer Berlin Heidelberg.

Sriurai, W. (2011). Improving text categorization by using a topic model. *Advanced Computing*, *2*, 21–27.

ACCEPTED MANUSCRIPT

Steyvers, M., & Griffiths, T. (2007). Probabilistic topic models. *Handbook of latent semantic analysis*, *427*, 424–440.

Triguero, I., García, S., & Herrera, F. (2015a). SEG-SSC : A Framework Based on Synthetic Examples Generation for Self-Labeled Semi-Supervised Classification. *IEEE Transactions on Cybernetics*, *45*, 622–634.

Triguero, I., García, S., & Herrera, F. (2015b). Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information Systems*, *42*, 245–284.

Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, *37*, 141–188.

Venugopalan, S., & Rai, V. (2015). Topic based classification and pattern identification in patents. *Technological Forecasting and Social Change*, *94*, 236–250.

Wang, D., Thint, M., & Al-Rubaie, A. (2012). Semi-Supervised Latent Dirichlet Allocation and Its Application for Document Classification. In *2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 03* (pp. 306–310). IEEE Computer Society.

Wang, P., Zhang, H., Wu, Y.-f., Xu, B., & Hao, H.-w. (2014). A Robust Framework for Short Text Categorization based on Topic Model and Integrated Classifier. In *International Joint Conference on Neural Networks (IJCNN)* (pp. 3534–3539). IEEE.

Wen, X., Shao, L., Xue, Y., & Fang, W. (2015). A rapid learning algorithm for vehicle classification. *Information Sciences*, *295*, 395–406.

Xie, H., Li, X., Wang, T., Lau, R. Y. K., Wong, T. L., Chen, L., Wang, F. L., & Li, Q. (2016). Incorporating sentiment into tag-based user profiles and resource profiles for personalized search in folksonomy. *Information Processing and Management*, *52*, 61–72.

Xie, P., & Xing, E. P. (2013). Integrating Document Clustering and Topic Modeling. In *Proceedings of the 29th conference on uncertainty in artificial intelligence* (pp. 694–703).

Yarowsky, D. (1995). Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics* (pp. 189–196). Association for Computational Linguistics.

Zhang, W., Tang, X., & Yoshida, T. (2015). TESC: An approach to Text classification using Semi-supervised Clustering. *Knowledge-Based Systems*, *75*, 152–160.

ACCEPTED MANUSCRIPT

Zhou, S., Li, K., & Liu, Y. (2009). Text categorization based on topic model. *International Journal of Computational Intelligence Systems*, *2*, 398–409.

Zhu, X., & Goldberg, A. B. (2009). *Introduction to Semi-Supervised Learning* volume 3. (1st ed.). Morgan and Claypool.

ACCEPTED MANUSCRIPT