Information Processing and Management 52 (2016) 217–257

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Optimization and label propagation in bipartite heterogeneous networks to improve transductive classification of texts

Rafael Geraldeli Rossi*, Alneu de Andrade Lopes, Solange Oliveira Rezende

Institute of Mathematics and Computer Science, University of São Paulo, Brazil

ARTICLE INFO

Article history:
Received 28 July 2014
Revised 22 April 2015
Accepted 6 July 2015
Available online 6 November 2015

Keywords:
Text classification
Transductive learning
Graph-based learning
Text mining
Label propagation
Bipartite heterogeneous network

ABSTRACT

Transductive classification is a useful way to classify texts when labeled training examples are insufficient. Several algorithms to perform transductive classification considering text collections represented in a vector space model have been proposed. However, the use of these algorithms is unfeasible in practical applications due to the independence assumption among instances or terms and the drawbacks of these algorithms. Network-based algorithms come up to avoid the drawbacks of the algorithms based on vector space model and to improve transductive classification. Networks are mostly used for label propagation, in which some labeled objects propagate their labels to other objects through the network connections. Bipartite networks are useful to represent text collections as networks and perform label propagation. The generation of this type of network avoids requirements such as collections with hyperlinks or citations, computation of similarities among all texts in the collection, as well as the setup of a number of parameters. In a bipartite heterogeneous network, objects correspond to documents and terms, and the connections are given by the occurrences of terms in documents. The label propagation is performed from documents to terms and then from terms to documents iteratively. Nevertheless, instead of using terms just as means of label propagation, in this article we propose the use of the bipartite network structure to define the relevance scores of terms for classes through an optimization process and then propagate these relevance scores to define labels for unlabeled documents. The new document labels are used to redefine the relevance scores of terms which consequently redefine the labels of unlabeled documents in an iterative process. We demonstrated that the proposed approach surpasses the algorithms for transductive classification based on vector space model or networks. Moreover, we demonstrated that the proposed algorithm effectively makes use of unlabeled documents to improve classification and it is faster than other transductive algorithms.

1. Introduction

Text automatic classification (TAC) is one of the most important tasks to manage, retrieve and extract knowledge from a huge number of textual documents (Manning, Raghavan, & Schütze, 2008; Nedjah, Mourelle, Kacprzyk, Frana, & de Souza, 2008; Berry & Castellanos, 2008; Li, Zhu, & Ogihara, 2008; He & Zhou, 2011; Uysal & Gunal, 2014). TAC automatically assigns a predefined category to a textual document.

* Corresponding author at: Department of Computer Science, Institute of Mathematics and Computer Science, University of São Paulo, Brazil. Tel.: +55 (16) 3373 9646; fax: +55 (16) 3373 9751.
E-mail address: ragero@icmc.usp.br (R.G. Rossi).

http://dx.doi.org/10.1016/j.ipm.2015.07.004
0306-4573/

Generally TAC is carried out by using inductive leaning algorithms (Weiss, Indurkhya, & Zhang, 2012; Sebastiani, 2002), which induce classification models to classify new or unseen texts. Usually a large number of labeled documents are necessary to induce an accurate classification model. Nevertheless, labeling texts is usually expensive and time consuming. Thus, a more practical approach is to employ methods which make use of the plenty of unlabeled texts available to perform and improve TAC.

Transductive approaches are widely used when labeled training data are insufficient. In this case, they make use of unlabeled data to improve classification performance (Kong, Ng, & Zhou, 2013; Chapelle, Schölkopf, & Zien, 2006; Belkin, Niyogi, & Sindhwani, 2006; Joachims,1999). Transductive classification directly estimates the labels of unlabeled instances without creating a model to classify new texts. Several algorithms considering texts represented in a vector space model have been developed to perform transductive classification such as Self-Training (Yarowsky,1995), Co-Training (Blum & Mitchell, 1998), Expectation Maximization (EM) (Nigam, McCallum, Thrun, & Mitchell, 2000), and Transductive Support Vector Machines (TSVM) (Joachims,1999). However, the use of these algorithms is unfeasible in practical applications due to the assumptions of these algorithms about the data distribution and computational cost. Moreover, the assumption that instances or terms are independent also impairs their classification performances.

Network-based algorithms came up to avoid the drawbacks of the algorithms based on vector space model and to improve transductive classification. Networks are mostly used for label propagation, in which some labeled objects propagate their labels to other objects through the network connections to perform transductive classification (Zhu & Goldberg, 2009; Rossi, Lopes, & Rezende, 2014; Subramanya & Bilmes, 2008; Zhou, Bousquet, Lal, Weston, & Schölkopf, 2004). Label propagation using just few labeled examples can obtain higher classification performance than inductive classification using a large number of labeled examples for TAC (Rossi, Lopes, & Rezende, 2014). Moreover, the use of networks to model text collections allows extracting patterns which are not extracted by algorithms based on vector-space model (VSM) (Breve, Zhao, Quiles, Pedrycz, & Liu, 2012).

Text collections are modeled as networks using homogeneous or heterogeneous networks. Homogeneous networks contain objects of a single type and heterogeneous networks are compounded by objects of different types. Document homogeneous networks have been used to model text collections as networks for label propagation (Jebara, Wang, & Chang, 2009; Kim, Pantel, Duan, & Gaffney, 2009; Subramanya & Bilmes, 2008; Wang & Zhang, 2006; Castillo, Donato, Gionis, Murdock, & Silvestri, 2007; Zhou et al., 2004; Zhu, Ghahramani, & Lafferty, 2003). In such networks, documents propagate their labels directly to other documents. Documents are connected according to hyperlinks, citations or similarities. The use of just hyperlinks and citations to build document networks reduces the quality of classification (Angelova & Weikum, 2006) and limits the application domains. On the other hand, documents wired considering similarity have been applied since they model any type of text collections and improve the classification quality (Angelova & Weikum, 2006). However, computing similarities poses a high computational cost, and the parameters such as minimum similarity or number of neighbors, significantly impact the classification accuracy (de Sousa, Rezende, & Batista, 2013).

Bipartite networks have come up as an alternative to model text collections as networks (Rossi, Faleiros, Lopes, & Rezende, 2012; Rossi, Lopes, Faleiros, & Rezende, 2014; Rossi, Lopes, & Rezende, 2014), in which objects correspond to documents and terms. Terms are linked to documents in which they are present. This network is easily generated, since there is no need to set parameters or compute similarities. Moreover, it has provided promising results for text classification (Rossi et al., 2012; Rossi, Lopes, Faleiros, et al., 2014: Rossi, Lopes, & Rezende, 2014). In such networks, documents propagate their labels to terms and then the terms propagate their labels to documents.

Instead of using the bipartite network structure just as means to propagate labels, this structure can be used to set the relevance scores of terms for classes, i.e., how much the presence of a term in a document increases or decreases the probability of a document belonging to a class. In (Rossi et al., 2012; Rossi, Lopes, Faleiros, et al., 2014) the relevance scores of terms for classes are induced using the bipartite network structure. These relevance scores were used to classify new/unseen documents, providing accuracies higher than state-of-the-art algorithms. However, scenarios with only few labeled documents impair the induction of term scores and consequently the classification accuracy.

In this paper we propose an algorithm to set the relevance scores of terms for classes considering labeled and unlabeled documents represented in a bipartite heterogeneous network. The relevance scores are obtained through an optimization process considering the current labels of the documents. The obtained relevance scores are propagated to define the new labels to unlabeled documents. Optimization and label propagation are repeated iteratively until converge, i.e., until the labels assigned to unlabeled documents do not change. The proposed algorithm, named TCBHN (*Transductive Classification based on Bipartite Heterogeneous Network*) obtains better classification performance and is faster than transductive algorithms based on vector space model or networks.

The main contributions of this article are fivefold:

- We propose a transductive classification algorithm which effectively makes use of unlabeled data to improve text classification.
- We propose a scalable transductive classification algorithm which makes use of bipartite networks to perform transductive classification.

- We propose an algorithm which surpasses the classification performance of state-of-the-art algorithms based on vector space model or networks.
- We conduct a rigorous comparative evaluation of the proposed classification algorithm with traditional and state-of-the-art algorithms based on vector space model and networks. The evaluation carried out in this article allows to highlight the drawbacks of the existing transductive classification algorithms and to highlight the advantages of the proposed algorithm. We also present the behavior of the algorithms for a different range of labeled documents.
- We present a trade-off between inductive classification and transductive classification. We analyse the differences between inductive classification and transductive classification considering classification evaluation measures and classification time.

The remainder of this paper is organized as follows. Section 2 presents background and related works about transductive classification, texts represented by networks, and the use of bipartite networks to induce relevance scores of terms for classes. Section 3 presents details on the proposed algorithm for transductive classification of texts using bipartite networks. Section 4 presents details of the experimental evaluation and the results. Finally, Section 5 presents the conclusions and points to future work.

2. Background and related work

In this section we present the notations and computational structures to perform transductive learning. We standardize both of them to be used in transductive learning on vector space model and networks. Next, we detail the algorithms which perform transductive learning on vector space model or networks, present their pseudocodes, drawbacks, and how to generate networks from text collections to be used as input to network-based algorithms. We also present how to induce the class information of terms in text collections in a scenario with all labeled documents.

2.1. General notations and structures to perform transductive learning

Let $\mathcal{C} = \{c_1, c_2, \ldots, c_l\}$ represent the set of class labels, let $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$ be the set of terms, and let $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$ be the set of documents of a text collection. In a transductive learning scenario, $\mathcal{D} = \mathcal{D}^L \cup \mathcal{D}^U$, in which $\mathcal{D}^L$ represents the set of labeled documents and $\mathcal{D}^U$ represent the set of unlabeled documents. Finally, let $\mathcal{Y} = \{y_1, y_2, \ldots, y_{|\mathcal{D}|}\}$ be the labels of the labeled documents and the labels assigned during the classification process, i.e., $\mathcal{Y} = \mathcal{Y}^L \cup \mathcal{Y}^U$.

Differently from supervised inductive classification, which aims to create a classification model to approximate a real category assignment function $R : \mathcal{D}^L \rightarrow \mathcal{Y}^L$, the goal of transductive learning is to find an admissible function $F : \mathcal{D}^{L+U} \rightarrow \mathcal{Y}^{L+U}$, in which the unlabeled data are used to improve classification performance. In practice, transductive learning assigns weights or relevance scores to documents for each one of the classes and the documents are classified considering these weights. In order to do so, let $\mathbf{f}_{d_i} = \{f_{c_1}, f_{c_2}, \ldots, f_{c_{|\mathcal{C}|}}\}$ be the weight vector of a document d_i which stores the weights of a document d_i for all classes in $\mathcal{C}$. Hence it is also referred to as class information vector. Let $\mathbf{F}(\mathcal{D}) = \{\mathbf{f}_{d_1}, \mathbf{f}_{d_2}, \ldots, \mathbf{f}_{d_{|\mathcal{D}|}}\}^T$ be a matrix which stores all the weight vectors of the documents. The values of the vector $\mathbf{f}$ can be binary, probabilities or real values.

The predefined labels for a document $d_i \in \mathcal{D}^L$ are stored in a weight vector $\mathbf{y}_{d_i} = \{y_1, y_2, \ldots, y_{|\mathcal{C}|}\}$, which has the value 1 in the position corresponding to the class of the document d_i and 0 to the others. The predefined labels of all labeled documents are stored in a matrix $\mathbf{Y}(\mathcal{D}^L) = \{\mathbf{y}_{d_1}, \mathbf{y}_{d_2}, \ldots, \mathbf{y}_{d_{|\mathcal{D}^L|}}.\}^T$. Most of the transductive classification algorithms restrict that $\mathbf{F}(\mathcal{D}^L) = \mathbf{Y}$, i.e., there is no change in the class information of labeled documents (Yarowsky, 1995; Blum & Mitchell, 1998; Joachims, 1999; Nigam et al., 2000; Zhu et al., 2003). However, some of them relax this restriction and allow to change the class information of labeled examples during classification (Zhou et al., 2004; Yin, Li, Mei, & Han, 2009, Ji et al., 2010).

Some algorithms need to obtain the weights of terms for classes to infer the class information of unlabeled documents. Thus, a structure to store class information from terms are also required. We used $\mathbf{f}_{t_j} = \{f_{c_1}, f_{c_2}, \ldots, f_{c_{|\mathcal{C}|}}\}$ to represent the class information of terms for classes and the matrix $\mathbf{F}(\mathcal{T}) = \{\mathbf{f}_{t_1}, \mathbf{f}_{t_2}, \ldots, \mathbf{f}_{t_{|\mathcal{T}|}}\}^T$ to store all class information of the terms.

The weights of the links among network objects are stored in a matrix $\mathbf{W}$. Link weights among a document d_i and other documents are represented by a vector $\mathbf{w}_{d_i} = \{w_{d_1}, w_{d_2}, \ldots, w_{d_{|\mathcal{D}|}}\}$. The same vector $\mathbf{w}_{d_i}$ is used to denote the link weights among a document d_i and its terms. In this case, $\mathbf{w}_{d_i} = \{w_{t_1}, w_{t_2}, \ldots, w_{t_{|\mathcal{T}|}}\}$. This vector is also used to represent a document d_i in a vector space model and store the frequency, or other frequency-based measure, of terms in the document d_i.

2.2. Transductive learning on vector space model

The first researches about transductive learning for text classification consider text collections represented in vector space model (Yarowsky, 1995; Blum & Mitchell, 1998; Joachims, 1999; Nigam et al., 2000). Usually a bag-of-words is used to represent the text collection, in which each document is represented by a vector and each dimension of the vector corresponds to a single word of the collection. The values in the vectors are based on the frequency of a term in a document, such as binary weights, term frequency (*tf*) or term frequency – inverse document frequency (*tf–idf*) (Salton, 1989).

Traditional and state-of-the-art transductive algorithms based on vector space model are (Zhu & Goldberg, 2009; Chapelle et al., 2006): Self-Training, Co-Training, Expectation Maximization, and Transductive Support Vector Machines. There are also some combinations or variations of these algorithms to perform transductive learning. In the next subsections we detail the algorithms based on vector space model mentioned above.

2.2.1. Self-Training

Considering texts represented in a vector space model, perhaps the most natural way to perform transductive learning is through Self-Training (Culp & Michailidis, 2008; Haffari & Sarkar, 2007; Yarowsky,1995). In this approach, initially the labeled documents are used to induce a classification model through supervised inductive learning. Any inductive algorithm, such as Multinomial Naive Bayes, Transductive Support Vector Machine, or k-Nearest Neighbor can be used to induce the classification model. This model is used to classify the unlabeled documents and the X most confident classified documents are added to the set of labeled documents. Then, the classification model is reinduced considering the new set of labeled documents. This process is repeated until all unlabeled documents were added to the set of labeled documents. Algorithm 1 presents the pseudocode of the Self-Training approach.

Algorithm 1. Self-Training.

Input : $\mathcal{D}^L$,$\mathcal{D}^U$, $\mathbf{W}$, $\mathbf{Y}$,
X - number of unlabeled documents to be include as labeled documents at each iteration
Output: $\mathbf{F}(\mathcal{D}^U)$

1 $\mathcal{D}^R = \mathcal{D}^U$ /* Copy of docs in $\mathcal{D}^U$ which is used in the iterative process */
2 **repeat**
3 $Classification\_Model = Inductive\_Learning(\mathcal{D}^L, \mathbf{W}, \mathbf{Y})$ /* Classification model induction considering labeled documents */
4 $\mathbf{F}(\mathcal{D}^R) = Classification\_Model(\mathcal{D}^R, \mathbf{W})$ /* Classification confidences for unlabeled documents */
5 $\mathcal{S} = Most\_Confident(\mathbf{F}(\mathcal{D}^R), X)$ /* $\mathcal{S}$ contains the X unlabeled document with the highest classification confidences */
6 **foreach** $d_i \in \mathcal{S}$ **do**
7 $\mathbf{f}_{d_i} = Arg\_Max(\mathbf{f}_{d_i})$ /* Defining the class of d_i considering the arg-max value of $\mathbf{f}_{d_i}$ */
8 **end**
9 $\mathcal{D}^R = \mathcal{D}^R - \mathcal{S}$ /* Remove from $\mathcal{D}^R$ the documents in $\mathcal{S}$ */
10 $\mathcal{D}^L = \mathcal{D}^L \cup \mathcal{S}$ /* Insert in $\mathcal{D}^L$ the documents in $\mathcal{S}$ */
11 **until** $\mathcal{D}^R = \emptyset$

The assumption of Self-Training is that the most confident classifications are correct. However, this is difficult to hold in practice and is true just when the classes are well separable in vector space model (Zhu & Goldberg, 2009). Moreover, mistakes in the most confident classifications degrade the classification performance in the next iterations.

Self-Training have to induce $|\mathcal{D}^U|/X$ classification models and may not scale well for small values of X. On the other hand, high values of X will practically perform an inductive supervised learning, since most of the unlabeled documents will be classified through an model induced considering only labeled documents.

Despite generating classification confidences to sort unlabeled documents, the most confident documents are inserted into the set of labeled documents considering the value 1 for the class with the highest confidence and 0 for others, i.e., Self-Training performs a hard classification of unlabeled data.

2.2.2. Co-Training

Co-Training is an extension of the Self-Training approach for text collections with two views (Blum & Mitchell, 1998). In a simplified version of Co-Training presented in (Zhu & Goldberg, 2009), the X most confident classifications in one view are added as labeled examples in the other view. Similarly to Self-Training, this process is repeated until all unlabeled documents be added to the set of labeled documents.

The assumptions of Co-Training are: (i) there are two views to represent the text collections; (ii) the views are independent to each other; and (iii) each view is able to induce an accurate classification model by itself. Nevertheless, these assumptions usually do not hold in practice. Moreover, Co-Training presents the same drawbacks of Self-Training such as degradation in classification performance due to error propagation (Laguna & de Andrade Lopes, 2010), and the cost to induce a classification model and sort documents by classification confidence repeatedly.

For collections which do not have two views, Co-Training is performed by splitting the feature space into two disjunct sets (Xu, Tao, & Xu, 2013; Li, Meng, Cao, & Sun, 2009; Bickel & Scheffer, 2004), which usually does not significantly improves results, or as performed in (Laguna & de Andrade Lopes, 2010). In the latter, the two views are produced by using two

different k-Nearest Neighbor classifiers with different biases. These distinct biases allow the effective cooperation between the two classifiers in the Co-Training learning phase. In Algorithm 2 we present the pseudocode of the simplified version of the Co-Training approach (Zhu & Goldberg, 2009) for a bag-of-words splitted into two views.

Algorithm 2. Co-Training.

Input : $\mathcal{D}^{L1}, \mathcal{D}^{L2}, \mathcal{D}^{U1}, \mathcal{D}^{U2}, \mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{Y}$,
X - number of unlabeled documents to be include as labeled documents at each iteration
Output: $\mathbf{F}(\mathcal{D}^{U})$

$\mathcal{D}^{R1} = \mathcal{D}^{R1}$ /* Copy of docs in both views of D^U which is used in the iterative process */
$\mathcal{D}^{R2} = \mathcal{D}^{R2}$
repeat
  $Classification\_Model\_View1 = Inductive\_Learning(\mathcal{D}^{L1}, \mathbf{W}^{(1)}, \mathbf{Y})$ /* Classification model induction considering labeled documents from the 1^{st} view */
  $Classification\_Model\_View2 = Inductive\_Learning(\mathcal{D}^{L2}, \mathbf{W}^{(2)}, \mathbf{Y})$ /* Classification model induction considering labeled documents from the 2^{nd} view */
  $\mathbf{F}(\mathcal{D}^{R1}) = Classification\_Model\_View1(\mathcal{D}^{R1}, \mathbf{W}^{(1)})$ /* Classification confidences for unlabeled documents in the 1^{st} view */
  $\mathbf{F}(\mathcal{D}^{R2}) = Classification\_Model\_View2(\mathcal{D}^{R2}, \mathbf{W}^{(2)})$ /* Classification confidences for unlabeled documents in the 2^{nd} view */
  $\mathcal{S}^1 = Most\_Confident(F(\mathcal{D}^{R1}), X)$ /* $\mathcal{S}^1$ contains the X unlabeled document with the highest classification confidences in the 1^{st}view */
  foreach $d_i \in \mathcal{S}^1$ **do**
    $\mathbf{f}_{d_i} = Arg\_Max(\mathbf{f}_{d_i})$ /* Defining the class of d_i considering the arg-max value of $\mathbf{f}_{d_i}$ */
  end
  $\mathcal{S}^2 = Mais\_Confident(F(\mathcal{D}^{R2}), X)$ /* $\mathcal{S}^2$ contains the X unlabeled document with the highest classification confidences in the 2^{nd}view */
  foreach $d_i \in \mathcal{S}^2$ **do**
    $\mathbf{f}_{d_i} = Arg\_Max(\mathbf{f}_{d_i})$
  end
  $\mathcal{D}^{R1} = \mathcal{D}^{R1} - \mathcal{S}^1$ /* Remove from $\mathcal{D}^{U1}$ the documents in $\mathcal{S}^1$ */
  $\mathcal{D}^{R2} = \mathcal{D}^{R2} - \mathcal{S}^2$ /* Remove from $\mathcal{D}^{U2}$ the documents in $\mathcal{S}^2$ */
  $\mathcal{D}^{L1} = \mathcal{D}^{L1} \cup \mathcal{S}^1$ /* Insert in $\mathcal{D}^{L1}$ the documents in $\mathcal{S}^1$ */
  $\mathcal{D}^{L2} = \mathcal{D}^{L1} \cup \mathcal{S}^2$ /* Insert in $\mathcal{D}^{L2}$ the documents in $\mathcal{S}^2$ */
until $\mathcal{D}^{R1} = \emptyset$ *and* $\mathcal{D}^{R2} = \emptyset$

2.2.3. Expectation maximization

The Expectation Maximization (EM) approach allows to assign class information to unlabeled documents iteratively without inducing classification models repeatedly. EM performs a soft classification on unlabeled data, i.e., classification confidences are assigned to unlabeled documents and these confidences are used in the next iteration of the approach. EM performs hill climbing search to estimate maximum a posteriori probability in problems with incomplete data (Dempster, Laird, & Rubin, 1977).

Nigam et al. (2000) presents an instantiation of the EM approach for transductive classifications of textual documents. In this work, the authors consider that each text is generated by one or more mixture components and each mixture component is associated with one class. A component for a class c_j corresponds to the class information of terms for the class c_j ($\mathbf{F}(\mathcal{T})$). Thus the goal of EM is to obtain the mixture components for each class using labeled and unlabeled documents.

Nigam et al. (2000) combines EM with Multinomial Naive Bayes to perform transductive classification. The class information of a document d_i for a class c_j according to Bayes rule is:

$$f_{d_i,c_j} = P(c_j|d_i) = \frac{P(c_j)P(d_i|c_j)}{P(d_i)}. \quad (1)$$

$P(c_j)$ is the probability of occurrence of the class c_j considering the class information of the documents. Therefore, $P(c_j)$ using Laplace estimator is:

$$P(c_j) = \frac{1 + \sum_{d_i \in \mathcal{D}} P(c_j|d_i)}{|\mathcal{C}| + |\mathcal{D}|}. \quad (2)$$

$P(d_i|c_j)$ is the probability of occurrence of a document d_i given a class c_j. The naive Bayes assumption states that the words of a document are generated independently of each other and of its position in the document. Therefore, the probability of occurrence of a document d_i given a class c_j is given by the probability of occurrence of its terms for a class c_j, i.e.:

$$P(d_i|c_j) = \prod_{t_k \in \mathcal{T}, w_{d_i,t_k} > 0} P(t_k|c_j). \tag{3}$$

The probability of occurrence of a term t_i in a class c_j given the current class information of the documents using Laplace estimator is:

$$f_{t_i,c_j} = P(t_i|c_j) = \frac{1 + \sum_{d_k \in \mathcal{D}} w_{t_i,d_k} P(c_j|d_i)}{|\mathcal{T}| + \sum_{t_l \in \mathcal{T}} \sum_{d_m \in \mathcal{D}} w_{t_l,d_m} P(c_j|d_i)}. \tag{4}$$

$P(d_i)$ is the normalization term, i.e.,

$$P(d_i) = \sum_{c_j \in \mathcal{C}} P(c_j) P(d_i|c_j). \tag{5}$$

Algorithm 3 presents the pseudocode of the EM approach for transductive text classification (Nigam et al., 2000). The computation of the class information of terms (line 3) is called E-step and the estimation of the class information of terms (line 4) is called M-step. E-Step and M-step are repeated until convergence, i.e., until the class information of labeled documents do not change too much in consecutive iterations, or until a fixed number of iterations.

Algorithm 3. Expectation Maximization.

Input : $\mathcal{D}^L$,$\mathcal{D}^U$,$\mathbf{F}$, $\mathbf{Y}$
Output: $\mathbf{F}(\mathcal{D}^U)$

1 $\mathbf{F}(\mathcal{D}^L) = \mathbf{Y}$
2 $\mathbf{F}(\mathcal{T}) = Estimate\_Prob\_Terms(\mathcal{D}^L, \mathbf{W}, \mathbf{F}(\mathcal{D}^L))$ /* Estimate the class information of the terms using Equation 4 and considering only labeled documents */
3 **repeat**
4 | **E-Step**: $\mathbf{F}(\mathcal{D}^U) = Estimate\_Prob\_Doc(\mathcal{T}, \mathbf{W}, \mathbf{F}(\mathcal{T}))$ /* Estimate the class information of the unlabeled documents using Equation 1 and considering the current class information from terms */
5 |
6 | **M-Step**: $\mathbf{F}(\mathcal{T}) = Estimate\_Prob\_Terms(\mathcal{D}^L, \mathcal{D}^U, \mathbf{W}, \mathbf{F}(\mathcal{D}))$ /* Estimate the class information of the terms using Equation 4 and considering the current class information of labeled and unlabeled documents */
7 **until** *convergence or fixed number of iterations*

The EM approach presented above can be extended or improved considering (Nigam et al., 2000): (i) weighting the importance of unlabeled documents in the computation of $\mathbf{F}(\mathcal{T})$ and (ii) considering multiples components for each class. In the first case a function $\Lambda(d_i)$, presented in Eq. (6), is used to weight $P(c_j|d_i;\mathbf{F}(\mathcal{T}))$ in Eqs. (2) and (4).

$$\Lambda(d_i) = \begin{cases} \lambda & \text{if } d_i \in \mathcal{D}^U \text{ (parameter set by the user)} \\ 1 & \text{if } d_i \in \mathcal{D}^L. \end{cases} \tag{6}$$

The use of multiple components for each class implies the use of EM to estimate the parameters of each component. Then, the class information of a document d_i for a class c_j considers the sum of the probability to occur d_i in each one of the components from class c_j.

EM classification is not accurate if the generative assumption is violated. A mix of Co-Training and EM is also found in literature to perform transductive classification in text collections with two views (Ghani, 2002; Nigam & Ghani, 2000).

2.2.4. Transductive support vector machines

Transductive Support Vector Machines (TSVM) (Vapnik, 1998; Joachims,1999) considers labeled and unlabeled documents to obtain a maximal margin hyperplane. It is a transductive version of the well-known Support Vector Machines. The coefficients of a hyperplane correspond to the class information of terms. Considering a binary classification problem, i.e., f_{d_i} and y_{d_i} are variables (or vectors with one dimension), in which y_{d_i} or $f_{d_i} = \{-1, 1\}$, the class information of a document d_i in TSVM is:

$$f_{d_i} = \begin{cases} +1, & \text{if } \sum_{t_k \in \mathcal{T}} f_{t_k} \cdot w_{d_i,t_k} + b > 0; \\ -1 & \text{if } \sum_{t_k \in \mathcal{T}} f_{t_k} \cdot w_{d_i,t_k} + b < 0 \end{cases} \tag{7}$$

Let $\mathcal{H}$ be the set of possible hyperplanes. Let $H_i \in \mathcal{H}$ be a hyperplane and let H_i^+ be the parallel hyperplane closest to H_i and a positive document d_i located on the hyperplane H_i^+, i.e.,

$$\sum_{t_k \in \mathcal{T}} f_{t_k} \cdot w_{d_i,t_k} + b = 0, \;\; f_{d_i} = +1. \tag{8}$$

Let H_i^- be the parallel hyperplane closest to H_i and a negative document d_j located on the decision boundary H_i^-, i.e.,

$$\sum_{t_k \in \mathcal{T}} f_{t_k} \cdot w_{d_j,t_k} + b = 0, \;\; f_{d_j} = -1. \tag{9}$$

Rescaling the hyperplane coefficients and b to make the optimization problem easier, H_i^+ and H_i^- are:

$$H_i^+ : \sum_{t_k \in \mathcal{T}} f_{t_k} \cdot w_{d_i,t_k} + b = 1, \;\; f_{d_i} = +1 \tag{10}$$

$$H_i^- : \sum_{t_k \in \mathcal{T}} f_{t_k} \cdot w_{d_j,t_k} + b = -1, \;\; f_{d_j} = -1. \tag{11}$$

Therefore the goal of TSVM is to induce the hyperplane coefficients and b to satisfy

$$\sum_{t_k \in \mathcal{T}} f_{t_k} \cdot w_{d_i,t_k} + b \geq 1 \text{ if } f_{d_i} = +1, \tag{12}$$

and

$$\sum_{t_k \in \mathcal{T}} f_{t_k} \cdot w_{d_j,t_k} + b \leq -1 \text{ if } f_{d_j} = -1. \tag{13}$$

The Eqs. (12) and (13) can be summarized by:

$$f_{d_i}\left(\sum_{t_k \in \mathcal{T}} f_{t_k} \cdot w_{d_j,t_k} + b\right) \geq 1, \;\; \forall d_i \in \mathcal{D}. \tag{14}$$

Therefore, scoring correctly the hyperplane coefficients and b depends on assigning values to f_{d_i} correctly.

The documents located on the margins are called support vectors. The Euclidean distance between the hyperplanes H_i^+ and H_i^- is referred to as margin. Let $dist(d_i, d_j)$ the distance between two points in which d_i is located on H_i^+ and d_j is on H_i^-. Then the margin is computed subtracting Eq. (9) from Eq. (8), i.e. (Tan, Steinbach, & Kumar, 2005)

$$\begin{aligned} &\mathbf{f} \cdot (d_i - d_j) = 2 \\ &\|\mathbf{f}\| \times dist(d_i, d_j) = 2 \\ &\therefore \;\; dist(d_i, d_j) = \frac{2}{\|\mathbf{f}\|} \end{aligned} \tag{15}$$

The optimization carried out by TSVM aims to maximize the margin or minimize

$$\min_{\mathbf{f},b} \frac{1}{2}\|\mathbf{f}\|^2 \tag{17}$$

$$\begin{aligned} \text{Subject to :} \quad & y_{d_i}\left(\sum_{t_k \in \mathcal{T}} f_{t_k} \cdot w_{d_j,t_k} + b \geq 1\right), \;\; \forall d_i \in \mathcal{D}^L \\ & f_{d_j}\left(\sum_{t_k \in \mathcal{T}} f_{t_k} \cdot w_{d_j,t_k} + b \geq 1\right), \;\; \forall d_j \in \mathcal{D}^U \\ & y_{d_i}, \;\; f_{d_j} \in \{-1, +1\}, \;\; \forall d_i \in \mathcal{D}^L, \;\; \forall d_j \in \mathcal{D}^U \end{aligned}$$

As in SVM, TSVM also allows the documents to be in the wrong side of the hyperplane to obtain hyperplanes with higher margins. To do so, the slack variables ξ are used in the TSVM optimization. Then, the function to be minimized with slack variables is

$$\min_{\mathbf{f},b} \frac{1}{2}\|\mathbf{f}\|^2 + C\sum_{d_i \in \mathcal{D}^L} \xi_{d_i} + C'\sum_{d_j \in \mathcal{D}^U} \xi_{d_j} \tag{18}$$

$$\begin{aligned} \text{Subject to :} \quad & y_{d_i}\left(\sum_{t_k \in \mathcal{T}} f_{t_k} \cdot w_{d_j,t_k} + b \geq 1 - \xi_{d_i}\right), \;\; \forall d_i \in \mathcal{D}^L \\ & f_{d_j}\left(\sum_{t_k \in \mathcal{T}} f_{t_k} \cdot w_{d_j,t_k} + b \geq 1 - \xi_{d_j}\right), \;\; \forall d_j \in \mathcal{D}^U \\ & y_{d_i}, \;\; f_{d_j} \in \{-1, +1\}, \;\; \forall d_i \in \mathcal{D}^L, \;\; \forall d_j \in \mathcal{D}^U \\ & \xi_{d_i} \geq 0, \;\; \forall d_i \in \mathcal{D}^L \\ & \xi'_{d_j} \geq 0, \;\; \forall d_j \in \mathcal{D}^U, \end{aligned}$$

where the parameters C and C' allow a trade-off between classification error and margin size considering labeled and unlabeled documents respectively.

The restriction that f_{d_i} contains only integer values makes the TSVM a non-convex optimization problem, and the global optimum solution is feasible only for few unlabeled documents (Joachims, 1999; Chapelle et al., 2006). Local search or a relaxation in the optimization problem is needed to make TSVM feasible for a large number of unlabeled documents. In Algorithm 4 we present the local search solution for TVSM used for transductive classification of texts proposed by Joachims (1999). In this algorithm, initially a hyperplane is induced considering only labeled examples using SVM. The unlabeled documents are classified according to the induced hyperplane, and two documents which falls on the wrong side of the hyperplane has their labels changed. Then a hyperplane is induced considering labeled documents and the labels assigned to unlabeled documents. The hyperplane induction and the change of labels among misclassified document are repeated until convergence, i.e., until the labels of unlabeled documents do not change too much, or a fixed number of iterations. Changing the labels of misclassified documents strictly improves the classification in the next step. Moreover, the algorithm starts with a small value of C' and rises it during the iterations. This allows more documents to be on the wrong side of the hyperplane in the first iterations and less documents in the last iterations.

Algorithm 4. Transductive Support Vector Machines (Binary Classification).

Input : $\mathcal{O}^L$,$\mathcal{O}^U$, $\mathbf{W}$, $\mathbf{Y}$,
C - SVM's parameter to induce a maximal margin hyperplane considering just labeled documents,
C' - Maximal value of the parameter C to induce a maximal margin hyperplane in the iterative process considering labeled and unlabeled documents,
$num+$ - number of positive documents
Output: $\mathbf{F}(\mathcal{D}^U)$

$Classification\_Model = SVM(\mathcal{D}^L, \mathbf{W}, \mathbf{Y}, C)$ /* Inducing the maximal margin hyperplane considering just labeled documents */
foreach $d_i \in D^U$ **do**
  $\mathbf{f}_{d_i} = Classification\_Model(\mathbf{w}_{d_i})$ /* Assigning labels to unlabeled documents */
end
/* A proportion of num+ test example are classified as positive if the user opts to used a balanced classification */
$C' = 10^{-5}$
$C'' = 10^{-5} * num+ / (|D^U| - num+)$
$change = false$
while $(C'_- < C')$ *or* $(C'_- < C')$ *or* $(change == false)$ *or fixed number of iterations* **do**
  $C = C'_- + C'_+$
  $Classification\_Model = SVM(\mathcal{D}^L, \mathcal{D}^U, \mathbf{W}, \mathbf{F}, C)$ /* Inducing the maximal margin hyperplane using the labels assigned to unlabeled documents */
  $change = false$
  $ErrorList = \emptyset$ /* List to store the misclassified documents */
  foreach $d_i \in \mathcal{D}^U$ **do**
    $\mathbf{f}'_{d_i} = Classification\_Model(\mathbf{w}_{d_i})$
    if $\mathbf{f}'_{d_i} \neq \mathbf{f}_{d_i}$ **then**
      $ErrorList = ErrorList + d_i$
    end
  end
  foreach $d_i \in ErrorList$ **do**
    /* Changing the labels of misclassified documents */
    foreach $d_j \in ErrorList, d_i \neq d_j$ **do**
      if $\mathbf{f}_{d_i} \neq \mathbf{f}_{d_j}$ **then**
        $\mathbf{f}' = \mathbf{f}_{d_i}, \mathbf{f}_{d_i} = \mathbf{f}_{d_j}, \mathbf{f}_{d_j} = \mathbf{f}'$
        $ErrorList = ErrorList - d_j$
        $change = true$;
      end
    end
  end
  $C'_- = min(2 * C'_-, C')$ /* Increase C'_- and C'_+ */
  $C'_+ = min(2 * C'_+, C')$
end

TSVM has the assumption that the classes are well-separated, so that the hyperplane with maximal margin falls into a low density region. When this assumption does not hold, the transductive classification obtained by TSVM is not accurate.

Despite the fact that to obtain local minima solutions be fast, TSVM is still not scalable for large datasets since it needs to build classification models repeatedly. Moreover, TSVM may present unbalanced solutions, i.e., most of the documents are assigned to a single class. In this case, a function is used to maintain the class distribution of the unlabeled documents the same as in the labeled documents (Chapelle et al., 2006).

2.3. Transductive learning on networks

Network-based representation is a natural and direct way to represent textual data for different tasks. The networks presented in this article are defined by $N = \langle \mathcal{O}, \mathcal{E}, \mathcal{W} \rangle$, in which $\mathcal{O}$ represents the set of objects (also called vertices or nodes), $\mathcal{E}$ represents the set of connections (also called relations or links) among the objects, and $\mathcal{W}$ represents the weights of the connections. The objects correspond to entities of a problem. In case of text collections, examples of objects are documents, terms, sentences or authors. When $\mathcal{O}$ is compounded by a single type of object, the network is called homogeneous network. When $\mathcal{O}$ is compounded by h different types of objects, i.e, $\mathcal{O} = \mathcal{O}_1 \cup \cdots \cup \mathcal{O}_h$ and $\mathcal{O}_i$ represents the objects of the i-th type, the network is called heterogeneous network (Sun & Han, 2012).

Different types of objects and different relations can be used to generate a network-based representation. We can extract objects representing the entire collection, as documents and terms, or representing pieces of a text, as terms, sentences or paragraphs. Documents are connected according to (i) "explicit relations" as hyperlinks or citations (Oh, Myaeng, & Lee, 2000; Page, Brin, Motwani, & Winograd, 1998; Mei, Cai, Zhang, & Zhai, 2008; Sun, Han, Gao, & Yu, 2009) or (ii) considering similarity (Angelova & Weikum, 2006; de Sousa et al., 2013). Terms are connected (i) if they precede or succeed each other in a text (Aggarwal & Zhao, 2013; Markov & Last, 2006), (ii) if they co-occur in pieces of texts as sentences/windows (Solé, Corominas-Murtra, Valverde, & Steels, 2010; Palshikar, 2007; Mihalcea & Tarau, 2004) or in the text collection (Wang, Do, & Lin, 2005; Tseng, Ho, Yang, & Chen, 2012; Matsuo, Sakaki, Uchiyama, & Ishizuka, 2006), or (iii) if they present syntactic/semantic relationship (Solé et al., 2010; Steyvers & Tenenbaum, 2005). Sentences or paragraphs are connected considering (i) similarities (Salton, Singhal, Mitra, & Buckley, 1997; Yang & Soo, 2012) or (ii) considering semantic similarity, co-reference resolution and discourse relations (Ferreira et al., 2013).

A combination of different types of objects is also used. Dhillon (2001), Rossi et al. (2012), and Rossi, Lopes, Faleiros, et al. (2014) use documents and terms to generate a bipartite network. In these cases, terms are connected to documents in which they occur. Wan, Yang, and Xiao (2007) uses sentences and words as network objects. In this case, terms are connected to sentences in which they occur, sentences are connected to each other considering the number of shared words, and terms are connected to each other considering the number of shared sentences.

The main algorithms for transductive classification in data represented as networks are based on regularization (Zhu & Goldberg, 2009), which have to satisfy two assumptions: (i) the class information of neighbors must be close and (ii) the class information assigned during the classification process must be close to the real class information. These two assumptions are satisfied through the minimization of the following regularization framework:

$$Q(\mathbf{F}) = \frac{1}{2} \sum_{o_i, o_j \in \mathcal{O}} w_{o_i,o_j} \Omega(\mathbf{f}_{o_i}, \mathbf{f}_{o_j}) + \mu \sum_{o_i \in \mathcal{O}^L} \Omega\prime(\mathbf{f}_{o_i}, \mathbf{y}_{o_i}), \tag{20}$$

where the first term corresponds to the first assumption, the second term corresponds to the second assumption, μ is a parameter to control how much the labeled objects must keep their class information in the transductive classification, and $\Omega(\dots)$ and $\Omega'(\dots)$ are distance functions. The differences among regularization-based algorithms are in the $\Omega(\dots)$ and $\Omega'(\dots)$ functions and how μ is set.

Eq. (20) can be minimized by closed solutions. However, this might be computationally expensive for large networks. Iterative solutions are preferable in this case, since they are less expensive and allow to set a maximum number of iterations, which speed up the transductive classification. The iterative solutions are called *label propagation*, since they propagate the labels among the network objects through the network connections in a way to minimize Eq. (20). The labels correspond to the class information vectors.

Performing transductive classification on networks requires modeling text collections in a way that documents are able to propagate their labels to other documents. In order to do so we can model text collections as document networks or bipartite heterogeneous networks (Rossi, Lopes, Faleiros, et al., 2014). In the next subsections we present regularization algorithms based on document and bipartite networks and the corresponding label propagation solutions to perform transductive classification. We also present how to generate the networks used as input for these algorithms.

2.3.1. Transductive learning on document networks

In a document network, $\mathcal{O} = \mathcal{D}$, i.e., the network objects represent the documents of a text collection. Therefore, this is a homogeneous network. Documents connected considering their similarity improve the classification accuracy (Angelova & Weikum, 2006) and do not limit the application domains such as the use of hyperlinks or citations. Thus we focus on similarity-based document networks in this article for comparison with the proposed approach.

Similarity based network generally is undirected, i.e., if there is an edge between a document d_i and a document d_j, there is also an edge between d_j and d_i. Both edges have the same weight. Usually two approaches are used to generate

similarity-based document networks (Zhu & Goldberg, 2009): (i) fully connected-network or (ii) nearest neighbor network. In this paper we consider the most representative type of each approach: (i) Exp network and (ii) Mutual k Nearest Neighbors (MkNN) network. In an Exp network, the weight of the relation between a document d_i and a document d_j (w_{d_i,d_j}) is given by a Gaussian function as $w_{d_i,d_j} = \exp(-\Omega(d_i,d_j)^2/\sigma^2)$, in which $\Omega(d_i,d_j)$ is the distance between the documents d_i and d_j, and σ controls the bandwidth of the Gaussian function. In the MkNN network, an object d_i and an object d_j are connected if d_j is one of the k nearest neighbors of d_i and d_i is one of the k nearest neighbors of d_j.

The two main regularization based algorithms to perform transductive classification on homogeneous networks are: (i) Gaussian Fields and Harmonic Functions (GFHF) and (ii) Learning with Local and Global Consistency (LLGC).

GFHF (Zhu et al., 2003) algorithm performs the transductive classification minimizing the following function:

$$Q(\mathbf{F}(\mathcal{D})) = \frac{1}{2}\sum_{d_i,d_j\in\mathcal{D}} w_{d_i,d_j}\|\mathbf{f}_{d_i}-\mathbf{f}_{d_j}\|^2 + \lim_{\mu\to\infty}\mu\sum_{d_i\in\mathcal{D}^L}\|\mathbf{f}_{d_i}-\mathbf{y}_{d_i}\|^2 \tag{21}$$

There is a restriction that $\mathbf{f}_{d_i}(\mathcal{D}^L) \equiv \mathbf{y}_{d_i}(\mathcal{D}^L)$, so the second term of Eq. (21) has a value tending to infinity. Class Mass Normalization (CMN), presented in Eq. (22), is used to classify the objects (Zhu et al., 2003) considering the final values of $\mathbf{f}_{d_i}$ vectors for $d_i \in \mathcal{D}^U$. The label propagation solution to minimize Eq. (21) is presented in Algorithm 5.

$$class(d_i) = \arg\max_{1\le l\le|\mathcal{C}|} Pr[c_l]\cdot\frac{f_{d_i,c_l}}{\sum_{d_j\in\mathcal{D}} f_{d_j,c_l}} \tag{22}$$

Algorithm 5. Gaussian Fields and Harmonic Functions.

Input : $\mathcal{D}$, $\mathbf{W}$, $\mathbf{Y}$
Output: $\mathbf{F}(\mathcal{D}^U)$

1 $\mathbf{D} = diag(\mathbf{W}\cdot\mathbf{I}_{|\mathcal{D}|})$ `/* diag(...) is the matrix diagonal operator */`
2 $\mathbf{P} \leftarrow (1/\mathbf{D}).\mathbf{W}$
3 **repeat**
4 $\quad\mathbf{F}(\mathcal{D}) \leftarrow \mathbf{P}\cdot\mathbf{F}(\mathcal{D})$
5 $\quad\mathbf{F}(\mathcal{D}^L) \leftarrow \mathbf{Y}(\mathcal{D}^L)$
6 **until** *convergence or fixed number of iterations*

LLGC (Zhou et al., 2004) decreases the influence of objects with a high degree in the definition of class information of neighboring objects. Besides, the class information of labeled documents can be changed during the classification process. The function to be minimized by LLGC is:

$$Q(\mathbf{F}(\mathcal{D})) = \frac{1}{2}\sum_{d_i,d_j\in\mathcal{D}} w_{d_i,d_j}\left\|\frac{\mathbf{f}_{d_i}}{\sqrt{\sum_{d_k\in\mathcal{D}} w_{d_i,d_k}}} - \frac{\mathbf{f}_{d_j}}{\sqrt{\sum_{d_k\in\mathcal{D}} w_{d_j,d_k}}}\right\|^2 + \mu\sum_{d_i\in\mathcal{D}^L}\|\mathbf{f}_{d_i}-\mathbf{y}_{d_i}\|^2. \tag{23}$$

The documents are classified considering the arg-max of the final values of $\mathbf{f}_{d_i}$ vectors for $d_i \in \mathcal{D}^U$. The label propagation solution to minimize Eq. (23) is presented in Algorithm 6.

Algorithm 6. Learning with Local and Global Consistency.

Input : $\mathcal{D}$,$\mathbf{W}$,$\mathbf{Y}$,
α - LLGC's parameter to attenuate differences of the class information of labeled documents in consecutive iterations
Output: $\mathbf{F}(\mathcal{D}^U)$

1 $\mathbf{D} = diag(\mathbf{W}\cdot\mathbf{I}_{|\mathcal{D}|})$
2 $\mathbf{S} = \mathbf{D}^{-1/2}\cdot\mathbf{W}\cdot\mathbf{D}^{-1/2}$
3 **repeat**
4 $\quad\mathbf{F}(\mathcal{D}) \leftarrow \alpha\cdot\mathbf{S}\cdot\mathbf{F}(\mathcal{D}) + (1-\alpha)\cdot\mathbf{Y}(\mathcal{D})$
5 **until** *convergence or fixed number of iterations*

2.3.2. Transductive learning on bipartite networks

In a bipartite heterogeneous network, $\mathcal{O} = \mathcal{D}\cup\mathcal{T}$, i.e., the network objects correspond to documents and terms of a text collection. $d_i \in \mathcal{D}$ and $t_j \in \mathcal{T}$ are linked if t_j occurs in d_i (Rossi et al., 2012). The weight of the relation between d_i and t_j (w_{d_i,t_j}) is based on the frequency of t_j in d_i. Thus, just the terms and their frequencies in the documents are necessary to generate a bipartite network. The bipartite networks used in this article are undirected. Therefore, w_{d_i,t_j} is equal to w_{t_j,d_i}.

Bipartite networks are generated faster than document networks, since they do not need to compute similarities among all documents of a text collection, and there are no parameters, which drastically change the classification performance in document networks (de Sousa et al., 2013; Rossi, Lopes, & Rezende, 2014). Moreover, bipartite networks do not need explicit links such as citations of hyperlinks, which allows them to model any type of text collection.

The general idea of regularization in bipartite networks is to minimize the differences among the class information of documents and their connected terms. In the case of label propagation, documents propagate their labels to terms and the terms propagate their labels to the documents. Therefore, terms are used as "bridges" to propagate the labels among documents. The two main regularization-based algorithms to perform transductive classification on bipartite networks are: (i) Tag-based Model (TM) and (ii) GNetMine (GM).

TM (Yin et al.,2009,2009) algorithm minimizes the differences between the (i) real class information of labeled documents ($\mathcal{D}^L$) or previous class information of unlabeled ($\mathcal{D}^U$) and the class information assigned to both of them, (ii) the real class information and the class information assigned to objects from other domains that aid the classification process ($\mathcal{A}$), like authors and conferences, and (iii) the class information among terms ($\mathcal{T}$) and objects in ($\mathcal{D}$) or ($\mathcal{A}$). The function to be minimized by TM is:

$$Q(\mathbf{F}) = \alpha \sum_{a_i \in \mathcal{A}} \|\mathbf{f}_{a_i} - \mathbf{y}_{a_i}\|^2 + \beta \sum_{d_i \in \mathcal{D}^L} \|\mathbf{f}_{d_i} - \mathbf{y}_{d_i}\|^2 + \gamma \sum_{d_i \in \mathcal{D}^U} \|\mathbf{f}_{d_i} - \mathbf{y}_{d_i}\|^2 + \sum_{o_i \in \mathcal{D} \cup \mathcal{A}} \sum_{t_j \in \mathcal{T}} w_{o_i,t_j} \|\mathbf{f}_{o_i} - \mathbf{f}_{t_j}\|^2. \quad (24)$$

The parameters α, β and γ control the importance given to the assumptions of TM. Documents are classified using class mass normalization (Eq. (22)). The label propagation solution to minimize Eq. (24) is presented in Algorithm 7.

Algorithm 7. Tag-Based Model.

Input : $\mathcal{D}$,$\mathcal{T}$,$\mathbf{W}$,$\mathbf{Y}$,
α - TM's parameter to attenuate differences of the class information of auxiliary objects in consecutive iterations,
β - TM's parameter to attenuate differences of the class information of labeled documents in consecutive iterations,
γ - TM's parameter to attenuate differences of the class information of unlabeled documents in consecutive iterations
Output: $\mathbf{F}(\mathcal{D}^U)$

repeat
  foreach $a_i \in \mathcal{A}$ **do**
    $\mathbf{f}_{a_i}(\mathcal{A}) \leftarrow \alpha \mathbf{f}_{a_i} / (\alpha + \sum_{t_j \in \mathcal{T}} w_{a_i,t_j}) + \sum_{t_j \in \mathcal{T}} w_{a_i,t_j} \mathbf{f}_j(\mathcal{T}) / (\alpha + \sum_{t_j \in \mathcal{T}} w_{a_i,t_j})$
  end
  foreach $d_i \in \mathcal{D}^L$ **do**
    $\mathbf{f}_{d_i}(\mathcal{D}^L) \leftarrow \beta \mathbf{f}_{d_i} / (\beta + \sum_{t_j \in \mathcal{T}} w_{d_i,t_j}) + \sum_{t_j \in \mathcal{T}} w_{d_i,t_j} \mathbf{f}_j(\mathcal{T}) / (\beta + \sum_{t_j \in \mathcal{T}} w_{d_i,t_j})$
  end
  foreach $d_i \in \mathcal{D}^U$ **do**
    $\mathbf{f}_{d_i}(\mathcal{D}^U) \leftarrow \gamma \mathbf{f}_{d_i} / (\gamma + \sum_{t_j \in \mathcal{T}} w_{d_i,t_j}) + \sum_{t_j \in \mathcal{T}} w_{d_i,t_j} \mathbf{f}_{t_j}(\mathcal{T}) / (\gamma + \sum_{t_j \in \mathcal{T}} w_{d_i,t_j})$
  end
  foreach $t_i \in \mathcal{T}$ **do**
    $\mathbf{f}_i(\mathcal{T}) \leftarrow \sum_{o_k \in \mathcal{A} \cup \mathcal{D}} w_{o_k,t_i} \mathbf{f}_{o_k} / \sum_{o_j \in \mathcal{A} \cup \mathcal{D}} w_{o_j,t_i}$
  end
until *convergence or fixed number of iterations*

GM (Ji, Sun, Danilevsky, Han, & Gao, 2010) is a general framework for classification in heterogeneous network based on LLGC algorithm. The difference between the algorithms is that GM considers the different types of relations among the different types of objects. For the problem of texts modeled as a bipartite network, GM minimizes the following functions:

$$Q(\mathbf{F}) = \sum_{d_i \in \mathcal{D}} \sum_{t_j \in \mathcal{T}} w_{d_i,t_j} \left\| \frac{\mathbf{f}_{d_i}}{\sqrt{\sum_{t_k \in \mathcal{T}} w_{d_i,t_k}}} - \frac{\mathbf{f}_{t_j}}{\sqrt{\sum_{d_k \in \mathcal{D}} w_{t_j,d_k}}} \right\|^2 + \sum_{d_i \in \mathcal{D}^L} \alpha \|\mathbf{f}_{d_i} - \mathbf{y}_{d_i}\|^2, \quad (25)$$

where $0 < \alpha < 1$. An unlabeled document d_i is classified according to the arg-max value of $\mathbf{f}_{d_i}$. The label propagation solution to minimize Eq. (25) is presented in Algorithm 8.

Algorithm 8. GNetMine.

Input : $\mathbf{W}$,$\mathbf{Y}$,
α - GM's parameter to attenuate differences of the class information of labeled documents in consecutive iterations
Output: $\mathbf{F}(\mathcal{D}^U)$

$\mathbf{D}(\mathcal{D}) = diag(\mathbf{W} \cdot \mathbf{I}_{|\mathcal{D}|})$
$\mathbf{D}(\mathcal{T}) = diag(\mathbf{W}^T \cdot \mathbf{I}_{|\mathcal{T}|})$
$\mathbf{S}(\mathcal{D}) = \mathbf{D}(\mathcal{D})^{-1/2} \cdot \mathbf{W} \cdot \mathbf{D}(\mathcal{T})^{-1/2}$
$\mathbf{S}(\mathcal{T}) = \mathbf{D}(\mathcal{T})^{-1/2} \cdot \mathbf{W}^T \cdot \mathbf{D}(\mathcal{D})^{-1/2}$
repeat
  $\mathbf{F}(\mathcal{T}) = \mathbf{S}(\mathcal{T})\mathbf{F}(\mathcal{D})$
  $\mathbf{F}(\mathcal{D}) = \mathbf{S}(\mathcal{D})\mathbf{F}(\mathcal{D}) + \alpha\mathbf{Y}(\mathcal{D})/(1+\alpha)$
until *convergence or fixed number of iterations*

2.4. Relevance scores induction of terms considering just labeled documents

Instead of using terms just as bridges to propagate labels in a bipartite network, the bipartite network structure can be used to induce relevance scores (class information) of terms for classes, i.e., how much a term increases (positive values) or decreases (negative values) the probability of a document belonging to a class. Rossi et al. (2012) and Rossi, Lopes, Faleiros, et al. (2014) present IMBHN (Inductive Model based on Bipartite Heterogeneous Network) algorithm, which induces the relevance scores of terms for classes through the minimization of the following function:

$$Q(\mathbf{F}(\mathcal{T})) = \frac{1}{2}\left(\sum_{c_j \in C}\sum_{d_k \in \mathcal{D}^L} \lambda\left(\sum_{t_i \in \mathcal{T}} w_{d_k,t_i} f_{t_i,c_j}\right) - y_{d_k,c_j}\right)^2, \tag{26}$$

where the function $\lambda(\dots)$ returns the value 1 for the class c_j with the highest value for $\sum_{t_i \in \mathcal{T}} w_{d_k,\,t_i} f_{t_i,\,c_j}$. The relevance scores of terms are induced to minimize the squared sum of the differences between the predicted and real classes of the training documents. The Least-Mean-Square method (Widrow & Hoff, 1960) is used to induce the relevance scores of terms. The iterative solution to minimize Eq. (26) is presented in Algorithm 9.

Algorithm 9. Inductive Model based on Bipartite Heterogeneous Networks.

Input : $\mathcal{D}^L, \mathcal{T}, C, \mathbf{W}, \mathbf{Y}$,
η - error correction rate,
Output : $\mathbf{F}(\mathcal{T})$

while *convergence or fixed number of iterations* **do**
  foreach $d_k \in \mathcal{D}^L$ **do**
    /* The output for each training document is calculated in this loop */
    $induced\_weights \leftarrow [\,]$
    foreach $c_j \in C$ **do**
      $class\_weight = 0$
      foreach $t_i \in \mathcal{T}$ **do**
        $class\_weight = class\_weight + f_{t_i,c_j} * w_{d_k,t_i}$
      end
      $out_{c_j} = class\_weight$
    end
    $out[\,] = class(induced\_weights)$ /* set the value 1 to the highest value and 0 to the others */
    /* Calculating the error */
    foreach $c_j \in C$ **do**
      $error = y_{d_k,c_j} - out_{c_j}$
      /* Weight correction for each term connected to the document */
      foreach $t_i \in \mathcal{T}$ **do**
        $current\_weight = f_{t_i,c_j}$
        $new\_weight = current\_weight + (\eta * w_{d_k,t_i} * error)$
        $f_{t_i,c_j} = new\_weight$
      end
    end
  end
end

The classification performance obtained by IMBHN is higher than that by Rossi, Lopes, Faleiros, et al. (2014). However, IMBHN just consider labeled documents to induce the relevance scores of terms for classes, which can decrease the classification performance when just few labeled documents are available. As labeling documents is time consuming, an algorithm to induce the relevance scores of terms also considering unlabeled data is interesting in scenarios with few labeled documents and plenty of unlabeled documents.

3. Proposed algorithm: transductive classification based on bipartite heterogeneous network

In this article we propose an algorithm which performs transductive classification of texts through the induction of the relevance scores of terms for classes using labeled and unlabeled documents. The proposed algorithm, named TCBHN (*Transductive Classification based on Bipartite Heterogeneous Network*) uses the bipartite network structure to induce the relevance scores of terms for classes and to define the labels of unlabeled documents. The relevance scores of terms are induced through an optimization process considering the current labels of the documents. The induced relevance scores are propagated through the bipartite network to define new labels to unlabeled documents which consequently refine the relevance scores of terms for classes. This process is repeated until convergence, i.e., until the labels of unlabeled documents remain the same. In the next subsections we present details of the algorithm, time complexity analysis, and an example using a toy and a real text collection to illustrate the TCBHN functioning.

3.1. Algorithm

The assumption of TCBHN is that the class information of documents in $\mathcal{D}^L$ and in $\mathcal{D}^U$ are useful to induce the class information of the terms, and the induced class information for the terms aids the improvement of the class information of documents in $\mathcal{D}^U$. Thus, the objective of TCBHN is to minimize the function

$$Q(\mathbf{F}) = \frac{1}{2}\left(\sum_{c_j\in\mathcal{C}}\sum_{d_k\in\mathcal{D}^L} y_{d_k,c_j} - \sum_{t_i\in\mathcal{T}}(w_{d_k,t_i} f_{t_i,c_j})\right)^2 + \frac{1}{2}\left(\sum_{c_j\in\mathcal{C}}\sum_{d_k\in\mathcal{D}^U} f_{d_k,c_j} - \sum_{t_i\in\mathcal{T}}(w_{d_k,t_i} f_{t_i,c_j})\right)^2, \tag{27}$$

i.e., the objective is to induce a matrix $\mathbf{F}(\mathcal{T})$, which contains the relevance scores of terms for classes, in a way that the frequencies of terms weighted by these relevance scores corresponds to the real class information of labeled documents and the class information assigned to unlabeled documents. The class information of labeled documents remain the same during transductive classification, i.e., we change $\mathbf{f}_{d_i}$ only for $d_i \in \mathcal{D}^U$. This is equivalent to allow changing the class information of labeled document and adding a term $\lim_{\mu\to\infty}\mu\sum_{d_i\in\mathcal{D}^L}(\mathbf{f}_{d_i}-\mathbf{y}_{d_i})^2$ in Eq. (27) as performed in GFHF (Eq. (21)).

The induction of the relevance scores of matrix $\mathbf{F}(\mathcal{T})$ is performed using the Lest-Mean-Square (LMS) method (Widrow & Hoff, 1960). LMS makes successive corrections in the relevance scores of terms in the direction of the negative gradient vector, which will lead to the minimum mean squared error. The relevance score updating equation using LMS is presented in Eq. (28). The direction of the gradient can be estimated by the derivative of $Q(\mathbf{F})$, as presented in Eq. (29).

$$\mathbf{f}^{(n+1)} = \mathbf{f}^{(n)} + \eta[-\nabla(Q(\mathbf{F}))] \tag{28}$$

$$\nabla(Q(\mathbf{F})) = \frac{\partial Q(\mathbf{F})}{\partial \mathbf{F}} = \sum_{c_j\in\mathcal{C}}\sum_{d_k\in\mathcal{D}^L} y_{d_k,c_j}\left(\sum_{t_i\in\mathcal{T}} w_{d_k,t_i} f_{t_i,c_j}\right)\sum_{c_j\in\mathcal{C}}\sum_{d_k\in\mathcal{D}^L}\sum_{t_i\in\mathcal{T}} w_{d_k,t_i} + \sum_{c_j\in\mathcal{C}}\sum_{d_k\in\mathcal{D}^U} f_{d_k,c_j}\left(\sum_{t_i\in\mathcal{T}} w_{d_k,t_i} f_{t_i,c_j}\right)\sum_{c_j\in\mathcal{C}}\sum_{d_k\in\mathcal{D}^U}\sum_{t_i\in\mathcal{T}} w_{d_k,t_i} \tag{29}$$

We call the difference $y_{d_k,c_j} - (\sum_{t_i\in\mathcal{T}} w_{d_k,t_i} f_{t_i,c_j})$ or $f_{d_k,c_j} - (\sum_{t_i\in\mathcal{T}} w_{d_k,t_i} f_{t_i,c_j})$ as $error_{d_k,c_j}$. Considering Eqs. (28) and (29), the score $f_{i,j}^{(s+1)}(\mathcal{T})$ of a term t_i for the class c_j in time $(s+1)$ is given by the following equation:

$$f_{t_i,c_j}(\mathcal{T})^{(s+1)} = f_{t_i,c_j}(\mathcal{T})^{(s)} + \eta\left(\sum_{d_k\in\mathcal{D}^L} w_{d_k,t_i} error^{(n)}_{d_k,c_j} + \sum_{d_k\in\mathcal{D}^U} w_{d_k,t_i} error^{(n)}_{d_k,c_j}\right), \tag{30}$$

where η is the error correction rate, i.e., the rate in which the error will be considered in the relevance score updating.

If we directly apply Eq. (30) iteratively, the relevance scores of terms are obtained considering just the labeled documents since there is no class information assigned to unlabeled documents at the beginning of the classification process. Thus, we propose a three-step approach to induce the relevance scores of terms considering labeled and unlabeled documents: (i) inducing the relevance scores of terms considering just labeled documents, (ii) propagating the relevance scores of terms to unlabeled documents, and (iii) refining the relevance scores of terms considering unlabeled documents. Hence, Eq. (30) is divided into two equations: one considering labeled documents (Eq. (31)), which is applied in the first step, and one considering unlabeled documents (Eq. (32)), which is applied in the third step.

$$f_{t_i,c_j}(\mathcal{T})^{(s+1)} = f_{t_i,c_j}(\mathcal{T})^{(s)} + \eta\left(\sum_{d_k\in\mathcal{D}^L} w_{d_k,t_i} error^{(n)}_{d_k,c_j}\right) \tag{31}$$

$$f_{t_i,c_j}(\mathcal{T})^{(s+1)} = f_{t_i,c_j}(\mathcal{T})^{(s)} + \eta\left(\sum_{d_k\in\mathcal{D}^U} w_{d_k,t_i} error^{(n)}_{d_k,c_j}\right) \tag{32}$$

Eqs. (31) and (32) are applied until a stopping criterion is reached. We adopted as stopping criteria the maximum number of iterations and the minimum mean squared error ($\in$), i.e., until the mean squared error considering labeled (for Eq. (31)) or unlabeled (for Eq. (32)) is lower than a user's threshold. We call the iterations carried out to optimize the relevance scores of terms using Eqs. (31) or (32) as *local iterations*.

The relevance scores of terms are propagated to unlabeled documents using the weighted linear function presented in Eq. (33) (second step).

$$f_{d_i,c_j}(\mathcal{D}^U) = \sum_{t_k \in \mathcal{T}} w_{d_i,t_k} f_{t_k,c_j} \tag{33}$$

Since the labeled documents have the sum of the class information equal to 1, we standardize the class information propagated to unlabeled documents using Eq. (34) to also make the sum equal to 1. In case of negative value in a class information vector of a document, we sum the module of the most negative value in all class information values before applying Eq. (34).

$$f_{d_i,c_j}(\mathcal{D}^U) = \frac{f_{d_i,c_j}(\mathcal{D}^U)}{\sum_{c_k \in \mathcal{C}} f_{d_i,c_k}(\mathcal{D}^U)} \tag{34}$$

We apply the 3 steps iteratively until a maximum number of iterations is reached or until the class information of labeled documents remains the same in two successive iterations. An iteration containing the three steps is called *global iteration*. The pseudocode of TCBHN algorithm is presented in Algorithm 10. Lines 2–12 induces the relevance scores of terms using labeled documents, lines 13–15 assign class information to unlabeled documents, and lines 16–26 induces the relevance scores of terms using the class information assigned to unlabeled documents. The function $ClassifyInstance(d_i, F(\mathcal{T}))$ applies the Eq. (33) to define the class information of documents using the current relevance scores of terms and Eq. (34) to make the sum of values of a class information vector equal to 1.

Algorithm 10. Transductive Categorization based on Bipartite Heterogeneous Network.

Input : $\mathcal{D}^L, \mathcal{D}^U, \mathcal{T}, \mathcal{C}, \mathbf{W}, \mathbf{Y}$,
η - error correction rate
Output : $\mathbf{F}(\mathcal{D}^U)$

1 **repeat**
  /* First Step - Induction of the relevance scores of terms considering labeled documents */
2 **repeat**
3 **foreach** $d_i \in \mathcal{D}^L$ **do**
4 $EstimatedClass[] = ClassifyInstance(d_i, F(\mathcal{T}))$ /* Defining class information of documents considering the current class information of terms (Equations 33 and 34) */
5
6 **foreach** $c_j \in C$ **do**
7 $error = yd_i, c_j - EstimatedClass[c_j]$
8 **foreach** $t_k \in T$ **do**
9 $f_{t_k,c_j}(\mathcal{T}) = f_{t_k,c_j}(\mathcal{T}) + \eta * w_{d_i,t_k} * error$
10 **end**
11 **end**
12 **end**
13 **until** *convergence of fixed number of iterations*
  /* Second Step - Propagating the relevance scores of terms considering unlabeled documents */
14 **foreach** $d_i \in \mathcal{D}^U$ **do**
15 $f_{d_i}(\mathcal{D}^U) = ClassifyInstance(d_i, F(\mathcal{T}))$
16 **end**
  /* Third Step - Refining the relevance scores of terms considering unlabeled documents */
17 **repeat**
18 **foreach** $d_i \in \mathcal{D}^u$ **do**
19 $EstimatedClass[] = ClassifyInstance(d_i, F(\mathcal{T}))$
20 **foreach** $c_j \in C$ **do**
21 $error = f_{d_i,c_j} - EstimatedClass[c_j]$
22 **foreach** $t_k \in T$ **do**
23 $f_{t_k,c_j}(\mathcal{T}) = f_{t_k,c_j}(\mathcal{T}) + \eta * w_{d_i,t_k} * error$
24 **end**
25 **end**
26 **end**
27 **until** *convergence or fixed number of iterations*
28 **until** *convergence or fixed number of iterations*

The relation weights for terms in a document are normalized to improve the optimization (Valin & Collings, 2007). Thus, the weight of a relation between a term t_j and a document d_i is

$$w_{d_i,t_j} = \frac{w_{d_i,t_j}}{\sum_{t_k \in \mathcal{T}} w_{d_i,t_k}}. \quad (35)$$

3.2. Time complexity analysis

The cost to induce the relevance scores of terms through optimization is $O(n_{Local} * |\mathcal{D}| * \overline{|\mathcal{T}|} * |\mathcal{C}|)$, since the relevance scores of terms from each document are updated up to n_{Local} times for each class. The cost to propagate the labels is $O(|\mathcal{D}| * \overline{|\mathcal{T}|} * |\mathcal{C}|)$, since each document receives the relevance scores from their terms for all classes. Optimization and label propagation are repeated up to n_{Global} times until convergence. Thus, the total cost for transductive classification using TCBHN is $O(n_{Global} * ((n_{Local} * |\mathcal{D}| * \overline{|\mathcal{T}|} * |\mathcal{C}|) + (|\mathcal{D}| * \overline{|T|} * |\mathcal{C}|)) \equiv O(n_{Global} * (n_{Local} * |\mathcal{D}| * \overline{|\mathcal{T}|} * |\mathcal{C}|))$.

3.3. Impact of the unlabeled documents in the relevance scores of terms

To illustrate the functioning of the proposed algorithm and show the impact of the unlabeled documents in the relevance scores of terms for classes, we ran TCBHN in a toy and in a real text collection. For the toy collection we generated a network with 8 documents and 9 terms, as presented in Fig. 1. In this figure, `Documents` 1, 2, 3, and 4 belong to `Class` 1 and `documents` 5, 6, 7, and 8 belong to `Class` 2. `Terms` 1, 2 and 3 belong exclusively to the documents of `Class` 1, terms 7, 8, and 9 belong exclusively to the documents of `Class` 2, and `Terms` 4, 5 and 6 belong to the document of both classes. We set term frequencies equal 1 and $\eta = 0.5$ to facilitate the understanding.

We started the transductive classification with just one labeled document for each class, `Doc` 1 for `Class` 1 and `Doc` 8 for `Class` 2, as presented in Fig. 1(a) (first step). Firstly, the relevance scores of terms for the classes were induced considering just the labeled documents, as presented in Fig. 1(b) (second step). These relevance scores were propagated to unlabeled documents to set their class information and normalized, as presented in Fig. 1(c) (third step). The new class information assigned to documents were used to optimize the relevance scores of terms for the classes. The illustrations presented in Fig. 1(a)–(c) are the first global iteration.

The new relevance scores of terms at the second global iteration and the class information of the documents after propagating these relevance scores are presented in Fig. 1(d). We notice that terms with no induced relevance score in the first global iteration have their relevance scores induced in the second iteration. Besides, the relevance scores of terms are changed for those terms with previous induced relevance scores.

Even if all the documents have class information, the relevance scores of terms for classes may change and consequently the class information of unlabeled documents, as presented in Fig. 1(e). At the end, labels are assigned to unlabeled documents according to the arg-max of their class information vectors. In this illustrative example we correctly classify all the documents. We notice that if we just use labeled documents (Fig. 1(b)), `Doc` 4 and `Doc` 5 will not receive any class information and will not be classified.

We also present the impact of the unlabeled documents in the relevance scores of terms for classes in a real collection. In order to do so, we used CSTR (Computer Science Technical Reports) collection (Rossi, Marcacini, & Rezende, 2013) which contains documents from 4 areas: Artificial Intelligence (AI)/Natural Language Processing (NLP), Robotics/Vision, Systems, and Theory. We selected 10 documents from each class randomly to be the labeled objects of the network. The maximum number of global iterations was 10, the maximum number of local iterations was 100, and we set $\eta = 0.05$. We used stemmed single words as terms. The words were stemmed using Porter's algorithm (Porter, 1980). We also run the algorithm IMBHN, which induces the relevance scores of terms just considering labeled documents, i.e., using just the first term of Eq. (27), to obtain the relevance scores of terms.

Table 1 presents the ranking of terms for each class without and with unlabeled documents respectively. We notice that important terms for AI/NLP such as "*learn*", "*label*", and "*language*" are not in the top ranked terms when just labeled documents are used. Other important terms as "*knowledge*" and "*class*" are in both rankings. However, terms as "*class*" and "*knowledge*" presents a lower relevance score for IA/NLP using just labeled documents than using labeled and unlabeled documents. Besides, we can notice that important terms for IA/NLP and not important for other areas of the CSTR collection, such as "*learn*", "*word*", "*label*", "*knowledge*", "*gener*", and "*discours*" present negative relevance scores for other classes, i.e., these terms decrease the class information of the documents belonging to these classes when labeled and unlabeled documents are considered. The differences in the relevance scores of terms provided by the use of unlabeled documents also impact the classification performance, as we present in Section 4.

4. Experimental evaluation

In the experimental evaluation we compared TCBHN with algorithms presented in Section 2, which consider text collections represented in a vector space model, document and bipartite networks. We also considered inductive supervised learning algorithms to demonstrate if and how much unlabeled documents are useful to improve classification performance. Moreover, our goal is to demonstrate that (i) our proposal surpass the classification performance obtained by

state-of-the-art algorithms for transductive classification of texts and (ii) our proposal is scalable for large text collections. In next sections we present the text collections used in the experimental evaluation, experiment configuration, evaluation criteria, results and discussion. Due to reasons concerning reproducibily, all source codes and text collections used in our experimental evaluation are freely available.[1]

4.1. Text collections

We used 30 textual document collections from different domains: e-mails (EM), medical documents (MD), news articles (NA), scientific documents (SD), sentiment analysis (SA), TREC (Text Retrieval Conference) documents (TD), and web pages (WP). Below we will give a brief description of the collections:

E-mails

SpamAssassin: collection for testing spam filtering systems. This collection is composed by spams and non-spam (ham) e-mails (Apache, 2013).
Trec7-3000: composed by spam and non-spam e-mails (Cormack & Lynam, 2007). We selected 3000 spam and 3000 non-spam e-mails.

Medical documents

Oh0, Oh5, Oh10, Oh15, Ohscal: these collections are subsets of the OHSUMED collection (Forman, 2006; Hersh, Buckley, Leone, & Hickam, 1994).

News articles

Foreign Broadcast Information Service (FBIS): this collection is composed by newspaper articles from around the world (Forman, 2006). FBIS is a subcollection of the TREC collection (TREC, 2013).
Hitech: composed by news about computers, electronics, health, medical, research, and technology from the San Jose Mercury newspaper (Karypis, 2013). This collection is part of the TREC collection (TREC (2013)).
La1, La2: these collections are composed by Los Angeles Times news articles extracted from TREC-5 (Forman, 2006; TREC, 2013).
Reuters Corpus Volume 1 – v2 – Top Four Categories (RCV1 Top-4): newswire stories available by Reuters[2] (Lewis, 2014). We consider just the top 4 categories of the category hierarchy (Economics, Government/Social, Corporate/Industrial, Equity Markets) and documents from a single category.
Re0, Re8: Re0 (Forman, 2006) and Re8 (Pang, 2014) are composed by articles from Reuters-21578 collection (Lewis, 2013).
Reviews: derived from the San Jose Mercury newspaper.[3] The news is about food, movies, music, radio, and restaurants (Karypis, 2013).

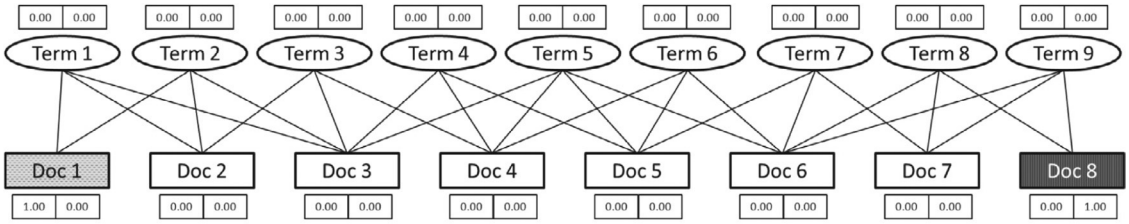
Scientific documents

Classic4: composed by the collections CACM (titles and abstracts from the journal Communications of the ACM), CISI (information retrieval papers), CRANFIELD (aeronautical system papers), and MEDLINE (medical journals) (D.M. Research, 2013).

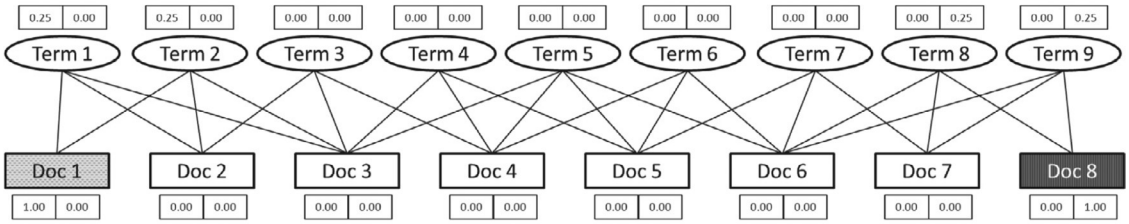
Sentiment analysis

Irish Sentiment: composed by articles labeled by volunteers as positive, negative or irrelevant. The articles were extracted from Irish online sources: RTE News, The Irish Times, and the Irish Independent (Group, 2012).
Multi Domaim Sentiment: contains product reviews taken from Amazon[4] from different product types (Blitzer, Dredze, & Pereira, 2013).
Review Polarity: composed by 1000 positive and 1000 negative reviews about movies (Pang & Lee, 2013).

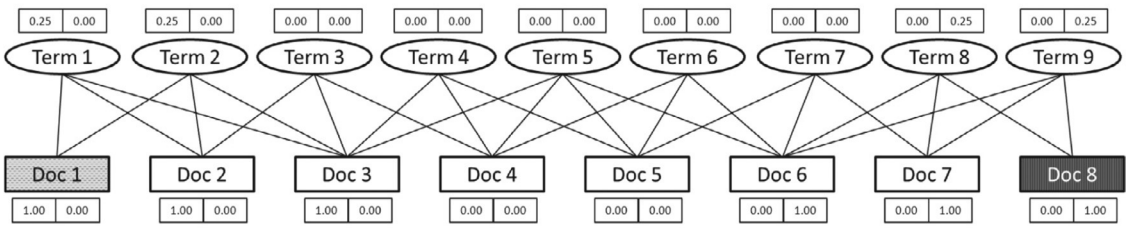
[1] http://sites.labic.icmc.usp.br/ragero/jipm_2015/.
[2] http://www.reuters.com/.
[3] http://www.mercurynews.com/.
[4] http://www.amazon.com/.



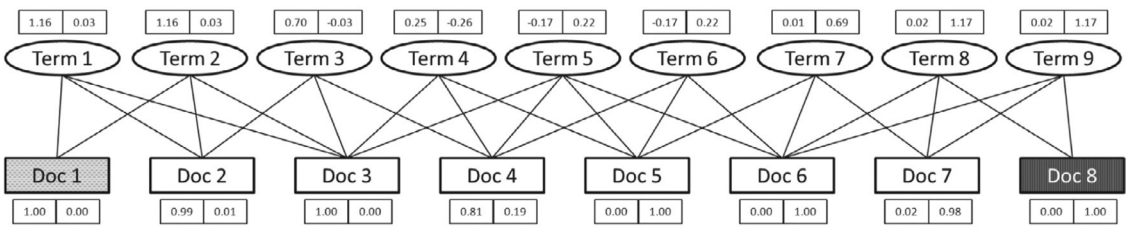
(a) Intial class information of documents and relevance scores of terms at the beginning of the classification process.



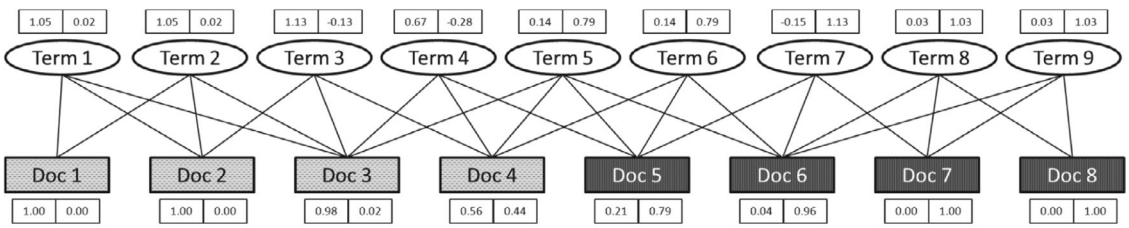
(b) Relevance scores of terms for classes considering just labeled documents.



(c) Class information of documents after label propagation.



(d) Redefined relevance scores of documents for classes and class information of documents.



(e) Classification of documents considering the arg-max of their class information.

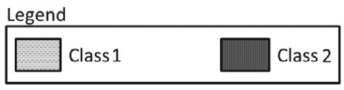


Fig. 1. Example of TCBHN functioning in the toy example.

Web pages

Dmoz-Health-500: composed by web pages from the subcategories of Health extracted from *DMOZ – Open Directory Project* (Netscape, 2013). We consider 500 document from each subcategory.

Table 1
Relevance scores of terms for classes using just labeled documents (IMBHN) and using labeled and unlabeled documents (TCBHN).

Terms	AI/NLP	Robotics	System	Theory
Labeled documents (IMBHN)				
extract	0.0065	−0.0065	0.0000	0.0000
knowledg	0.0065	−0.0065	0.0000	0.0000
acquisit	0.0022	−0.0022	0.0000	0.0000
addit	0.0022	−0.0035	0.0000	0.0014
analysi	0.0022	−0.0022	0.0000	0.0000
area	0.0022	−0.0022	0.0000	0.0000
art	0.0022	0.0003	0.0000	0.0000
class	0.0022	−0.0022	0.0000	0.0000
compar	0.0022	−0.0022	0.0000	0.0000
condit	0.0022	−0.0022	0.0000	0.0000
current	0.0022	−0.0022	0.0000	0.0000
enabl	0.0022	−0.0022	0.0000	0.0000
entiti	0.0022	−0.0022	0.0000	0.0000
event	0.0022	−0.0022	0.0000	0.0000
exampl	0.0022	−0.0022	0.0000	0.0000
exist	0.0022	0.0003	0.0000	0.0000
explor	0.0022	−0.0022	0.0000	0.0000
fact	0.0022	−0.0022	0.0000	0.0000
field	0.0022	0.0003	0.0000	0.0000
focus	0.0022	−0.0022	0.0000	0.0000
follow	0.0022	−0.0022	0.0000	0.0000
gener	0.0022	−0.0022	0.0000	0.0000
give	0.0022	−0.0022	0.0000	0.0000
high	0.0022	−0.0022	0.0000	0.0000
idea	0.0022	−0.0022	0.0000	0.0000
individu	0.0022	−0.0022	0.0000	0.0000
inform	0.0022	−0.0022	0.0000	0.0000
linguist	0.0022	−0.0022	0.0000	0.0000
make	0.0022	−0.0022	0.0000	0.0000
norm	0.0022	−0.0022	0.0000	0.0000
order	0.0022	−0.0022	0.0037	−0.0037
pertain	0.0022	−0.0022	0.0000	0.0000
prefer	0.0022	−0.0026	0.0012	−0.0008
research	0.0022	−0.0022	0.0000	0.0000
respect	0.0022	−0.0022	0.0012	−0.0012
simpl	0.0022	−0.0022	0.0000	0.0000
specif	0.0022	−0.0022	0.0000	0.0000
state	0.0022	0.0003	0.0000	0.0000
techniqu	0.0022	−0.0001	0.0000	0.0005
throughput	0.0022	−0.0022	0.0000	0.0000
volum	0.0022	−0.0022	0.0000	0.0000
work	0.0022	−0.0022	0.0000	0.0000
abandon	0.0000	0.0000	0.0000	0.0000
abil	0.0000	0.0000	0.0000	0.0000
abort	0.0000	0.0000	0.0000	0.0000
absenc	0.0000	0.0000	0.0000	0.0000
absolut	0.0000	0.0000	0.0000	0.0000
abstract	0.0000	0.0000	0.0000	0.0000
acceler	0.0000	0.0000	0.0000	0.0000
accept	0.0000	0.0000	0.0000	0.0000
Labeled + unlabeled documents (TCBHN)				
learn	2.8961	−0.2465	−0.4497	−0.5075
system	2.8264	0.2930	0.8924	0.3062
reason	2.2541	−0.3650	−0.3968	−0.3164
word	2.2303	−0.1227	−0.1046	−0.1820
label	2.2020	−0.9021	−0.4309	−0.4910
languag	2.0938	−0.0441	0.0103	0.0151
task	1.9653	0.3391	0.1562	0.0782
approach	1.8651	0.4169	0.0604	−0.2662
knowledg	1.8584	−0.0647	−0.1534	−0.3536
goal	1.8519	0.0248	−0.1752	−0.3624
gener	1.7993	−0.6234	−0.3756	1.3022
discours	1.6799	−0.2984	−0.0774	−0.3274
mean	1.6573	−0.3072	−0.3520	−0.3141
includ	1.6386	−0.8190	−0.1023	0.7117
class	1.5588	−0.0769	−0.0597	0.3397
natur	1.5527	−0.1602	−0.2187	0.5213

(*continued on next page*)

Table 1 (*continued*)

Terms	AI/NLP	Robotics	System	Theory
speech	1.5240	0.1083	0.0615	−0.0823
featur	1.4760	0.4561	−0.6301	−0.0584
work	1.4612	0.3024	−0.0503	−0.1407
network	1.3922	−0.0962	−0.0399	−0.1426
plan	1.3797	0.6019	0.0395	0.0899
extract	1.2469	−0.4217	−0.1095	−0.2492
dialogu	1.2307	−0.1604	−0.0449	−0.2403
utter	1.1998	−0.2322	−0.0837	−0.2575
resolut	1.1665	−0.3889	−0.1815	−0.2486
combin	1.1544	−0.2087	0.0432	−0.0751
paper	1.1501	0.6362	0.5113	0.0915
modul	1.1335	−0.3997	−0.2140	−0.2892
represent	1.1070	0.5214	−0.0314	−0.2501
organ	1.0750	−0.2204	−0.1899	−0.2716
inform	1.0534	1.3847	−0.4025	−0.1084
structur	1.0294	0.4220	0.3224	−0.2304
understand	1.0230	−0.1736	−0.0709	−0.2406
speaker	1.0027	0.0359	0.0433	−0.0755
spoken	1.0020	−0.1459	−0.0415	−0.1680
output	0.9955	−0.0297	0.0275	−0.3262
process	0.9942	1.0418	−0.0144	−0.5005
supervis	0.9876	−0.4469	−0.2053	−0.3192
user	0.9858	0.0475	0.2628	−0.3132
convers	0.9676	−0.1906	0.0858	−0.1869
domain	0.9321	0.2116	0.1029	0.0787
requir	0.9230	0.3648	0.0985	−0.1814
relat	0.9125	0.3661	−0.1467	0.2192
specif	0.8733	−0.0887	0.1438	−0.0257
report	0.8716	0.7876	−0.1076	−0.1491
train	0.8546	0.5378	−0.1125	−0.0517
appli	0.8513	−0.0968	−0.0611	0.0256
individu	0.8020	−0.1921	−0.0656	−0.1773
transcript	0.7851	−0.1900	−0.0351	−0.1285
present	0.7822	1.3742	0.2544	0.2427

Dmoz-Science-500: composed by web pages from the subcategories of Science extracted from DMOZ (Netscape, 2013). We consider 500 documents from each subcategory.
Dmoz-Sports-500: composed by web pages from the subcategories of Sports extracted from DMOZ (Netscape, 2013). We consider 500 documents from each subcategory.
Industry Sector: composed by web pages of companies from various economic sectors (Nigam, 2012).
Syskill & **Webert**: composed by web pages about bands, sheep, goats, and biomedical (Pazzani, 2013).
WebACE Project (WAP): composed by web pages from de WebACE Project (Han et al., 1998; Forman, 2006). The web pages belong to the subject hierarchy of Yahoo.[5]
World Wide Knowledge Base (WebKB): composed by web pages collected from computer science departments of various universities in January 1997 by the CMU Text Learning Group[6] (Group, 2013).

TREC documents

Tr11, Tr31, Tr41, Tr45: these collections (Forman, 2006) are derived from Trec-5, Trec-6 and Trec-7 collections (TREC, 2013). The documents are labeled with queries which they were considered relevant.

The collections have different characteristics. The number of documents ($|\mathcal{D}|$) ranges from 334 to 685,071, the number of terms ($|\mathcal{T}|$) from 2001 to 153,458, the average number of terms per document ($|\overline{\mathcal{T}}|$) from 11.52 to 281.66, the number of classes ($|\mathcal{C}|$) from 2 to 20, the standard deviation considering the class percentages in each collection ($\sigma(\mathcal{C})$) from 0 to 34.45, and the percentage of the majority class ($\max(\mathcal{C})$) from 5.31 to 74.36. Table 2 presents the characteristics of the collections.

For the collections La1s, La2s, Oh0, Oh10, Oh15, Oh5, Re0, Tr11, Tr31, Tr41, Tr45, and WAP (Forman, 2006) no preprocessing was performed since these collections were already preprocessed. For the other collections, single words were considered as terms, stopwords were removed, terms were stemmed using the Porter's algorithm (Porter, 1980), HTML tags

[5] http://dir.yahoo.com/.
[6] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-4/text-learning/www/index.html.

Table 2
Characteristics of the textual document collections.

Collection	$\lvert\mathcal{D}\rvert$	$\lvert\mathcal{T}\rvert$	$\overline{\lvert\mathcal{T}\rvert}$	$\lvert\mathcal{C}\rvert$	$\sigma(\mathcal{C})$	$\max(\mathcal{C})$
Classic4 (SD)	7095	7749	35.28	4	1.94	45.16
Dmoz-Health-500 (WP)	6500	4217	12.40	13	0.00	7.69
Dmoz-Science-500 (WP)	6000	4821	11.52	12	0.00	9.63
Dmoz-Sports-500 (WP)	13,500	5682	11.87	27	0.00	3.70
FBIS (NA)	2463	2001	159.24	17	5.66	26.54
Hitech (NA)	2301	12,942	141.93	6	8.25	26.21
Industry-Sector (PW)	8817	21,490	88.49	12	7.37	11.24
Irish_Sentiment (SA)	1660	8659	112.65	3	6.83	39.46
La1s (NA)	3204	13,196	144.64	6	8.22	29.43
La2s (NA)	3075	12,433	144.83	6	8.59	29.43
Multi_Domaim_Sentiment (SA)	8000	13,360	42.36	2	0.00	50.00
Oh0 (MD)	1003	3183	52.50	10	5.33	19.34
Oh10 (MD)	1050	3239	55.64	10	4.25	15.71
Oh15 (MD)	913	3101	59.30	10	4.27	17.20
Oh5 (MD)	918	3013	54.43	10	3.72	16.23
Ohscal (MD)	11,162	11,466	60.39	10	2.66	14.52
RCV1 Top-4 (NA)	685,071	153,458	74.61	4	14.46	43.63
Re0 (NA)	1504	2887	51.73	13	11.56	40.43
Re8 (NA)	7674	8901	35.31	8	18.24	51.12
Review Polarity (SA)	2000	15,698	205.06	2	0.00	50.00
Reviews (NA)	4069	22,927	183.10	5	12.80	34.11
SpamAssassin (EM)	9348	97,851	108.02	2	34.45	74.36
Syskill & Webert (WP)	334	4340	93.16	4	10.75	41.02
Tr11 (TD)	414	6430	281.66	9	9.80	31.88
Tr31 (TD)	927	10,129	268.50	7	13.37	37.97
Tr41 (TD)	878	7455	195.33	10	9.13	27.68
Tr45 (TD)	690	8262	280.58	10	6.69	23.19
Trec7-3000 (EM)	6000	100,464	244.08	2	0.00	50.00
WAP (WP)	1560	8461	141.33	20	5.20	21.86
WebKB (WP)	8282	22,892	89.78	7	15.19	45.45

were removed, and only terms with document frequency ≥ 2 were considered. We used term frequency to weight terms in documents. More details about the collections are presented in Rossi et al. (2013).

4.2. Experiment configuration and evaluation criteria

We compared TCBHN with traditional and state-of-the-art transductive algorithms based on vector space model, and based on document and bipartite networks. We also ran inductive supervised learning algorithms to verify if and how much unlabeled documents improve classification performance and the cost to move from inductive supervised learning to transductive learning. All algorithms used for comparison were presented in Section 2. The parameter values used in the experimental evaluation were based on the values found in the proposal of the algorithms or empirically.

The transductive learning algorithms based on vector space model, considerations, and parameters are:

- **Self-Training (MNB-Se):** we considered Multinomial Nave Bayes (MNB) as inductive learning algorithm for Self-Training since it presents the best trade-off between classification performance and time (Rossi et al., 2012; Rossi, Lopes, Faleiros, et al., 2014). We used $X = \{5, 10, 15, 20\}$.
- **Co-Training (MNB-Co):** we considered MNB as inductive supervised learning algorithm for Co-Training. We used $X = \{5, 10, 15, 20\}$. We randomly split the feature set into two disjunct sets to allow running Co-Training (Xu et al., 2013; Li et al., 2009; Bickel & Scheffer, 2004). Other ways to generate views or other inductive supervised learning algorithms would make impracticable the execution of Co-Training. The classification performance of Co-Training is an average of the classification performances obtained by 10 different random splits in feature set, since different splits can generate different classification performances.
- **Expectation Maximization (EM)**: we considered the EM instantiation for text classification presented in Nigam et al. (2000). We used $\lambda = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and 1, 2, 5, 10 components for each class.
- **Transductive Support Vector Machines (TSVM)**: we considered the iterative solution for text classification proposed by Joachims (1999). We used $C = 1.0$ to induce a maximal margin hyperplane considering labeled documents since this is one or the best parameters for text classification (Rossi et al., 2012). We vary C' by a factor of ten from 10^{-5} to 10^{1}. We run TSVM with and without the function proposed in Joachims (1999) to maintain the same class proportion of labeled documents in the classification of unlabeled documents.

For the transductive algorithms based on networks we used the iterative solutions presented on Section 2 (algorithms used for comparison) and Section 3 (our proposal). We set 1000 as the maximum number of iterations. For algorithms based

on document networks we generated Mutual k-Nearest Neighbor (MkNN) and Exp networks. To build MkNN networks we used $k = \{7, 17, 37, 57\}$, and to build Exp networks we used $\sigma = \{0.05, 0.2, 0.35, 0.5\}$.

The algorithms based on document networks, considerations, and parameters are:

- **Gaussian Fields and Harmonic Functions (GFHF)**: There are no parameters for GFHF.
- **Learning with Local and Global Consistency (LLGC)**: We used $\alpha = \{0.1, 0.3, 0.5, 0.7, 0.9\}$.

The algorithms based on bipartite networks, considerations, and parameters are:

- **Tag-based Model)(TM)**: we used $\alpha = 0$, since there are no objects from different domains, $\beta = \{0.1, 1, 10, 100, 1000\}$, and $\gamma = \{0.1, 1, 10, 100, 1000\}$.
- **GNetMine**: we used $\alpha = \{0.1, 0.3, 0.5, 0.7, 0.9\}$.
- **Transductive Categorization based on Bipartite Heterogeneous Networks (TCBHN)**: we used $\eta = \{0.01, 0.05, 0.1, 0.5\}$, $\in = 0.01$, 10 as maximum number of global iterations and 100 as maximum number of local iterations, which gives a total of 1000 iterations.

We also run inductive supervised learning algorithms to analyse the trade-off between computational cost and classification performance. This also allows us to analyse if the use of unlabeled documents actually improve classification performance. The algorithms, parameters, and considerations of the inductive supervised learning algorithms are:

- **Multinomial Nave Bayes (MNB)**: we considered MNB since it is the learning algorithm used in Self-Training, Co-Training and Expectation Maximization. This allowed us to measure the difference in classification performance and time to move from inductive supervised learning to transductive learning for algorithms based on vector space model. There are no parameters for MNB.
- **Inductive Model based on Bipartite Heterogeneous Network (IMBHN$^{\mathbf{R}}$)**: we induced the relevance scores of terms using just labeled documents, i.e., the first term of Eq. (27). This corresponds to IMBHN's equation (Rossi et al., 2012; Rossi, Lopes, Faleiros, et al., 2014) without the class function. We call this algorithm by IMBHN$^{\mathrm{R}}$, since it performs regression (R) on the information class of labeled documents to induce the relevance scores of terms. The use of IMBHN$^{\mathrm{R}}$ allowed us to measure the differences in classification performance and time when considering unlabeled data to define the relevance scores of terms. We used $\eta = \{0.01, 0.05, 0.1, 0.5\}$, $\in = 0.01$, and 1000 as maximum number of iterations. These were the same parameters of TCBHN.

Only for RCV1 Top-4 collection we did not run LLGC and GFGF algorithms, since computing, storing, and building document networks for such amount of documents is impracticable. We also did not run Self-Training and Co-Training since the parameters used for these algorithms would make to reinduce classification models about 34,000 times for each configuration, which is also impracticable. TSVM also demonstrated to be impracticable for this collection.

We used the F^1 measure to compare the classification results. F^1 is the harmonic mean of precision and recall measures, in which both measures have the same weight, i.e.

$$F^1 = 2 * \frac{Precision * Recall}{Precision + Recall}. \tag{36}$$

Precision and recall were computed for each class in multiclass evaluation. The precision and recall of a class c_i are:

$$Precision_{c_i} = \frac{TP_{c_i}}{TP_{c_i} + FP_{c_i}}, \tag{37}$$

$$Recall_{c_i} = \frac{TP_{c_i}}{TP_{c_i} + FN_{c_i}}, \tag{38}$$

where TP (True Positive) means the number of test documents correctly assigned to class c_i, FP (False Positive) means the number of test documents from class c_j ($c_j \neq c_i$) but assigned to class c_i, and FN (False Negative) is the number of test documents from class c_i but assigned to class c_j ($c_j \neq c_i$). Precision returns the percentage of documents correctly classified as c_i considering all documents classified as c_i, and recall returns the percentage of documents correctly classified as c_i considering all documents which actually belong to class c_i.

Two strategies to summarize the results of precision and recall computed for each class of a text collection are: (i) **micro-averaging** and **macro-averaging** (Sokolova & Lapalme, 2009; Manning et al., 2008; Sebastiani, 2002). The micro-averaging strategy performs a sum of the terms of the evaluation measures. Therefore, the precision and recall using the micro-averaging strategy are:

$$Precision^{Micro} = \frac{\sum_{c_i \in \mathcal{C}} TP_{c_i}}{\sum_{c_i \in \mathcal{C}} (TP_{c_i} + FP_{c_i})}, \tag{39}$$

$$Recall^{Micro} = \frac{\sum_{c_i \in \mathcal{C}} TP_{c_i}}{\sum_{c_i \in \mathcal{C}} (TP_{c_i} + FN_{c_i})}. \quad (40)$$

The macro-averaging strategy performs and average over the evaluations measures for each class. Therefore, the precision and recall using macro-averaging strategy are:

$$Precision^{Macro} = \frac{\sum_{c_i \in \mathcal{C}} Precision_{c_i}}{|\mathcal{C}|}, \quad (41)$$

$$Recall^{Macro} = \frac{\sum_{c_i \in \mathcal{C}} Recall_{c_i}}{|\mathcal{C}|}. \quad (42)$$

Micro-averaging scores are dominated by the number of *TP*. Therefore, large classes dominate small classes in micro-averaging scores. On the other hand, macro-averaging gives equal weight to each class. In this case, the number of *TP* in small classes are emphasized in macro-averaging scores. These two strategies give different scores and are complementary to each other. We denote F^1 computed through micro-averaging of precision and recall by *Micro*-F^1, and through macro-averaging by *Macro*-F^1.

Micro-F^1 and *Macro*-F^1 scores were obtained considering the average from 10 runs. In each run we randomly selected N documents from each class as labeled documents. We carried out experiments using $N = \{1, 10, 20, 30, 40, 50\}$. We started with the minimum number of labeled document per class and varied by a factor of ten from 10 to 50. This variation in the number of labeled documents allowed us to better demonstrate the behavior of the algorithms for different number of labeled documents, a trade-off between the number of labeled documents and classification performance, and the differences among inductive supervised learning algorithms and transductive learning algorithms as we increase the number of labeled documents. The remaining $|\mathcal{D}| - (N * |\mathcal{C}|)$ documents were used to evaluate the classification.

4.3. Results

In this section we present the best *Micro*-F^1 and *Macro*-F^1 values obtained in the experimental evaluation to facilitate the analysis and to make a fair comparison among the algorithms. All the classification performances obtained by the different parameters of the algorithms are available at http://sites.labic.icmc.usp.br/ragero/jipm_2015/complete_results/. A detailed analysis about the performance of TCBHN with different parameter values are presented in Appendix A.

Figs. 2 and 3 present *Micro*-F^1 values and Figs. 4 and 5 present *Macro*-F^1 values obtained by the different algorithms and number of labeled document per class. We displayed MNB algorithm and transductive algorithms based on MNB (MNB-Se, MNB-Co, and EM) in red-scale. TSVM is displayed in blue. The transductive algorithms based on networks used for comparison with TCBHN are displayed in green-scale. TCBHN is displayed in black line and its supervised version (IMBHN^R) is displayed in a gray line. The numerical values used to generate the charts are presented in Appendix B.

TCBHN obtained the highest or close to the highest *Micro*-F^1 and *Macro*-F^1 values. TCBHN also obtained higher classification performance than its inductive version IMBHN, which indicates a classification improvement when using unlabeled documents. On the other hand, EM, MNB-Se, MNB-Co surpass the *Micro*-F^1 and *Macro*-F^1 values of MNB just in few collections. For instance, EM surpasses MNB in Classic4, Oh0, and Oh10 collections. MNB-Co and MMB-Se surpass MNB in Oh0, Oh10 and Tr41 collections. However, MNB surpasses MNB-Se, MNB-Co and EM in several text collections. This indicates that the proposed approach makes a better use of unlabeled documents to improve classification performance than vector space model algorithms.

TSVM obtained the highest or close to the highest *Micro*-F^1 and *Macro*-F^1 values just in few collections such as Multi Domain Sentiment and Review Polarity. In general, network-based algorithms were better than VSM-based algorithms. The label propagation algorithms based on document networks were better than existing algorithms based on bipartite networks mainly when using few labeled documents per class.

Micro-F^1 and *Macro*-F^1 values tended to increase as the number of labeled documents grows. For most of the collections there was a significantly increase in *Micro*-F^1 and *Macro*-F^1 values when moving from 1 labeled document to 10 labeled documents per class, and a slightly increasing when using more than 10 labeled documents. There were some decreases in *Macro*-F^1 values as we increased the number of labeled documents. This fact occurs just for small text collections in which all documents of a class were selected as labeled documents. In this case, the values of precision and recall for a class are 0 since there were no documents available for the test.

We submitted the data presented on Figs. 2–5 (or the tables presented on Appendix B) to Friedman test and Li's post hoc test with 95% of confidence level to assess statistically significant differences (SSD) among the classification algorithms.[7] This

[7] Friedman test is a non-parametric test based on average ranking differences. It ranks the algorithms for each text collection individually, in which the algorithm with the highest performance have the rank of 1, the second best performance 2, and so on. In the case of ties average ranks are assigned. Then the average ranking is computed for each algorithm considering the ranks in each text collection. If there are statistically significant differences on the rankings, the Li's post test is used to find pairs of algorithms which produce differences.

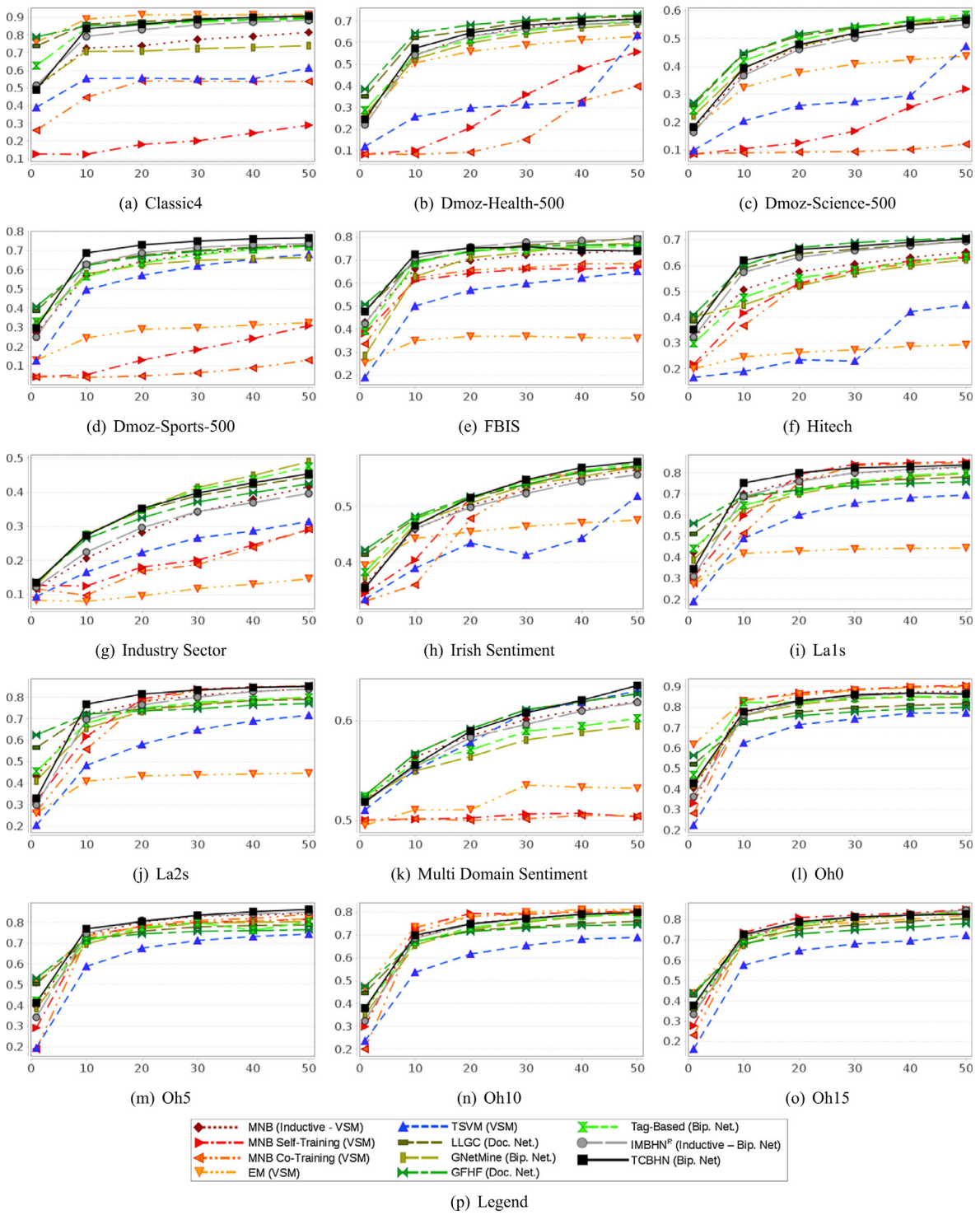


Fig. 2. *Micro-F1*: *x*-axis presents the number of labeled documents per class and *y*-axis presents *Micro-F1* values.

is an advisable statistically significant difference test to use when there is a control algorithm (usually the proposed one) and results from multiple datasets (García, Fernández, Luengo, & Herrera, 2010; Trawinski, Smetek, Telec, & Lasota, 2012). The null hypothesis states that all the algorithms performed equivalently and therefore their ranks should be equal. In our case we want to determine if the proposed algorithm obtained better results with statistically significant differences in comparison to other algorithms.

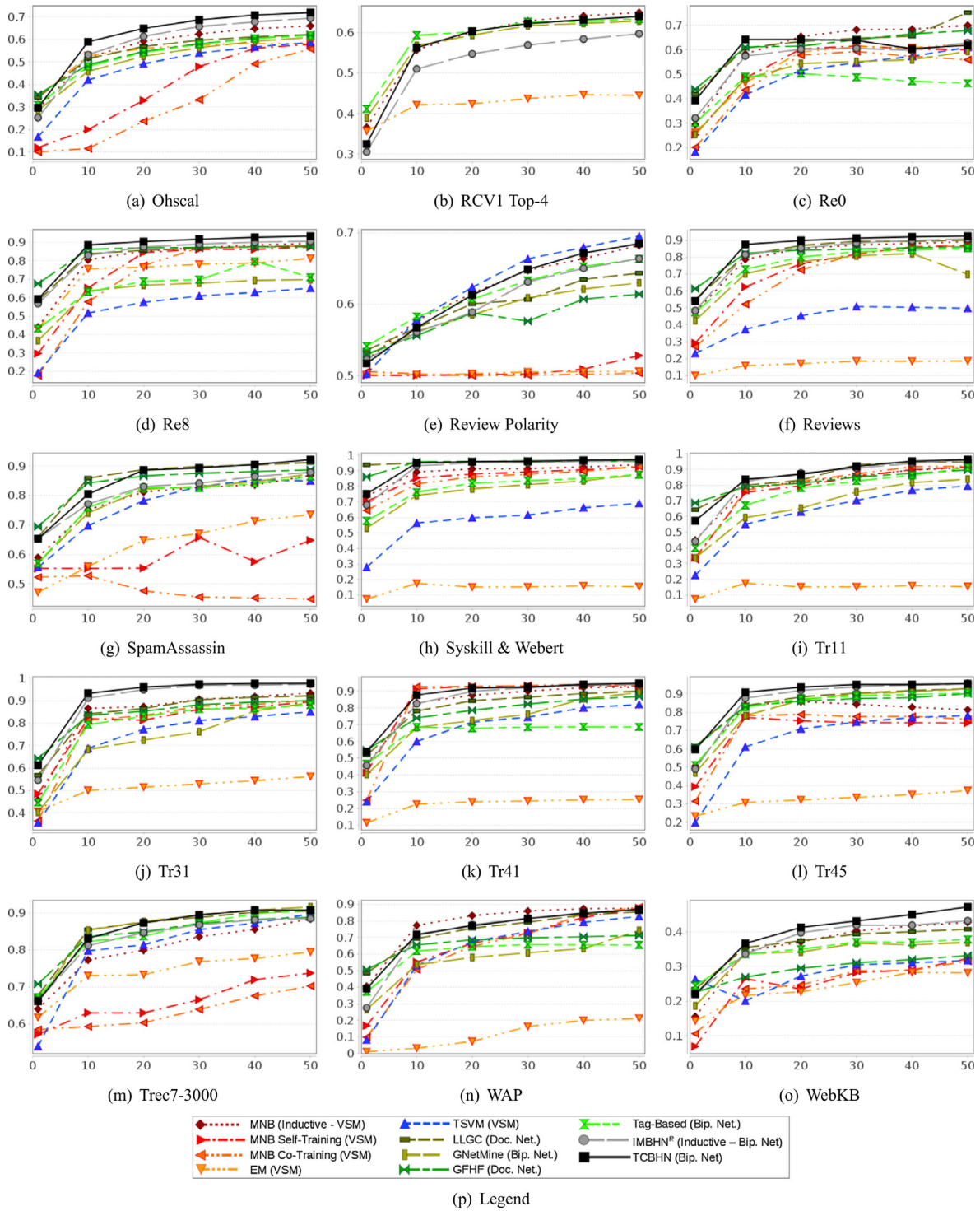


Fig. 3. *Micro-F1*: *x*-axis presents the number of labeled documents per class and *y*-axis presents *Micro-F1* values.

In Tables 3 and 4 we present the results of the statistical test for *Micro-F*^{1} and *Macro-F*^{1} values respectively. In these tables we presents the average rank (AR), the general rank (GR), i.e., the ranking of the algorithms considering the average rank, the *p*-value, and the value of *p* which produces statistically significant differences (SSD). The results with SSD are highlighted in italic.

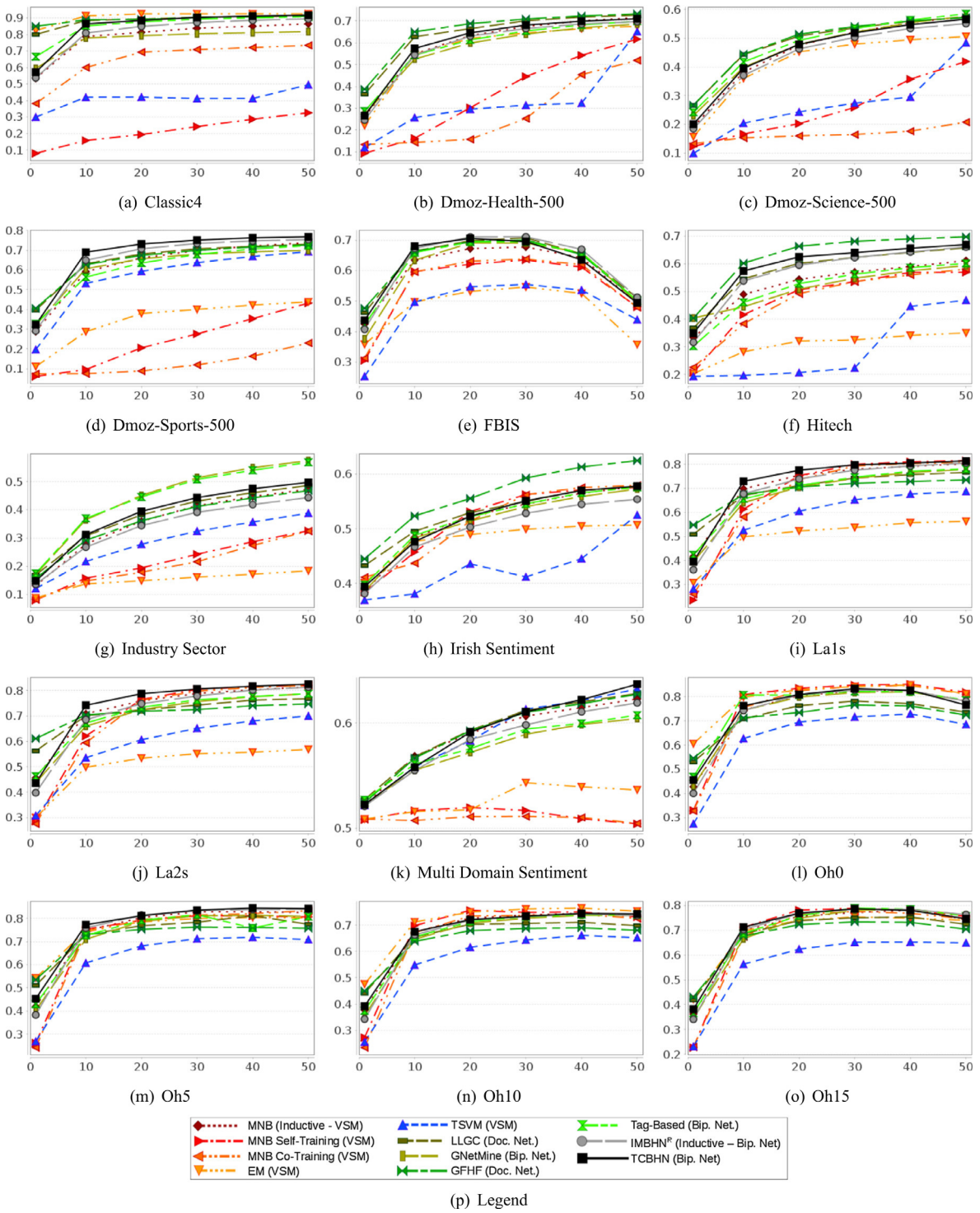


Fig. 4. *Macro-F1*: *x*-axis presents the number of labeled documents per class and *y*-axis presents *Macro-F1* values.

The control algorithm is the one with the best average ranking by default. GFHF algorithm presented the best average ranking with SSD for all algorithms except LLGC when using one labeled documents for each class. When using 10 or more, TCBHN presented the best average ranking and SSD for all algorithms except LLGC. Using 20, 30 and 40 labeled document for each class, TCBHN presented a better average ranking than all algorithms and SSD for all algorithms. Finally, when using 50

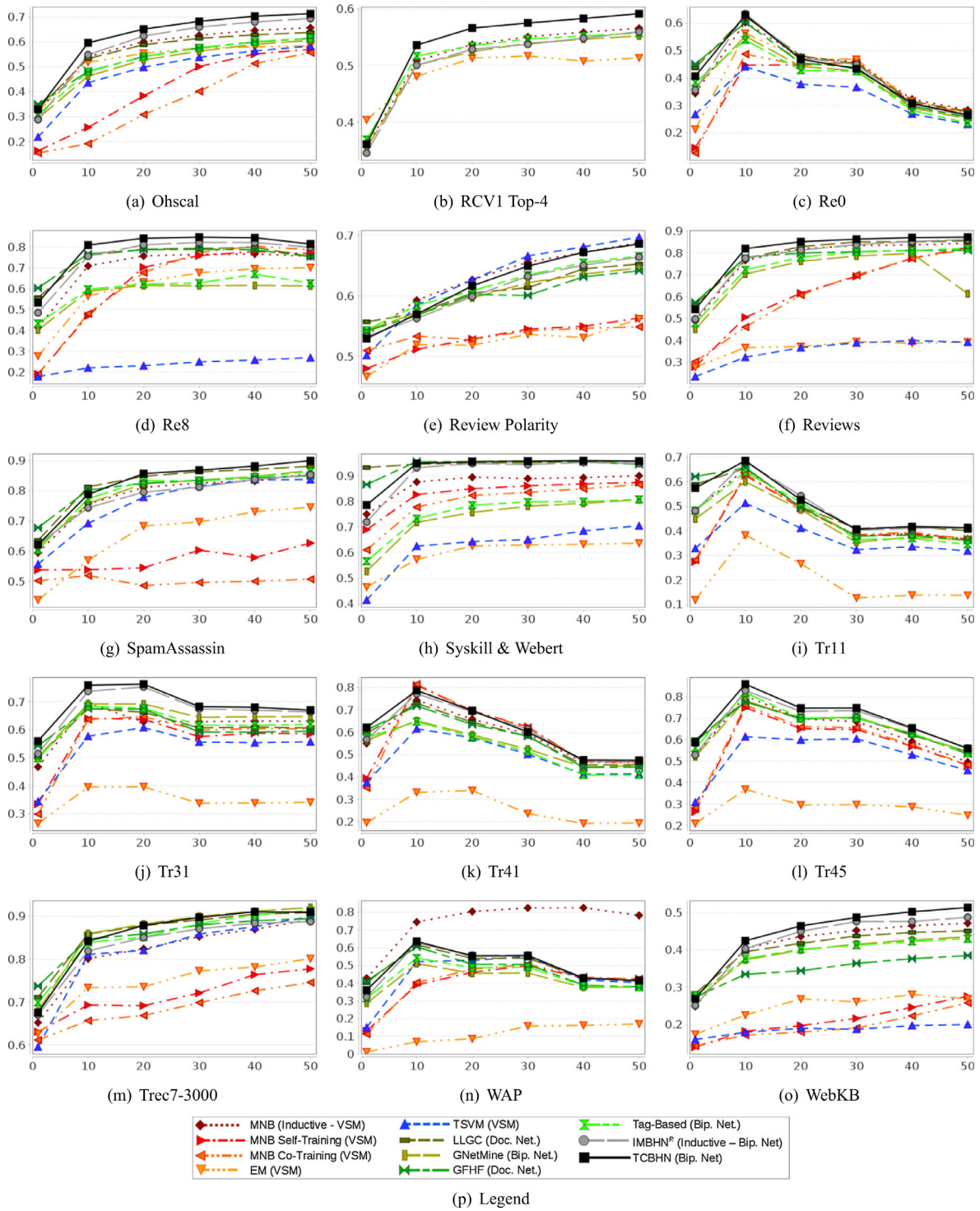


Fig. 5. *Macro-F1*: *x*-axis presents the number of labeled documents per class and *y*-axis presents *Macro-F1* values.

labeled documents, TCBHN also presented a better average ranking than all algorithms and SSD for all algorithms except IMBHN^R for *Micro-F*[1].

The average rankings and the results of the statistical significant tests allowed us to conclude that TCBHN has a better classification performance with statistically significant differences than other algorithms based on vector space model or networks. Moreover, TCBHN surpasses the classification performance of the inductive supervised learning algorithms. Thus,

Table 3
Average ranking (AR), general ranking (GR) and *p*-value considering *Micro-F*[1] values.

Alg.	1 labeled doc			10 labeled docs			20 labeled docs			30 labeled docs			40 labeled docs			50 labeled docs		
	AR	GR	*p*-value	AR	GR	*p*-value	AR	GR	*p*-value	AR	GR	*p*-value	AR	GR	*p*-value	AR	GR	*p*-value
TCBHN	4.55	4th	0.000315	2.38	1st	–	2.21	1st	–	2.33	1st	–	2.34	1st	–	2.69	1st	–
IMBHN^{R}	6.14	6th	0	4.41	4th	0.019500	3.95	2nd	0.045574	4.21	2nd	0.030953	4.10	2nd	0.043476	3.93	2nd	0.154,084
TB	4.38	3rd	0.000662	5.76	6th	0.000105	6.31	6th	0.000002	6.21	6th	0.000008	6.28	6th	0.000006	6.26	6th	0.000042
GFHF	1.41	1st	–	4.17	3rd	0.039523	5.07	5th	0.001016	5.47	5th	0.000315	5.76	5th	0.000089	6.00	5th	0.000144
GM	6.21	7th	0	7.14	7th	0	7.38	8th	0	7.22	9th	0	7.00	9th	0	6.60	7th	0.000007
LLGC	2.38	2nd	0.267,633	3.69	2nd	0.132,470	4.24	3rd	0.019500	4.26	3rd	0.026619	4.52	3rd	0.012624	4.59	3rd	0.029445
TSVM	9.38	10th	0	9.45	11th	0	9.07	11th	0	8.79	11th	0	8.86	11th	0	8.59	10th	0
EM	7.62	8th	0	8.29	9th	0	8.62	10th	0	8.72	10th	0	8.79	10th	0	8.90	11th	0
MNB-Co	9.64	11th	0	8.55	10th	0	7.62	9th	0	7.21	8th	0	7.00	8th	0	6.93	9th	0.000001
MNB-Se	8.66	9th	0	7.53	8th	0	6.86	7th	0	6.76	7th	0	6.69	7th	0.000001	6.90	8th	0.000001
MNB	5.64	5th	0.000001	4.62	5th	0.010071	4.67	4th	0.004644	4.83	4th	0.004101	4.66	4th	0.007989	4.62	4th	0.026619
	SSD $p \leq 0.038546$			SSD $p \leq 0.045659$			SSD $p \leq 0.05$			SSD $p \leq 0.049931$			SSD $p \leq 0.05$			SSD $p \leq 0.044522$		

Table 4
Average ranking (AR), general ranking (GR) and p-value considenring $Macro\text{-}F^1$ values.

Alg.	1 labeled doc			10 labeled docs			20 labeled docs			30 labeled docs			40 labeled docs			50 labeled docs		
	AR	GR	p-value	AR	GR	p-value	AR	GR	p-value	AR	GR	p-value	AR	GR	p-value	AR	GR	p-value
TCBHN	4.10	3rd	0.004644	2.48	1st	–	2.36	1st	–	2.62	1st	–	2.50	1st	–	2.62	1st	–
IMBHN^R	6.31	7th	0	4.59	5th	0.015735	4.28	3rd	0.028001	4.33	2nd	0.032536	4.33	2nd	0.035880	4.38	2nd	0.043476
TB	4.34	4th	0.001885	5.45	6th	0.000662	5.74	6th	50.000105	5.90	6th	0.000169	5.90	5th	0.000096	5.69	5th	0.000426
GFHF	1.64	1st	–	4.38	4th	0.029445	5.26	5th	0.000882	5.66	5th	0.000494	5.97	6th	0.000069	6.41	7th	0.000013
GM	5.50	5th	0.000009	6.59	7th	0.000002	6.95	8th	0	6.55	7th	0.000006	6.22	7th	0.000019	5.93	6th	0.000144
LLGC	2.24	2nd	0.488415	4.17	2nd	0.052388	4.26	2nd	0.029445	4.55	3rd	0.026619	4.71	4th	0.011284	4.90	4th	0.008976
TSVM	9.31	9th	0	9.38	11th	0	9.38	11th	0	9.10	11th	0	9.21	11th	0	9.07	11th	0
EM	7.66	8th	0	7.98	9th	0	8.34	10th	0	8.31	10th	0	8.48	10th	0	8.48	10th	0
MNB-Co	9.41	10th	0	8.79	10th	0	7.84	9th	0	7.21	9th	0	7.14	9th	0	7.21	9th	0
MNB-Se	9.66	11th	0	7.90	8th	0	6.93	7th	0	6.90	8th	0.000001	6.86	8th	0.000001	6.93	8th	0.000001
MNB	5.38	6th	0.000002	4.29	3rd	0.037664	4.66	4th	0.008469	4.72	4th	0.015735	4.69	3rd	0.011937	4.38	2nd	0.043476
	SSD $p \leq 0.02693$			SSD $p \leq 0.04987$			SSD $p \leq 0.05$			SSD $p \leq 0.05$			SSD $p \leq 0.05$			SSD $p \leq 0.05$		

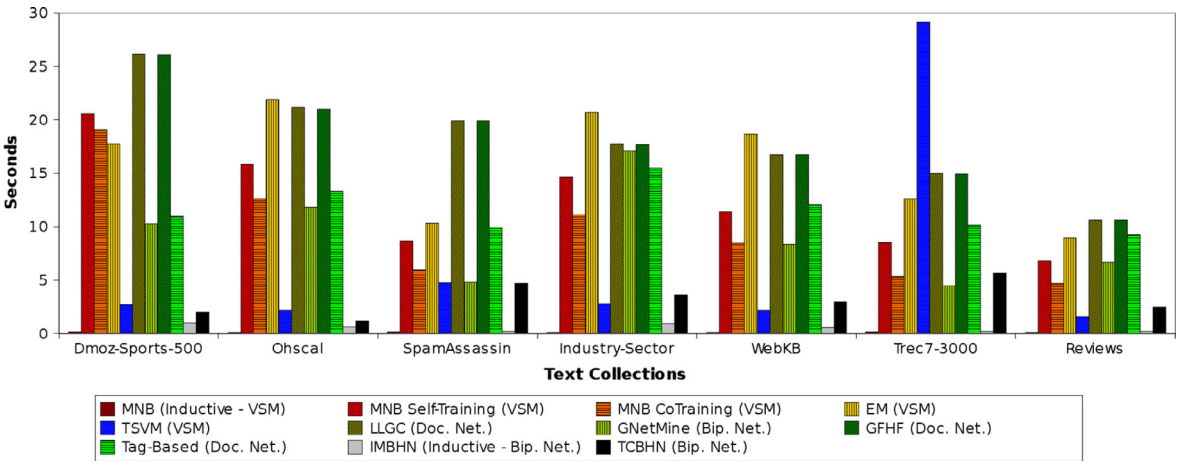


Fig. 6. Classification time.

this is an advisable algorithm to classify text collections in which all documents and few labeled examples are available. TCBHN only did not surpass the average ranking of the algorithms based on document networks (GFHF and LLGC) just using 1 labeled document per class. However, *Micro-F*^{1} and *Macro-F*^{1} values were significantly inferior than the obtained by other number of labeled documents as presented before.

We also analysed the classification time to analyse the scalability of TCBHN and the cost to move from inductive to transductive learning. For this analysis we considered 5 collections which have the highest number of documents (Dmoz-Sports-500, Ohscal, SpamAssassin, Industry Sector, WebKB) and the 5 collections with the highest number of terms (Trec7-3000, SpamAssassin, Reviews, Industry Sector, WebKB) and ran our proposed algorithm and the algorithms used for comparison. We obtained the classification time in a computer with Intel Xeon E5-2690 v2 3.00 GHz processor with 128 Gb of ram memory. We incorporated the time to build the networks for network-based algorithms. We consider the best parameter of each algorithm for each collection to obtain the classification time. We just consider the Mutual KNN approach to generate document networks since this speeds up the classification time of algorithms based on document networks and make them more competitive.

Fig. 6 presents the square root of the classification time (seconds) to better visualize the results. In general, the algorithms based on bipartite networks presented a lower computation time than algorithms based on document networks even using Mutual KNN networks. Computing similarities and building networks corresponds about 98% of the computations time obtained by algorithms based on document networks. Therefore, building document networks has a high impact in the classification time.

Splitting the feature set into two disjoint sets speed up transductive learning, as we can notice in the comparison between Co-Training and Self-Training. In general EM obtained a higher computation time than Self-Training and Co-Training. The iterative solution for TSVM was one of the fastest solutions for transductive learning, except for TREC7-3000 collection. However, TCBHN was faster than other transductive learning algorithms for most of the collections.

Transductive learning algorithms have to deal with a higher number of documents and terms in comparison with inductive supervised learning algorithms. The proposed approach presented a lower increase in the computational time than other approaches when dealing with unlabeled documents. For instance, the computation cost of TCBHN was 400% higher than IMBHN^R for Dmoz-Sports-500 collection even considering 4900% more documents. For WebKB, there was an increase of 3000% in the computational time but considering 11,700% more documents. Oh the other hand, Self-Training and EM approach had an increase in the computation time of 31,736,000% and 2,359,600% respectively in comparison to MNB for Dmoz-Sports-500 collection, and 2,295,000% and 6,154,300% for WebKB collection. Therefore, our proposal had a lower increase in computational time than other algorithms when incorporating unlabeled documents, and had a lower computational cost in general that other transductive learning algorithms.

The analysis carried out in this article demonstrate that our proposal is advisable in practical situations since it presents higher classification performance and lower classification time than other algorithms for most of the evaluated collections. Besides, TCBHN uses the structure of a bipartite heterogeneous network, which does not require the definition of parameters or additional computations to generate such network, which also makes its application easier in practical situation.

5. Conclusions and future work

In this article we presented an algorithm which uses the structure of a bipartite heterogeneous network to perform transductive classification of texts. The proposed algorithm, named TCBHN (Transductive Classification based on Bipartite Heterogeneous Networks) obtains the relevance scores of terms for classes through an optimization process considering

the labels of labeled documents and the labels assigned to unlabeled documents. The induced relevance scores are propagated and employed to assign new labels to the unlabeled documents. Optimization and label propagation are repeated until there is no change in the labels assigned for unlabeled documents.

We demonstrated that optimization plus label propagation lead to better classification performance than just label propagation in document or bipartite networks. The proposed algorithm also obtained a better classification performance than algorithms based on vector space model and proved to make better use on unlabeled documents to improve classification

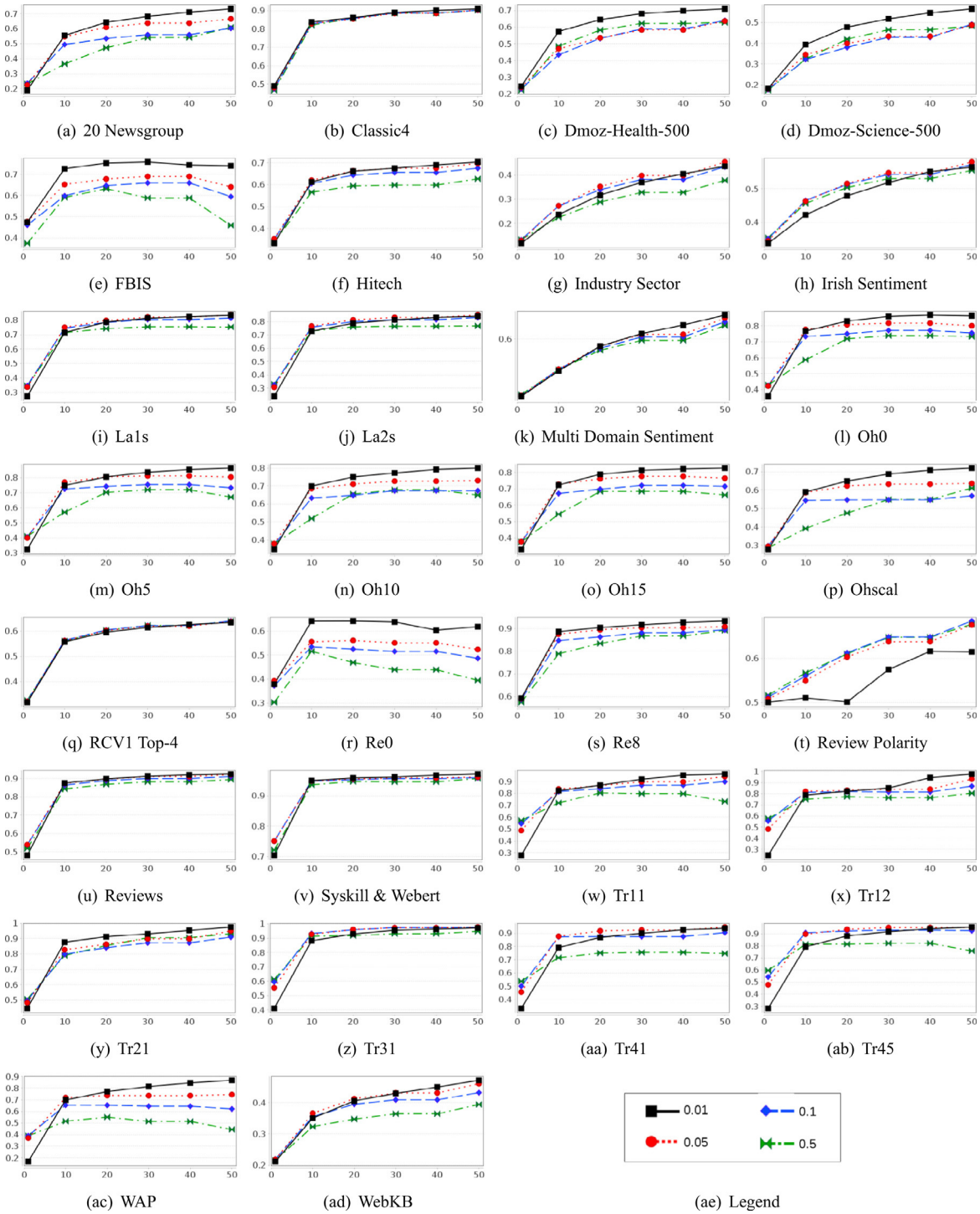


Fig. A.7. *Micro-F1*: *x*-axis presents the number of labeled documents per class and *y*-axis presents *Micro-F1* values.

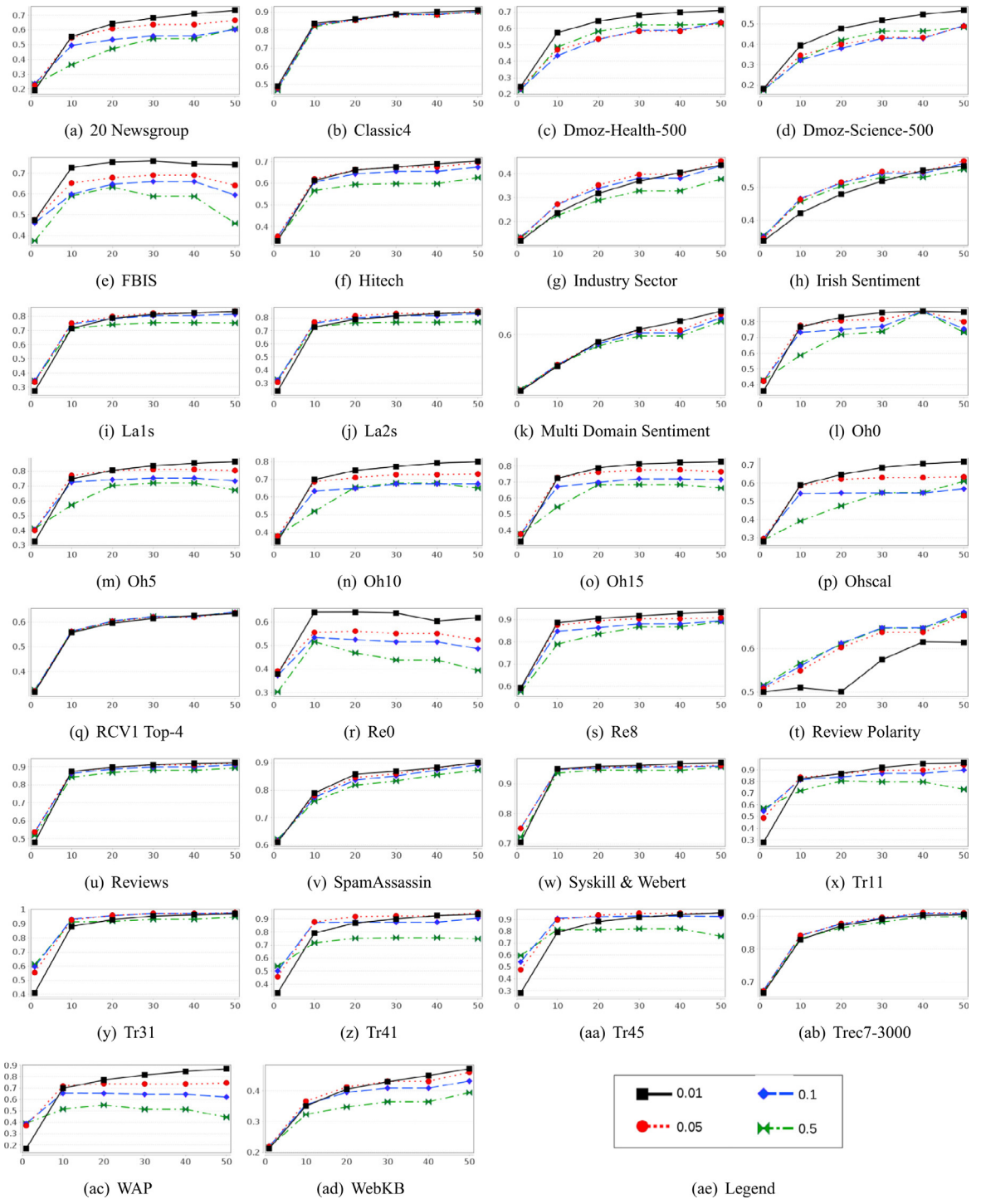
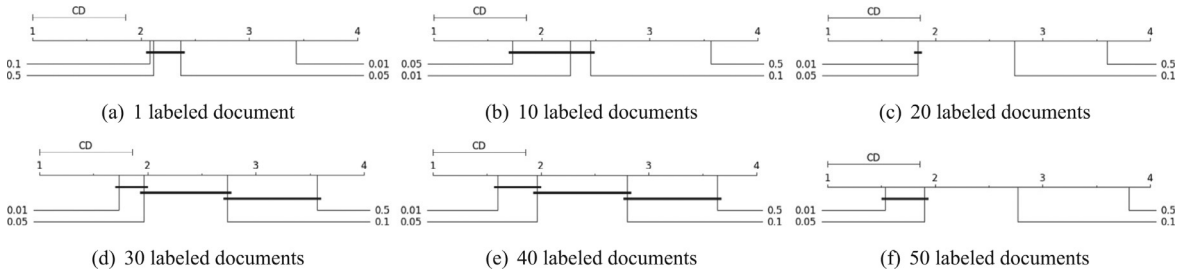


Fig. A.8. *Macro-F1*: *x*-axis presents the number of labeled documents per class and *y*-axis presents *Micro-F1* values.

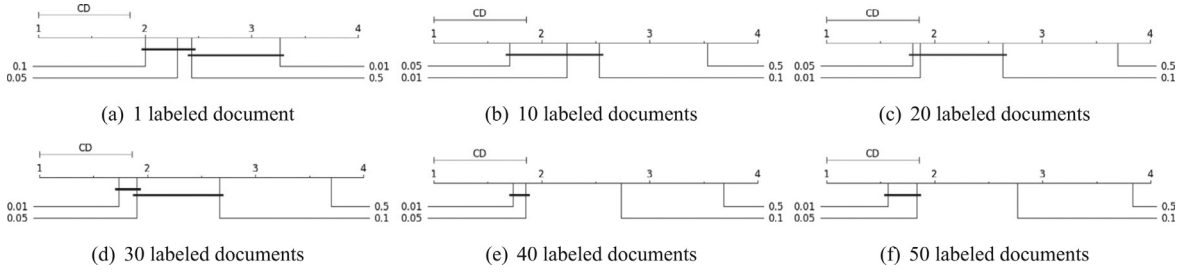
than other algorithms. TCBHN also presented better results with statistically significant differences than other algorithms used for comparisons when using more than 10 labeled documents per class. Moreover, TCBHN presents a lower classification time than other algorithms, which makes it useful for classification in large text collections in which all documents are known.



(a) 1 labeled document (b) 10 labeled documents (c) 20 labeled documents

(d) 30 labeled documents (e) 40 labeled documents (f) 50 labeled documents

Fig. A.9. Critical difference diagrams for Micro-F^1 considering different values of η.



(a) 1 labeled document (b) 10 labeled documents (c) 20 labeled documents

(d) 30 labeled documents (e) 40 labeled documents (f) 50 labeled documents

Fig. A.10. Critical difference diagrams for Macro-F^1 considering different values of η.

As future work we intend to: (i) incorporate other types of relations as `document-document` or `term-term` with `document-term` relations and analyse the impact in the classification performance and (ii) use the relevance scores of terms induced by TCBHN to classify unseen/new documents.

Acknowledgements

Grants 2011/12823-6, 2011/22749-8 and 2014/08996-0, Sao Paulo Research Foundation (FAPESP).

Appendix A. Analisys of the parameter η in TCBHN algorithm

We also analyse the impact of the parameter η in the accuracies obtained by TCBHN. In this article we used $\eta = \{0.01, 0.05, 0.1, 0.5\}$. Figs. A.7 and A.8 present *Micro*-F^1 and *Macro*-F^1 values obtained by the different values of η and 1, 10, 20, 30, 40, and 50 labeled document per class.

$\eta = 0.01$ usually obtained lower *Micro*-F^1 and *Macro*-F^1 than other values of η when using one labeled document for each class. On the other hand, $\eta = 0.01$ obtained the highest *Micro*-F^1 and *Macro*-F^1 for most of the text collections when using 10 or more labeled documents for each class. $\eta = 0.05$ presented better results than $\eta = 0.01$ for 1 labeled document and close results for more than 10 labeled documents for each class. $\eta = \{0.1, 0.5\}$ presented better results than $\eta = \{0.01, 0.05\}$ for 1 labeled documents and lower results when using 10 or more labeled documents for each class. Therefore, high values of η obtained better classification performance than the small values for few labeled documents, and small values of η are better the high values for 10 or more labeled documents per class.

We submitted the data presented in Figs. A.7 and A.8 to $N \times N$ Friedman test and Nemenyi's post hoc test with 95% of confidence level to assess statistical significant differences (SSD) among the results with different values of η. This is an advisable test to compare the results from different algorithms and multiple datasets when there is not a control algorithm (Demsar, 2006). In the Figs. A.9 and A.10 we present the critical difference diagrams for *Micro*-F^1 and *Macro*-F^1 to illustrate the results of the statistical significance test. These diagrams present the average ranks obtained by the different values of η and the values connected by a line do not present statistically significant differences (SSD) among them.

$\eta = 0.1, 0.5$ present better results with SSD than $\eta = 0.01$ for 1 labeled document per class. For 10 labeled documents, $\eta = 0.05$ obtained the best average ranking and $\eta = \{0.01, 0.05, 0.1\}$ obtained better results with SSD than $\eta = 0.05$. $\eta = 0.01$ and $\eta = 0.05$ obtained the same average ranking for *Micro*-F^1 and $\eta = 0.05$ obtained a better average ranking than $\eta = 0.01$ for *Macro*-F^1 when using 20 labeled documents for each class. For 30, 40, and 50 labeled documents, $\eta = 0.01$ presented a better average ranking than $\eta = 0.05$ but there are not SSD between them. Both of them presented better results with SSD than $\eta = 0.5$.

Table B.5
Micro-F[1] values considering 1 labeled document for each class.

Collections	MNB	MNB-Se	MNB-Co	EM	TSVM	LLGC	GM	GFHF	TM	IMBHN[R]	TCBHN
Classic4	0.4993	0.1264	0.2613	0.7600	0.3908	0.7380	0.5126	**0.7894**	0.6271	0.5152	0.4907
Dmoz Health-500	0.2561	0.0833	0.0842	0.2430	0.1194	0.3518	0.2676	**0.3850**	0.2875	0.2199	0.2452
Dmoz Science-500	0.1798	0.0850	0.0868	0.1734	0.0991	0.2616	0.2229	**0.2696**	0.2421	0.1636	0.1830
Dmoz Sports-500	0.2720	0.0438	0.0429	0.1304	0.1274	0.3869	0.3191	**0.4093**	0.3309	0.2502	0.2973
FBIS	0.4316	0.3825	0.3363	0.2543	0.1899	0.4803	0.2849	**0.5081**	0.3922	0.4237	0.4764
Hitech	0.3217	0.2179	0.2099	0.2003	0.1668	0.3844	0.4016	**0.4089**	0.2988	0.3233	0.3532
Industry Sector	0.1164	0.1264	0.1164	0.0826	0.0935	0.1320	0.1274	**0.1370**	0.1330	0.1213	0.1344
IrishSentiment	0.3616	0.3453	0.3306	0.3952	0.3342	0.4144	0.3746	**0.4228**	0.3833	0.3561	0.3544
La1s	0.4233	0.3065	0.2782	0.2722	0.1915	0.5113	0.3879	**0.5607**	0.4417	0.3100	0.3458
La2s	0.4329	0.3186	0.2681	0.2617	0.2054	0.5659	0.4115	**0.6245**	0.4561	0.2996	0.3308
Multi Domain Sentiment	0.5197	0.5001	0.5000	0.4949	0.5101	**0.5247**	0.5207	0.5198	0.5240	0.5181	0.5183
Oh0	0.4049	0.3303	0.2815	**0.6176**	0.2231	0.5204	0.4201	0.5640	0.4695	0.3627	0.4287
Oh10	0.3690	0.2993	0.2002	0.4681	0.2357	0.4466	0.3499	**0.4759**	0.3805	0.3238	0.3798
Oh15	0.3544	0.2780	0.2313	0.4378	0.1628	0.4278	0.3384	**0.4384**	0.3756	0.3342	0.3774
Oh5	0.3961	0.2911	0.1882	0.5093	0.1931	0.5039	0.3857	**0.5311**	0.4229	0.3424	0.4124
Ohscal	0.2914	0.1201	0.1006	0.3455	0.1673	0.3424	0.2855	**0.3559**	0.3093	0.2530	0.2966
RCV1 Top-4	0.3661	–	–	0.3568	–	–	0.3891	–	**0.4115**	0.3052	0.3248
Re0	0.2950	0.2531	0.2018	0.2627	0.1830	0.4193	0.2538	**0.4384**	0.3015	0.3211	0.3928
Re8	0.4406	0.2963	0.1763	0.4307	0.1914	0.5831	0.3666	**0.6756**	0.4321	0.5672	0.5918
Review Polarity	0.5221	0.5006	0.5057	0.5015	0.5024	0.5359	0.5286	0.5302	**0.5413**	0.5227	0.5171
Reviews	0.4754	0.2918	0.2701	0.0988	0.2306	0.5510	0.4251	**0.6125**	0.4729	0.4835	0.5391
SpamAssassin	0.5895	0.5522	0.5223	0.4708	0.5560	0.6577	0.5699	**0.6943**	0.5693	0.6532	0.6530
Syskill & Webert	0.7370	0.7024	0.6449	0.0738	0.2794	0.9371	0.5306	**0.8600**	0.5776	0.6797	0.7509
Tr11	0.4494	0.3378	0.3269	0.0738	0.2259	0.6432	0.3363	**0.6867**	0.3995	0.4415	0.5728
Tr31	0.4568	0.4846	0.3642	0.4053	0.3554	0.5678	0.4026	**0.6426**	0.4459	0.5453	0.6126
Tr41	0.4616	0.4145	0.2490	0.1143	0.2408	0.5184	0.4026	**0.5479**	0.4669	0.4569	0.5386
Tr45	0.5065	0.3938	0.3143	0.2332	0.1969	**0.6147**	0.4693	0.6094	0.5119	0.4900	0.5963
Trec7-3000	0.6401	0.5713	0.5846	0.6172	0.5390	0.6765	0.6628	**0.7078**	0.6715	0.6627	0.6606
WAP	0.4049	0.1669	0.0969	0.0096	0.0814	0.4847	0.2653	**0.5058**	0.3718	0.2746	0.3908
WebKB	0.1549	0.0689	0.1054	0.2161	**0.2629**	0.2342	0.1860	0.2251	0.2444	0.2211	0.2189

Table B.6
Micro-F[1] values considering 10 labeled documents for each class.

Collections	MNB	MNB-Se	MNB-Co	EM	TSVM	LLGC	GM	GFHF	TM	IMBHN[R]	TCBHN
Classic4	0.7259	0.1247	0.4472	**0.8900**	0.5544	0.8618	0.7082	0.8529	0.8374	0.7916	0.8364
Dmoz Health-500	0.5449	0.1009	0.0833	0.5065	0.2573	0.6225	0.5252	**0.6453**	0.5458	0.5424	0.5743
Dmoz Science-500	0.3762	0.1048	0.0898	0.3244	0.2043	0.4453	0.3947	**0.4475**	0.4196	0.3668	0.3941
Dmoz-Sports-500	0.5710	0.0540	0.0401	0.2460	0.4969	0.6292	0.5798	0.6227	0.5655	0.6288	**0.6880**
FBIS	0.6623	0.6115	0.6229	0.3500	0.5010	0.6940	0.6282	0.6935	0.6850	0.7085	**0.7262**
Hitech	0.5073	0.4161	0.3675	0.2463	0.1899	0.5898	0.4477	0.6012	0.4768	0.5747	**0.6206**
Industry Sector	0.2062	0.1247	0.0973	0.0796	0.1656	**0.2761**	0.2759	0.2636	0.2746	0.2244	0.2728
IrishSentiment	0.4590	0.4046	0.3602	0.4430	0.3899	0.4793	0.4680	**0.4826**	0.4773	0.4609	0.4656
La1s	0.7003	0.5971	0.5132	0.4190	0.4899	0.6823	0.6250	0.6901	0.6465	0.6869	**0.7514**
La2s	0.7203	0.6170	0.5585	0.4097	0.4834	0.7196	0.6560	0.7220	0.6795	0.6959	**0.7675**
Multi Domain Sentiment	0.5635	0.5009	0.5014	0.5103	0.5506	**0.5668**	0.5495	0.5629	0.5574	0.5534	0.5555
Oh0	0.7731	**0.8327**	0.7749	0.8312	0.6251	0.7230	0.7586	0.7287	0.8229	0.7607	0.7780
Oh10	0.6908	0.7326	0.7109	**0.7365**	0.5369	0.6726	0.6567	0.6721	0.6691	0.6858	0.6993
Oh15	0.7143	**0.7359**	0.6944	0.7267	0.5754	0.7002	0.6737	0.6814	0.7026	0.7188	0.7272
Oh5	0.7413	0.7363	0.7215	0.7363	0.5884	0.7165	0.6963	0.7214	0.7128	0.7471	**0.7699**
Ohscal	0.5203	0.2003	0.1155	0.5277	0.4205	0.5149	0.4584	0.4876	0.4805	0.5322	**0.5898**
RCV1 Top-4	0.5575	–	–	0.4217	–	–	0.5683	–	**0.5935**	0.5104	0.5627
Re0	0.5828	0.4846	0.4356	0.4650	0.4164	0.6067	0.4812	0.6105	0.4896	0.5742	**0.6416**
Re8	0.8056	0.6531	0.5771	0.7562	0.5163	0.8382	0.6363	0.8625	0.6313	0.8294	**0.8863**
Review Polarity	0.5773	0.5006	0.5028	0.5016	0.5785	0.5668	0.5623	0.5559	**0.5830**	0.5605	0.5674
Reviews	0.7863	0.6233	0.5214	0.1574	0.3725	0.8101	0.7009	0.8231	0.7281	0.8150	**0.8744**
SpamAssassin	0.7516	0.5523	0.5275	0.5589	0.6973	**0.8587**	0.7404	0.8428	0.7591	0.7711	0.8045
Syskill & Webert	0.8915	0.8537	0.8155	0.1753	0.5629	0.9492	0.7405	**0.9578**	0.7622	0.9306	0.9497
Tr11	0.7784	0.7530	0.7670	0.1753	0.5506	0.7988	0.5930	0.7890	0.6713	0.8235	**0.8357**
Tr31	0.8657	0.8193	0.8135	0.5000	0.6874	0.8413	0.6827	0.8346	0.7932	0.9110	**0.9329**
Tr41	0.8304	0.9139	**0.9259**	0.2252	0.6000	0.7818	0.6827	0.7402	0.6858	0.8248	0.8770
Tr45	0.8392	0.7710	0.7809	0.3076	0.6108	0.8347	0.7807	0.8327	0.8273	0.8758	**0.9081**
Trec7-3000	0.7729	0.6299	0.5925	0.7304	0.7973	**0.8548**	0.8530	0.8361	0.8231	0.8118	0.8312
WAP	**0.7731**	0.5519	0.5061	0.0311	0.5303	0.6933	0.5355	0.6552	0.6173	0.7122	0.7186
WebKB	0.3433	0.2644	0.2349	0.2266	0.2001	0.3544	0.3367	0.2690	0.3347	0.3354	**0.3659**

Table B.7
Micro-F[1] values considering 20 labeled documents for each class.

Collections	MNB	MNB-Se	MNB-Co	EM	TSVM	LLGC	GM	GFHF	TM	IMBHN[R]	TCBHN
Classic4	0.7395	0.1806	0.5406	**0.9151**	0.5566	0.8764	0.7092	0.8653	0.8578	0.8304	0.8605
Dmoz Health-500	0.6293	0.2065	0.0923	0.5599	0.2985	0.6569	0.5995	**0.6824**	0.6148	0.6293	0.6454
Dmoz Science-500	0.4693	0.1259	0.0927	0.3778	0.2597	0.5098	0.4735	**0.5162**	0.4939	0.4610	0.4783
Dmoz Sports-500	0.6438	0.1307	0.0475	0.2913	0.5713	0.6775	0.6265	0.6715	0.6363	0.6884	**0.7302**
FBIS	0.6985	0.6433	0.6573	0.3690	0.5705	0.7400	0.7115	0.7375	0.7451	**0.7574**	0.7529
Hitech	0.5770	0.5317	0.5266	0.2639	0.2348	0.6458	0.5221	**0.6704**	0.5522	0.6326	0.6634
Industry Sector	0.2819	0.1806	0.1689	0.0957	0.2232	0.3454	0.3491	0.3247	0.3469	0.2960	**0.3522**
Irish Sentiment	0.5026	0.5158	0.4787	0.4551	0.4350	0.5150	0.5064	**0.5180**	0.5155	0.4985	0.5159
La1s	0.7664	0.7921	0.7558	0.4297	0.6006	0.7193	0.7007	0.7197	0.7121	0.7590	**0.7991**
La2s	0.7781	0.7917	0.7795	0.4346	0.5806	0.7457	0.7339	0.7368	0.7492	0.7645	**0.8146**
Multi Domain Sentiment	0.5848	0.5022	0.4998	0.5105	0.5779	**0.5918**	0.5638	0.5890	0.5706	0.5831	0.5894
Oh0	0.8323	**0.8700**	0.8576	0.8639	0.7144	0.7758	0.8134	0.7572	0.8229	0.8273	0.8308
Oh10	0.7511	**0.7954**	0.7818	0.7813	0.6174	0.7212	0.7264	0.7161	0.7335	0.7476	0.7499
Oh15	0.7785	**0.8097**	0.7915	0.7683	0.6471	0.7530	0.7638	0.7292	0.7766	0.7847	0.7884
Oh5	0.8028	0.7825	0.7870	0.7809	0.6763	0.7591	0.7698	0.7451	0.7760	**0.8107**	0.8049
Ohscal	0.5920	0.3296	0.2370	0.5599	0.4919	0.5686	0.5257	0.5466	0.5423	0.6138	**0.6489**
RCV1 Top-4	0.6020	–	–	0.4242	–	–	0.5936	–	0.6018	0.5477	**0.6039**
Re0	**0.6536**	0.6028	0.5779	0.5912	0.5161	0.6341	0.5434	0.6145	0.5029	0.6017	0.6420
Re8	0.8522	0.8448	0.7747	0.7652	0.5748	0.8565	0.6676	0.8705	0.6879	0.8758	**0.9046**
Review Polarity	0.6152	0.5015	0.5002	0.5030	**0.6236**	0.6006	0.5853	0.5879	0.6072	0.5889	0.6128
Reviews	0.8504	0.7610	0.7237	0.1695	0.4528	0.8688	0.7739	0.8403	0.8002	0.8538	**0.8975**
SpamAssassin	0.8118	0.5532	0.4758	0.6479	0.7824	**0.8881**	0.8213	0.8660	0.8260	0.8300	0.8860
Syskill & Webert	0.9118	0.8780	0.8598	0.1514	0.5972	0.9547	0.7831	**0.9591**	0.8201	0.9516	0.9583
Tr11	0.8226	0.7953	0.8234	0.1514	0.6296	0.8307	0.6518	0.8117	0.7809	**0.8763**	0.8673
Tr31	0.8734	0.8130	0.8335	0.5144	0.7724	0.8652	0.7243	0.8529	0.8304	0.9494	**0.9604**
Tr41	0.8753	**0.9285**	0.9274	0.2391	0.7148	0.8424	0.7243	0.7858	0.6781	0.8990	0.9181
Tr45	0.8570	0.7528	0.7873	0.3219	0.7090	0.8833	0.8651	0.8600	0.8717	0.9187	**0.9359**
Trec7-3000	0.7990	0.6299	0.6030	0.7329	0.8139	0.8746	**0.8769**	0.8489	0.8373	0.8467	0.8732
WAP	0.8323	0.6644	0.6465	0.0720	0.6739	0.7560	0.5791	0.6856	0.6441	**0.7766**	0.7705
WebKB	0.3705	0.2345	0.2464	0.2529	0.2718	0.3734	0.3412	0.2944	0.3495	0.3972	**0.4123**

Table B.8
Micro-F[1] values considering 30 labeled documents for each class.

Collections	MNB	MNB-Se	MNB-Co	EM	TSVM	LLGC	GM	GFHF	TM	IMBHN[R]	TCBHN
Classic4	0.7757	0.2006	0.5396	**0.9151**	0.5515	0.8928	0.7230	0.8811	0.8746	0.8577	0.8883
Dmoz Health-500	0.6714	0.3602	0.1512	0.5889	0.3138	0.6977	0.6406	**0.7049**	0.6560	0.6641	0.6814
Dmoz Science-500	0.5162	0.1684	0.0944	0.4085	0.2740	0.5381	0.5186	**0.5427**	0.5355	0.5016	0.5186
Dmoz Sports-500	0.6847	0.1854	0.0637	0.2985	0.6212	0.7018	0.6496	0.6963	0.6785	0.7172	**0.7499**
FBIS	0.7231	0.6621	0.6684	0.3693	0.5988	0.7662	0.7354	0.7599	0.7451	**0.7785**	0.7583
Hitech	0.6066	0.5830	0.5811	0.2741	0.2306	0.6644	0.5695	**0.6895**	0.5845	0.6587	0.6756
Industry Sector	0.3421	0.2006	0.1871	0.1170	0.2657	0.3894	**0.4135**	0.3712	0.4067	0.3428	0.3973
IrishSentiment	0.5288	**0.5487**	0.5394	0.4645	0.4135	0.5396	0.5384	0.5396	0.5453	0.5234	0.5478
La1s	0.7978	**0.8397**	0.8332	0.4395	0.6571	0.7552	0.7473	0.7404	0.7532	0.7994	0.8231
La2s	0.8096	**0.8375**	0.8312	0.4396	0.6488	0.7644	0.7677	0.7460	0.7766	0.8004	0.8333
Multi Domain Sentiment	0.6012	0.5062	0.5013	0.5353	0.6091	**0.6111**	0.5807	0.6071	0.5894	0.5963	0.6079
Oh0	0.8600	**0.8861**	0.8808	0.8824	0.7424	0.7972	0.8395	0.7775	0.8440	0.8522	0.8603
Oh10	0.7749	0.7916	0.7917	**0.8011**	0.6547	0.7357	0.7560	0.7312	0.7619	0.7700	0.7724
Oh15	0.8015	**0.8215**	0.8123	0.7861	0.6804	0.7728	0.8010	0.7481	0.8113	0.8104	0.8126
Oh5	0.8309	0.8018	0.8107	0.7995	0.7136	0.7794	0.7977	0.7579	0.8002	0.8333	**0.8359**
Ohscal	0.6245	0.4799	0.3325	0.5753	0.5398	0.5969	0.5630	0.5811	0.5783	0.6572	**0.6868**
RCV1 Top-4	**0.6289**	–	–	0.4371	–	–	0.6173	–	0.6269	0.5694	0.6224
Re0	**0.6813**	0.6150	0.5928	0.6119	0.5473	0.6466	0.5520	0.6408	0.4877	0.6055	0.6382
Re8	0.8721	0.8626	0.8699	0.7814	0.6092	0.8694	0.6789	0.8739	0.6978	0.8907	**0.9169**
Review Polarity	0.6475	0.5025	0.5009	0.5052	**0.6635**	0.6066	0.6090	0.5764	0.6331	0.6312	0.6486
Reviews	0.8721	0.8164	0.8187	0.1843	0.5073	0.8926	0.8071	0.8473	0.8311	0.8834	**0.9113**
SpamAssassin	0.8274	0.6574	0.4550	0.6706	0.8303	**0.8980**	0.8302	0.8755	0.8253	0.8419	0.8936
Syskill & Webert	0.9131	0.8921	0.8769	0.1517	0.6145	0.9561	0.8107	**0.9621**	0.8360	0.9500	0.9612
Tr11	0.8580	0.8536	0.8709	0.1517	0.7029	**0.9285**	0.7560	0.8507	0.8256	0.9068	0.9184
Tr31	0.9058	0.8615	0.8756	0.5284	0.8115	0.9044	0.7618	0.8822	0.8627	0.9680	**0.9735**
Tr41	0.9000	0.9314	**0.9337**	0.2444	0.7446	0.8639	0.7618	0.8228	0.6849	0.9198	0.9241
Tr45	0.8435	0.7426	0.7756	0.3352	0.7457	0.9038	0.9005	0.8737	0.8892	0.9380	**0.9505**
Trec7-3000	0.8357	0.6654	0.6390	0.7685	0.8552	0.8882	0.8943	0.8719	0.8733	0.8680	**0.8946**
WAP	**0.8600**	0.7271	0.7230	0.1618	0.7366	0.7924	0.6067	0.6970	0.6573	0.8170	0.8140
WebKB	0.4023	0.2826	0.2871	0.2529	0.3043	0.3933	0.3667	0.3100	0.3703	0.4187	**0.4309**

Table B.9
Micro-F[1] values considering 40 labeled documents for each class.

Collections	MNB	MNB-Se	MNB-Co	EM	TSVM	LLGC	GM	GFHF	TM	IMBHN[R]	TCBHN
Classic4	0.7925	0.2451	0.5381	**0.9153**	0.5521	0.8987	0.7306	0.8895	0.8833	0.8729	0.9000
Dmoz Health-500	0.6975	0.4796	0.3299	0.6126	0.3232	**0.7137**	0.6695	0.7192	0.6834	0.6842	0.6984
Dmoz Science-500	0.5495	0.2547	0.1021	0.4238	0.2953	0.5609	0.5519	0.5618	**0.5645**	0.5331	0.5474
Dmoz Sports-500	0.7093	0.2430	0.0896	0.3131	0.6558	0.7183	0.6583	0.7142	0.7058	0.7301	**0.7617**
FBIS	0.7322	0.6616	0.6839	0.3635	0.6237	0.7776	0.7646	0.7664	0.7562	**0.7855**	0.7438
Hitech	0.6318	0.6191	0.6079	0.2882	0.4213	0.6802	0.5999	**0.6996**	0.6143	0.6775	0.6905
Industry Sector	0.3791	0.2451	0.2382	0.1302	0.2864	0.4186	**0.4489**	0.3983	0.4371	0.3692	0.4274
Irish Sentiment	0.5510	0.5624	0.5633	0.4710	0.4436	0.5613	0.5536	0.5617	0.5642	0.5449	**0.5696**
La1s	0.8136	**0.8471**	0.8416	0.4426	0.6823	0.7695	0.7806	0.7499	0.7882	0.8155	0.8280
La2s	0.8267	**0.8474**	0.8448	0.4434	0.6899	0.7849	0.7879	0.7633	0.7886	0.8265	0.8450
Multi Domain Sentiment	0.6112	0.5069	0.5047	0.5332	0.6184	0.6192	0.5885	0.6160	0.5946	0.6097	**0.6206**
Oh0	0.8740	**0.8995**	0.8958	0.8952	0.7701	0.8088	0.8489	0.7915	0.8516	0.8667	0.8687
Oh10	0.7915	0.8011	0.8044	**0.8117**	0.6825	0.7512	0.7814	0.7417	0.7835	0.7895	0.7920
Oh15	0.8259	**0.8318**	0.8225	0.8019	0.6953	0.7910	0.8189	0.7630	0.8222	0.8253	0.8211
Oh5	0.8359	0.8079	0.8200	0.8069	0.7328	0.7859	0.8014	0.7612	0.7635	0.8417	**0.8533**
Ohscal	0.6476	0.5603	0.4921	0.5822	0.5668	0.6112	0.5921	0.6057	0.6043	0.6779	**0.7082**
RCV1 Top-4	**0.6418**	–	–	0.4468	–	–	0.6229	–	0.6275	0.5842	0.6317
Re0	**0.6835**	0.6028	0.5727	0.6081	0.5734	0.6554	0.5589	0.6646	0.4714	0.5977	0.6036
Re8	0.8859	0.8620	0.8768	0.7881	0.6289	0.8761	0.6935	0.8750	0.7978	0.9009	**0.9273**
Review Polarity	0.6638	0.5092	0.5020	0.5055	0.6794	0.6347	0.6212	0.6071	0.6529	0.6502	**0.6715**
Reviews	0.8805	0.8610	0.8574	0.1837	0.5029	0.8926	0.8207	0.8539	0.8415	0.8973	**0.9193**
SpamAssassin	0.8361	0.5753	0.4516	0.7133	0.8536	0.9042	0.8473	0.8812	0.8403	0.8638	**0.9048**
Syskill & Webert	0.9236	0.9063	0.8960	0.1593	0.6621	0.9632	0.8351	0.9615	0.8489	0.9598	**0.9672**
Tr11	0.8958	0.8982	0.9161	0.1593	0.7677	0.9449	0.8162	0.8737	0.8617	0.9383	**0.9521**
Tr31	0.9187	0.8722	0.8835	0.5433	0.8295	0.9152	0.8546	0.8930	0.8659	0.9693	**0.9764**
Tr41	0.9210	0.9312	0.9395	0.2519	0.8007	0.8855	0.8546	0.8517	0.6865	0.9346	**0.9404**
Tr45	0.8269	0.7424	0.7769	0.3512	0.7702	0.9173	0.9117	0.8792	0.8956	0.9465	**0.9509**
Trec7-3000	0.8559	0.7194	0.6755	0.7769	0.8725	0.9019	**0.9078**	0.8823	0.8977	0.8813	0.9076
WAP	**0.8740**	0.8205	0.8415	0.1992	0.7916	0.8365	0.6327	0.7049	0.6556	0.8482	0.8457
WebKB	0.4165	0.2892	0.2879	0.2829	0.3100	0.4002	0.3622	0.3196	0.3696	0.4187	**0.4496**

Table B.10
Micro-F[1] values considering 50 labeled documents for each class.

Collections	MNB	MNB-Se	MNB-Co	EM	TSVM	LLGC	GM	GFHF	TM	IMBHN[R]	TCBHN
Classic4	0.8157	0.2899	0.5377	**0.9156**	0.6141	0.9029	0.7397	0.8941	0.8907	0.8836	0.9080
Dmoz Health-500	0.7125	0.5574	0.3983	0.6291	0.6355	0.7230	0.6865	**0.7297**	0.7004	0.6944	0.7100
Dmoz Science-500	0.5717	0.3190	0.1212	0.4368	0.4721	0.5736	0.5742	0.5739	**0.5852**	0.5498	0.5666
Dmoz Sports-500	0.7282	0.3101	0.1306	0.3251	0.6821	0.7281	0.6649	0.7247	0.7225	0.7365	**0.7669**
FBIS	0.7426	0.6687	0.6855	0.3606	0.4391	**0.7974**	0.7752	0.7657	0.7594	0.7928	0.7401
Hitech	0.6529	0.6314	0.6377	0.2937	0.4487	0.6941	0.6232	**0.7079**	0.6352	0.6946	0.7043
Industry Sector	0.4157	0.2899	0.2919	0.1460	0.3139	0.4447	**0.4890**	0.4245	0.4750	0.3962	0.4536
IrishSentiment	0.5658	0.5702	0.5700	0.4758	0.5189	0.5720	0.5709	0.5728	0.5776	0.5567	**0.5800**
La1s	0.8258	**0.8518**	0.8464	0.4447	0.6948	0.7792	0.7959	0.7568	0.7987	0.8284	0.8367
La2s	0.8381	0.8514	**0.8542**	0.4474	0.7162	0.7901	0.7978	0.7708	0.8004	0.8378	0.8506
Multi Domain Sentiment	0.6194	0.5033	0.5042	0.5322	0.6300	0.6270	0.5948	0.6240	0.6025	0.6184	**0.6353**
Oh0	0.8751	**0.9050**	0.9013	0.8974	0.7720	0.8165	0.8461	0.7994	0.8497	0.8688	0.8636
Oh10	0.8002	0.8045	0.8015	**0.8124**	0.6902	0.7602	0.7935	0.7456	0.7913	0.7985	0.7998
Oh15	0.8370	**0.8492**	0.8458	0.8199	0.7223	0.8048	0.8344	0.7801	0.8366	0.8465	0.8266
Oh5	0.8457	0.8163	0.8411	0.8187	0.7445	0.7890	0.8053	0.7658	0.8036	0.8514	**0.8634**
Ohscal	0.6606	0.5789	0.5577	0.5825	0.5875	0.6222	0.6093	0.6202	0.6178	0.6942	**0.7195**
RCV1 Top-4	**0.6498**	–	–	0.445	–	–	0.6288	–	0.6339	0.5973	0.6399
Re0	0.6995	0.6002	0.5590	0.6250	0.6098	**0.7516**	0.5934	0.6775	0.4631	0.6287	0.6181
Re8	0.8941	0.8750	0.8816	0.8135	0.5963	0.8828	0.6978	0.8760	0.7105	0.9059	**0.9337**
Review Polarity	0.6821	0.5282	0.5036	0.5061	**0.6951**	0.6434	0.6298	0.6140	0.6633	0.6638	0.6847
Reviews	0.8903	0.8699	0.8665	0.1852	0.4970	0.9005	0.6978	0.8610	0.8517	0.9077	**0.9234**
SpamAssassin	0.8676	0.6480	0.4477	0.7355	0.8498	0.9115	0.8728	0.8871	0.8642	0.8789	**0.9216**
Syskill & Webert	0.9388	0.9239	0.9281	0.1535	0.6896	0.9701	0.8746	0.9590	0.8746	0.9627	**0.9709**
Tr11	0.9110	0.9142	0.9203	0.1535	0.7937	0.9465	0.8370	0.8961	0.9118	0.9606	**0.9614**
Tr31	0.9335	0.8951	0.9024	0.5625	0.8505	0.9214	0.8911	0.9003	0.8807	0.9740	**0.9772**
Tr41	0.9245	0.9304	0.9362	0.2533	0.8205	0.8998	0.8911	0.8675	0.6858	0.9404	**0.9462**
Tr45	0.8127	0.7404	0.7606	0.3727	0.7822	0.9280	0.9262	0.9007	0.9044	**0.9567**	0.9542
Trec7-3000	0.8842	0.7374	0.7024	0.7944	0.8979	0.9055	**0.9164**	0.8886	0.9081	0.8852	0.9076
WAP	0.8751	0.8781	**0.8832**	0.2112	0.8276	0.8541	0.7422	0.7125	0.6550	0.8676	0.8685
WebKB	0.4270	0.3224	0.3154	0.2809	0.3170	0.4073	0.3692	0.3307	0.3779	0.4317	**0.4716**

Table B.11
Macro-F[1] values considering 1 labeled document for each class.

Collections	MNB	MNB-Se	MNB-Co	EM	TSVM	LLGC	GM	GFHF	TM	IMBHN[R]	TCBHN
Classic4	0.5356	0.0796	0.3822	0.8243	0.2998	0.8001	0.5942	**0.8500**	0.6659	0.5377	0.5719
Dmoz Health-500	0.2611	0.0929	0.1335	0.2209	0.1194	0.3642	0.2720	**0.3867**	0.2865	0.2458	0.2659
Dmoz Science-500	0.1913	0.1231	0.1330	0.1571	0.0991	0.2663	0.2308	**0.2669**	0.2424	0.1849	0.2007
Dmoz Sports-500	0.2883	0.0606	0.0737	0.1101	0.1964	0.3958	0.3202	**0.4053**	0.3277	0.2912	0.3225
FBIS	0.4250	0.3055	0.3127	0.3584	0.2532	0.4619	0.3762	**0.4765**	0.4227	0.4075	0.4369
Hitech	0.3334	0.2068	0.2256	0.2029	0.1931	0.3685	**0.4040**	**0.4040**	0.3025	0.3164	0.3490
Industry Sector	0.1305	0.0796	0.0816	0.0873	0.1204	0.1691	0.1708	0.1575	**0.1747**	0.1360	0.1474
Irish Sentiment	0.3839	0.3841	0.4116	0.4015	0.3696	0.4328	0.3859	**0.4450**	0.3978	0.3814	0.3942
La1s	0.4166	0.2356	0.2602	0.3080	0.2816	0.5088	0.3955	**0.5476**	0.4235	0.3616	0.3953
La2s	0.4563	0.2853	0.2754	0.2998	0.3077	0.5625	0.4373	**0.6114**	0.4630	0.3980	0.4361
Multi Domain Sentiment	0.5210	0.5080	0.5085	0.5086	0.5216	**0.5272**	0.5227	0.5242	0.5270	0.5209	0.5224
Oh0	0.4279	0.3301	0.3279	**0.6057**	0.2741	0.5301	0.4310	0.5459	0.4682	0.4005	0.4555
Oh10	0.3776	0.2730	0.2348	**0.4761**	0.2563	0.4412	0.3567	0.4504	0.3748	0.3435	0.3921
Oh15	0.3619	0.2275	0.2299	0.4218	0.2316	0.4188	0.3450	**0.4316**	0.3662	0.3419	0.3822
Oh5	0.4199	0.2621	0.2426	**0.5433**	0.2693	0.5109	0.4043	0.5356	0.4298	0.3838	0.4534
Ohscal	0.3204	0.1636	0.1544	0.3219	0.2186	0.3321	0.2927	**0.3503**	0.3094	0.2887	0.3297
RCV1 Top-4	0.3573	–	–	**0.4043**	–	–	0.3547	–	0.3696	0.3457	0.3613
Re0	0.3445	0.1452	0.1262	0.2128	0.2675	0.4366	0.3653	**0.4508**	0.3839	0.3564	0.4066
Re8	0.4143	0.1909	0.1892	0.2763	0.1784	0.5570	0.4021	**0.6033**	0.4333	0.4847	0.5335
Reviews Polarity	0.5349	0.4803	0.5100	0.4669	0.5020	**0.5573**	0.5434	0.5421	0.5438	0.5333	0.5296
Reviews	0.4885	0.2799	0.3036	0.2779	0.2349	0.5637	0.4512	**0.5737**	0.4782	0.4981	0.5423
SpamAssassin	0.5962	0.5393	0.5034	0.4396	0.5584	0.6366	0.6155	**0.6787**	0.6056	0.6243	0.6225
Syskill & Webert	0.7497	0.6895	0.6103	0.4650	0.4145	**0.9315**	0.5269	0.8656	0.5657	0.7190	0.7854
Tr11	0.4832	0.2746	0.2813	0.1178	0.3283	0.5891	0.4480	**0.6212**	0.4825	0.4818	0.5742
Tr31	0.4678	0.3348	0.2988	0.2655	0.3444	0.5021	0.5032	0.5430	0.4982	0.5151	**0.5600**
Tr41	0.5503	0.3944	0.3520	0.1953	0.3750	0.5948	0.5636	0.6104	0.5764	0.5681	**0.6213**
Tr45	0.5270	0.2637	0.2833	0.2097	0.3084	**0.6006**	0.5212	0.5910	0.5613	0.5295	0.5873
Trec7-3000	0.6525	0.6311	0.6116	0.6258	0.5956	0.7117	0.6791	**0.7375**	0.6993	0.6705	0.6756
WAP	**0.4279**	0.1299	0.1115	0.0110	0.1495	0.4030	0.2864	0.4112	0.3144	0.3213	0.3609
WebKB	0.2488	0.1396	0.1422	0.2507	0.1594	**0.2833**	0.2668	0.2735	0.2778	0.2507	0.2693

Table B.12
Macro-F[1] values considering 10 labeled documents for each class.

Collections	MNB	MNB-Se	MNB-Co	EM	TSVM	LLGC	GM	GFHF	TM	IMBHN[R]	TCBHN
Classic4	0.7873	0.1577	0.6002	**0.9128**	0.4204	0.8855	0.7788	0.8889	0.8523	0.8088	0.8645
Dmoz Health-500	0.5492	0.1620	0.1431	0.5468	0.2573	0.6270	0.5249	**0.6498**	0.5432	0.5439	0.5736
Dmoz Science-500	0.3817	0.1669	0.1530	0.3599	0.2043	0.4427	0.3985	**0.4444**	0.4171	0.3703	0.3954
Dmoz Sports-500	0.5909	0.0952	0.0744	0.2870	0.5315	0.6310	0.6056	0.6266	0.5634	0.6488	**0.6894**
FBIS	0.6343	0.5966	0.5957	0.4973	0.4967	0.6636	0.6338	0.6627	0.6586	0.6727	**0.6800**
Hitech	0.4896	0.4162	0.3837	0.2820	0.1963	0.5461	0.4448	**0.6023**	0.4610	0.5383	0.5742
Industry Sector	0.2788	0.1577	0.1476	0.1369	0.2171	0.3077	0.3650	0.2921	**0.3699**	0.2679	0.3120
Irish Sentiment	0.4846	0.4573	0.4373	0.4773	0.3810	0.4955	0.4834	**0.5236**	0.4880	0.4670	0.4766
La1s	0.6976	0.6132	0.5806	0.4982	0.5255	0.6702	0.6408	0.6744	0.6556	0.6769	**0.7286**
La2s	0.7092	0.6215	0.5948	0.4985	0.5358	0.7015	0.6602	0.7048	0.6726	0.6856	**0.7424**
Multi Domain Sentiment	**0.5685**	0.5167	0.5073	0.5157	0.5582	0.5678	0.5551	0.5665	0.5608	0.5549	0.5580
Oh0	0.7452	**0.8088**	0.7559	0.7978	0.6275	0.7084	0.7444	0.7128	0.8071	0.7432	0.7629
Oh10	0.6748	0.6982	0.6720	**0.7122**	0.5489	0.6548	0.6463	0.6385	0.6556	0.6633	0.6751
Oh15	0.6975	0.7115	0.6745	0.6975	0.5631	0.6852	0.6653	0.6761	0.6869	0.7026	**0.7121**
Oh5	0.7467	0.7526	0.7403	0.7466	0.6079	0.7298	0.7099	0.7303	0.7262	0.7588	**0.7725**
Ohscal	0.5405	0.2575	0.1919	0.5178	0.4362	0.5330	0.4626	0.4818	0.4807	0.5481	**0.5959**
RCV1 Top-4	0.5086	–	0.4812	–	–	–	0.5014	–	0.5174	0.5008	**0.5365**
Re0	0.6031	0.4470	0.4875	0.5629	0.4414	0.6244	0.5528	0.6025	0.5387	**0.6346**	0.6291
Re8	0.7095	0.4703	0.4806	0.5645	0.2201	0.7604	0.5883	0.7676	0.5973	0.7564	**0.8094**
Review Polarity	**0.5933**	0.5120	0.5336	0.5197	0.5848	0.5713	0.5722	0.5681	0.5845	0.5630	0.5700
Reviews	0.7661	0.5056	0.4610	0.3670	0.3235	0.7743	0.7033	0.7800	0.7229	0.7755	**0.8197**
SpamAssassin	0.7572	0.5399	0.5205	0.5714	0.6932	**0.8136**	0.7582	0.8034	0.7746	0.7445	0.7891
Syskill & Webert	0.8754	0.8263	0.7775	0.5734	0.6247	0.9451	0.7176	**0.9551**	0.7326	0.9307	0.9479
Tr11	0.6521	0.6258	0.6315	0.3822	0.5128	0.6545	0.5999	0.6574	0.6385	0.6809	**0.6860**
Tr31	0.6946	0.6406	0.6367	0.3958	0.5771	0.6766	0.6925	0.6756	0.6847	0.7377	**0.7593**
Tr41	0.7439	0.8105	**0.8156**	0.3316	0.6157	0.7297	0.6521	0.7182	0.6472	0.7719	0.7871
Tr45	0.8039	0.7494	0.7615	0.3689	0.6142	0.7775	0.7827	0.7763	0.8195	0.8289	**0.8592**
Trec7-3000	0.8011	0.6933	0.6570	0.7336	0.8109	0.8585	**0.8593**	0.8467	0.8389	0.8190	0.8417
WAP	**0.7452**	0.3900	0.4040	0.0685	0.5215	0.6160	0.5106	0.6050	0.5419	0.6266	0.6354
WebKB	0.4042	0.1819	0.1713	0.4036	0.1789	0.3954	0.3735	0.3348	0.3765	0.4036	**0.4244**

Table B.13
Macro-F^1 values considering 20 labeled documents for each class.

Collections	MNB	MNB-Se	MNB-Co	EM	TSVM	LLGC	GM	GFHF	TM	IMBHN^R	TCBHN
Classic4	0.8132	0.1940	0.6934	0.9252	0.4212	0.8933	0.7925	**0.8920**	0.8770	0.8492	0.8850
Dmoz Health-500	0.6324	0.3021	0.1573	0.6197	0.2956	0.6644	0.5988	**0.6874**	0.6129	0.6302	0.6451
Dmoz Science-500	0.4754	0.2026	0.1601	0.4523	0.2435	0.5073	0.4784	**0.5136**	0.4922	0.4626	0.4782
Dmoz Sports-500	0.6604	0.2053	0.0880	0.3794	0.5927	0.6802	0.6562	0.6735	0.6351	0.7068	**0.7311**
FBIS	0.6717	0.6208	0.6308	0.5315	0.5470	0.6974	0.6904	0.6940	0.6941	**0.7099**	0.7049
Hitech	0.5439	0.5064	0.4921	0.3211	0.2068	0.6019	0.5084	**0.6637**	0.5284	0.5953	0.6250
Industry Sector	0.3597	0.1940	0.1790	0.1482	0.2783	0.3823	**0.4513**	0.3613	0.4458	0.3443	0.3949
IrishSentiment	0.5235	0.5315	0.5144	0.4896	0.4360	0.5296	0.5144	**0.5555**	0.5220	0.5031	0.5231
La1s	0.7540	0.7520	0.7363	0.5226	0.6048	0.7114	0.7048	0.7050	0.7113	0.7385	**0.7747**
La2s	0.7598	0.7648	0.7618	0.5340	0.6069	0.7243	0.7253	0.7190	0.7348	0.7486	**0.7877**
Multi Domain Sentiment	0.5900	0.5196	0.5109	0.5174	0.5831	**0.5930**	0.5719	0.5925	0.5755	0.5842	0.5911
Oh0	0.8043	**0.8350**	0.8247	0.8289	0.6955	0.7636	0.7988	0.7350	0.8071	0.8077	0.8090
Oh10	0.7269	**0.7568**	0.7344	0.7499	0.6162	0.7037	0.7082	0.6793	0.7141	0.7195	0.7211
Oh15	0.7557	**0.7809**	0.7627	0.7390	0.6236	0.7383	0.7481	0.7225	0.7580	0.7665	0.7685
Oh5	0.8032	0.7923	0.7943	0.7847	0.6811	0.7682	0.7862	0.7525	0.7920	**0.8139**	0.8111
Ohscal	0.6007	0.3835	0.3087	0.5541	0.4985	0.5884	0.5279	0.5412	0.5412	0.6231	**0.6509**
RCV1 Top-4	0.5376	–	–	0.5134	–	–	0.5246	–	0.5354	0.5290	**0.5665**
Re0	0.4762	0.4478	0.4466	0.4717	0.3777	**0.4807**	0.4424	0.4545	0.4282	0.4729	0.4686
Re8	0.7574	0.7031	0.6768	0.6320	0.2306	0.7885	0.6150	0.7875	0.6216	0.8107	**0.8422**
Review Polarity	**0.6270**	0.5290	0.5282	0.5188	0.6267	0.6048	0.5969	0.6022	0.6088	0.5991	0.6168
Reviews	0.8193	0.6158	0.6075	0.3721	0.3669	0.8285	0.7616	0.7991	0.7803	0.8128	**0.8509**
SpamAssassin	0.8118	0.5465	0.4873	0.6845	0.7793	0.8488	0.8243	0.8258	0.8345	0.7963	**0.8575**
Syskill & Webert	0.8939	0.8490	0.8227	0.6255	0.6429	0.9501	0.7565	**0.9550**	0.7848	0.9476	0.9546
Tr11	0.4941	0.4925	0.4938	0.2658	0.4110	0.4977	0.4854	0.4961	0.5165	**0.5426**	0.5260
Tr31	0.6285	0.6378	0.6469	0.3967	0.6081	0.6627	0.6922	0.6742	0.6770	0.7529	**0.7632**
Tr41	0.6591	0.6965	0.6961	0.3395	0.5755	0.6453	0.5880	0.6348	0.5798	0.6920	**0.6969**
Tr45	0.6978	0.6515	0.6599	0.2961	0.5992	0.7004	0.6969	0.6973	0.7009	0.7322	**0.7472**
Trec7-3000	0.8243	0.6918	0.6689	0.7357	0.8205	0.8782	**0.8815**	0.8587	0.8511	0.8503	0.8782
WAP	**0.8043**	0.4618	0.4781	0.0852	0.5285	0.5395	0.4555	0.5066	0.4867	0.5572	0.5531
WebKB	0.4360	0.1971	0.1793	0.4500	0.1896	0.4177	0.3998	0.3443	0.4018	0.4500	**0.4645**

Table B.14
Macro-F^1 values considering 30 labeled documents for each class.

Collections	MNB	MNB-Se	MNB-Co	EM	TSVM	LLGC	GM	GFHF	TM	IMBHN^R	TCBHN
Classic4	0.8385	0.2425	0.7089	**0.9250**	0.4122	0.9040	0.8046	0.9022	0.8929	0.8726	0.9038
Dmoz Health-500	0.6741	0.4451	0.2528	0.6446	0.3138	0.7007	0.6393	**0.7084**	0.6537	0.6649	0.6808
Dmoz Science-500	0.5210	0.2591	0.1645	0.4792	0.2740	0.5358	0.5205	**0.5406**	0.5334	0.5028	0.5184
Dmoz Sports-500	0.6980	0.2759	0.1185	0.3997	0.6371	0.7069	0.6802	0.6987	0.6780	0.7340	**0.7504**
FBIS	0.6771	0.6350	0.6376	0.5459	0.5545	0.7070	0.6899	0.6999	0.6941	**0.7103**	0.6950
Hitech	0.5702	0.5350	0.5358	0.3250	0.2234	0.6232	0.5481	**0.6814**	0.5626	0.6228	0.6403
Industry Sector	0.4151	0.2425	0.2160	0.1614	0.3246	0.4313	**0.5142**	0.4121	0.5081	0.3915	0.4435
Irish Sentiment	0.5452	0.5628	0.5621	0.4992	0.4122	0.5501	0.5406	**0.5928**	0.5456	0.5284	0.5511
La1s	0.7792	**0.7994**	0.7919	0.5370	0.6523	0.7421	0.7414	0.7204	0.7456	0.7754	0.7970
La2s	0.7886	0.8051	0.7977	0.5516	0.6518	0.7431	0.7570	0.7254	0.7635	0.7786	**0.8060**
Multi Domain Sentiment	0.6064	0.5168	0.5111	0.5432	**0.6130**	0.6118	0.5892	0.6100	0.5941	0.5980	0.6106
Oh0	0.8247	**0.8481**	0.8433	0.8417	0.7166	0.7821	0.8174	0.7662	0.8221	0.8274	0.8333
Oh10	0.7386	0.7484	0.7374	**0.7615**	0.6440	0.7068	0.7260	0.6877	0.7322	0.7329	0.7342
Oh15	0.7708	0.7877	0.7751	0.7532	0.6514	0.7496	0.7802	0.7334	**0.7889**	0.7849	0.7858
Oh5	0.8281	0.8107	0.8144	0.7994	0.7124	0.7835	0.8112	0.7623	0.8156	0.8339	**0.8355**
Ohscal	0.6263	0.5009	0.4017	0.5750	0.5383	0.6139	0.5625	0.5752	0.5757	0.6591	**0.6819**
RCV1 Top-4	0.5511	–	–	0.5174	–	–	0.5398	–	0.5473	0.5378	**0.5754**
Re0	0.4591	0.4581	0.4562	**0.4678**	0.3662	0.4520	0.4261	0.4386	0.4242	0.4493	0.4336
Re8	0.7650	0.7585	0.7743	0.6767	0.2489	0.7936	0.6144	0.7897	0.6278	0.8224	**0.8474**
Review Polarity	**0.6561**	0.5453	0.5424	0.5367	0.6663	0.6138	0.6219	0.6011	0.6346	0.6328	0.6494
Reviews	0.8336	0.6931	0.6991	0.3943	0.3888	0.8500	0.7857	0.8052	0.8041	0.8377	**0.8624**
SpamAssassin	0.8268	0.6044	0.4976	0.6976	0.8180	0.8634	0.8356	0.8353	0.8333	0.8128	**0.8689**
Syskill & Webert	0.8889	0.8605	0.8348	0.6297	0.6508	0.9503	0.7806	**0.9567**	0.7973	0.9435	0.9563
Tr11	0.3805	0.3787	0.3844	0.1258	0.3228	**0.4100**	0.3533	0.3779	0.3642	0.4029	0.4063
Tr31	0.6285	0.5756	0.5986	0.3379	0.5565	0.6068	0.6447	0.5919	0.6162	0.6745	**0.6827**
Tr41	0.5922	**0.6248**	0.6216	0.2362	0.5019	0.5816	0.5245	0.5830	0.5141	0.6138	0.6012
Tr45	0.6846	0.6475	0.6567	0.2976	0.6044	0.7024	0.7056	0.7011	0.7017	0.7354	**0.7479**
Trec7-3000	0.8523	0.7213	0.6987	0.7729	0.8583	0.8913	**0.8995**	0.8796	0.8824	0.8703	0.8970
WAP	**0.8247**	0.4987	0.5067	0.1579	0.5506	0.5416	0.4572	0.5064	0.4905	0.5575	0.5556
WebKB	0.4524	0.2163	0.1904	0.4762	0.1871	0.4375	0.4155	0.3642	0.4127	0.4762	**0.4871**

Table B.15
Macro-F[1] values considering 40 labeled documents for each class.

Collections	MNB	MNB-Se	MNB-Co	EM	TSVM	LLGC	GM	GFHF	TM	IMBHN[R]	TCBHN
Classic4	0.8503	0.2871	0.7209	**0.9249**	0.4126	0.9082	0.8109	0.9079	0.9010	0.8859	0.9115
Dmoz Health-500	0.7001	0.5422	0.4538	0.6641	0.3232	0.7171	0.6684	**0.7223**	0.6814	0.6860	0.6982
Dmoz Science-500	0.5536	0.3573	0.1765	0.4942	0.2953	0.5591	0.5536	0.5596	**0.5630**	0.5338	0.5470
Dmoz Sports-500	0.7210	0.3531	0.1626	0.4220	0.6681	0.7193	0.6915	0.7163	0.7060	0.7468	**0.7624**
FBIS	0.6416	0.6112	0.6193	0.5250	0.5358	0.6544	0.6594	0.6526	0.6579	**0.6695**	0.6349
Hitech	0.5903	0.5690	0.5599	0.3414	0.4462	0.6415	0.5746	**0.6900**	0.5859	0.6431	0.6563
Industry Sector	0.4472	0.2871	0.2738	0.1718	0.3569	0.4608	**0.5496**	0.4425	0.5407	0.4189	0.4751
IrishSentiment	0.5650	0.5710	0.5756	0.5050	0.4455	0.5701	0.5591	**0.6130**	0.5657	0.5449	0.5701
La1s	0.7911	**0.8099**	0.8030	0.5569	0.6759	0.7566	0.7654	0.7283	0.7709	0.7926	0.8044
La2s	0.8031	0.8159	0.8122	0.5579	0.6811	0.7626	0.7756	0.7408	0.7769	0.8020	**0.8172**
Multi Domain Sentiment	0.6145	0.5089	0.5102	0.5393	0.6211	0.6199	0.5984	0.6182	0.5995	0.6105	**0.6219**
Oh0	0.8255	**0.8516**	0.8460	0.8486	0.7291	0.7717	0.8192	0.7620	0.8212	0.8255	0.8261
Oh10	0.7452	0.7502	0.7411	**0.7651**	0.6612	0.7112	0.7377	0.6900	0.7401	0.7428	0.7448
Oh15	0.7802	0.7812	0.7680	0.7480	0.6522	0.7530	0.7839	0.7315	**0.7846**	0.7833	0.7794
Oh5	0.8283	0.8101	0.8200	0.8036	0.7184	0.8112	0.8130	0.7609	0.7584	0.8382	**0.8444**
Ohscal	0.6458	0.5514	0.5131	0.5797	0.5635	0.6277	0.5897	0.5993	0.6004	0.6795	**0.7024**
RCV1 Top-4	0.5592	–	–	0.5081	–	–	0.5468	–	0.5525	0.5487	**0.5834**
Re0	**0.3231**	0.3085	0.3075	0.3163	0.2691	0.3153	0.2906	0.2972	0.2818	0.3024	0.3076
Re8	0.7668	0.7793	0.8045	0.6953	0.2579	0.7984	0.6156	0.7868	0.6696	0.8217	**0.8446**
Review Polarity	0.6719	0.5500	0.5466	0.5313	**0.6813**	0.6449	0.6361	0.6317	0.6552	0.6510	0.6723
Reviews	0.8360	0.7752	0.7747	0.3874	0.3989	0.8500	0.7972	0.8114	0.8111	0.8520	**0.8697**
SpamAssassin	0.8332	0.5803	0.5016	0.7318	0.8375	0.8718	0.8482	0.8434	0.8460	0.8381	**0.8823**
Syskill & Webert	0.8923	0.8681	0.8487	0.6325	0.6844	0.9548	0.7920	0.9539	0.7986	0.9513	**0.9586**
Tr11	0.3885	0.3886	0.3944	0.1377	0.3357	0.4156	0.3747	0.3827	0.3697	0.4097	**0.4172**
Tr31	0.6314	0.5880	0.6045	0.3386	0.5538	0.6097	0.6468	0.5919	0.6159	0.6699	**0.6802**
Tr41	0.4665	0.4683	0.4719	0.1919	0.4122	0.4537	0.4444	0.4427	0.4089	0.4691	**0.4762**
Tr45	0.5926	0.5698	0.5759	0.2873	0.5295	0.6206	0.6261	0.6286	0.6293	0.6513	**0.6553**
Trec7-3000	0.8694	0.7641	0.7265	0.7818	0.8755	0.9046	**0.9114**	0.8892	0.9034	0.8836	0.9102
WAP	**0.8255**	0.4292	0.4346	0.1618	0.4180	0.4281	0.3760	0.3890	0.3855	0.4308	0.4282
WebKB	0.4650	0.2462	0.2230	0.4762	0.1971	0.4469	0.4267	0.3766	0.4225	0.4762	**0.5022**

Table B.16
Macro-F[1] values considering 50 labeled documents for each class.

Collections	MNB	MNB-Se	MNB-Co	EM	TSVM	LLGC	GM	GFHF	TM	IMBHN[R]	TCBHN
Classic4	0.8651	0.3259	0.7344	**0.9250**	0.4957	0.9115	0.8173	0.9106	0.9078	0.8948	0.9174
Dmoz Health-500	0.7148	0.6163	0.5191	0.6752	0.6512	0.7260	0.6852	**0.7321**	0.6985	0.6966	0.7095
Dmoz Science-500	0.5755	0.4188	0.2080	0.5056	0.4839	0.5720	0.5763	0.5717	**0.5844**	0.5506	0.5660
Dmoz Sports-500	0.7381	0.4320	0.2297	0.4379	0.6922	0.7296	0.6984	0.7265	0.7229	0.7532	**0.7673**
FBIS	0.4951	0.4808	0.4792	0.3576	0.4391	0.5103	0.4955	0.5018	0.5010	**0.5120**	0.4939
Hitech	0.6116	0.5693	0.5808	0.3503	0.4687	0.6575	0.5936	**0.6975**	0.6017	0.6609	0.6696
Industry Sector	0.4732	0.3259	0.3249	0.1838	0.3884	0.4866	**0.5750**	0.4683	0.5690	0.4438	0.4975
IrishSentiment	0.5778	0.5758	0.5793	0.5068	0.5253	0.5768	0.5728	**0.6247**	0.5761	0.5540	0.5779
La1s	0.8015	**0.8148**	0.8068	0.5627	0.6868	0.7653	0.7779	0.7344	0.7798	0.8047	0.8134
La2s	0.8144	0.8204	0.8208	0.5687	0.7004	0.7679	0.7880	0.7476	0.7879	0.8134	**0.8250**
Multi Domain Sentiment	0.6226	0.5041	0.5045	0.5363	0.6321	0.6278	0.6042	0.6271	0.6073	0.6190	**0.6366**
Oh0	0.7825	**0.8184**	0.8107	0.8117	0.6857	0.7362	0.7838	0.7243	0.7856	0.7815	0.7659
Oh10	0.7383	0.7316	0.7258	0.7531	0.6525	0.6983	0.7372	0.6808	0.7386	0.7365	**0.7429**
Oh15	0.7433	0.7551	0.7451	0.7396	0.6489	0.7236	0.7611	0.7038	0.7565	**0.7626**	0.7461
Oh5	0.8251	0.8061	0.8333	0.8016	0.7086	0.7746	0.8080	0.7572	0.8075	0.8392	**0.8420**
Ohscal	0.6565	0.5696	0.5568	0.5838	0.5827	0.6375	0.6064	0.6142	0.6141	0.6934	**0.7126**
RCV1 Top-4	0.5662	–	–	0.5140	–	–	0.5525	–	0.5575	0.5599	**0.5919**
Re0	**0.2834**	0.2779	0.2721	0.2739	0.2315	0.2778	0.2520	0.2572	0.2365	0.2624	0.2652
Re8	0.7554	0.7685	0.7866	0.7009	0.2690	0.7546	0.6134	0.7539	0.6267	0.7991	**0.8146**
Review Polarity	0.6886	0.5636	0.5492	0.5623	**0.6973**	0.6530	0.6466	0.6417	0.6661	0.6646	0.6861
Reviews	0.8439	0.8251	0.8168	0.3926	0.3922	0.8553	0.6134	0.8117	0.8203	0.8607	**0.8717**
SpamAssassin	0.8583	0.6279	0.5085	0.7466	0.8378	0.8816	0.8677	0.8503	0.8645	0.8539	**0.9004**
Syskill & Webert	0.8995	0.8734	0.8654	0.6364	0.7043	0.9563	0.8059	0.9444	0.8076	0.9476	**0.9567**
Tr11	0.3670	0.3667	0.3663	0.1370	0.3181	0.4008	0.3624	0.3613	0.3443	0.4125	**0.4126**
Tr31	0.6308	0.5853	0.6037	0.3411	0.5574	0.6065	0.6474	0.5951	0.6194	0.6644	**0.6704**
Tr41	0.4649	0.4662	0.4682	0.1935	0.4140	0.4548	0.4505	0.4447	0.4091	0.4697	**0.4749**
Tr45	0.4964	0.4813	0.4826	0.2476	0.4572	0.5444	0.5449	0.5335	0.5381	**0.5612**	0.5593
Trec7-3000	0.8917	0.7776	0.7459	0.8012	0.8992	0.9082	**0.9201**	0.8943	0.9131	0.8871	0.9096
WAP	**0.7825**	0.4240	0.4230	0.1692	0.4033	0.4120	0.3778	0.3791	0.3754	0.4164	0.4148
WebKB	0.4715	0.2757	0.2580	0.4876	0.2001	0.4511	0.4349	0.3851	0.4300	0.4876	**0.5138**

Appendix B. Best *Micro-F*^{1} and *Macro-F*^{1} values

In this appendix we present the numerical values of *Micro-F*^{1} and *Macro-F*^{1} which were used to generate the charts in Figs. 2–5. Tables B.5, B.6, B.7, B.8, B.9, and B.10 present *Micro-F*^{1} values for 1, 10, 20, 30, 40, and 50 labeled documents per class respectively, and Tables B.11, B.12, B.13, B.14, B.15, and B.16 present *Macro-F*^{1} values for 1, 10, 20, 30, 40, and 50 labeled documents per class respectively. The highest *Micro-F*^{1} and *Macro-F*^{1} values for each text collection are highlighted in bold.

GFHF presented the highest *Micro-F*^{1} and *Macro-F*^{1} for most of the text collections considering 1 labeled example for both evaluation measures. Nevertheless, using just 1 labeled example for each class provides a much lower *Micro-F*^{1} and *Macro-F*^{1} values than 10 or more labeled documents per class.

TCBHN obtains the highest *Micro-F*^{1} and *Macro-F*^{1} values for more text collection than other algorithms considering 10 or more labeled documents per class. We highlight that TCBHN make a better use of unlabeled documents than other algorithms. TCBHN was better than its supervised version (IMBHN^R) in about 27 of 30 collections for both evaluation measures. The lesser the number of labeled documents, the higher the average difference in *Micro-F*1 and *Macro-F*^{1} values between TCBHN and IMBHN. TCBHN presents 0.04 higher *Micro-F*^{1} and *Macro-F*^{1} values than IMBHN in average using one labeled documents. The difference is 0.01 in average for *Micro-F*^{1} and 0.017 for *Macro-F*^{1}.

On the other hand, MNB-Se, MNB-Co and EM obtained worst *Micro-F*^{1} and *Macro-F*^{1} than their supervised version (MNB) for most of the text collections. EM was better than MNB in few collections, mainly when using 1 labeled document for each class. EM presented significantly improvements in classification performance, such as 0.28 in Classic4 and 0.17 in Oh-0 for *Macro-F*^{1} and 1 labeled document per class. However, EM also presented significantly decreases in classification performance, such as 0.68 in Reviews and 0.67 in Tr11 for *Micro-F*1 and 20 labeled documents per class.

Self-Training and Co-Training obtained high decreases than increases in classification performance. For instance, Self-Training surpass MNB in 0.06 for Oh and 0.08 for Tr41 considering *Micro-F*^{1} and 10 labeled documents per class, while Co-Training surpasses MNB in 0.09 for Tr41 considering *Micro-F*^{1} and 10 labeled documents, and 0.07 for Tr41 considering *Macro-F*^{1} and 10 labeled documents per class. On the other hand, Self-Training obtained a lower classification performance than MNB in 0.62 for Classic4 considering *Macro-F*^{1} and 10 labeled document, while Co-Training obtained a lower classification performance than MNB in 0.21 for Tr45 considering *Macro-F*^{1} and 1 labeled document per class.

References

Aggarwal, C. C., & Zhao, P. (2013). Towards graphical models for text processing. *Knowledge & Information Systems, 36*(1), 1–21.

Angelova, R., & Weikum, G. (2006). Graph-based text classification: Learn from your neighbors. *Proceedings of the special interest group on information retrieval conference* (pp. 485–492). ACM.

Apache (2006). The apache spamassassin project. <http://spamassassin.apache.org/publiccorpus/> Last accessed 6.11.13.

Belkin, M., Niyogi, P., & Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research, 7*, 2399–2434.

Berry, M. W., & Castellanos, M. (2008). *Survey of text mining II: Clustering, classification, and retrieval* (1st ed.). Springer.

Bickel, S., & Scheffer, T. (2004). Multi-view clustering. *Proceedings of the international conference on data mining* (pp. 19–26). IEEE Computer Society.

Blitzer, J., Dredze, M., & Pereira, F. (2009). Multi-domain sentiment dataset (version 2.0). <http://www.cs.jhu.edu//mdredze/datasets/sentiment/> Last accessed 6.11.13.

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

Breve, F. A., Zhao, L., Quiles, M. G., Pedrycz, W., & Liu, J. (2012). Particle competition and cooperation in networks for semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering, 24*(9), 1686–1698.

Castillo, C., Donato, D., Gionis, A., Murdock, V., & Silvestri, F. (2007). Know your neighbors: Web spam detection using the web topology. *Proceedings international conference on research and development in information retrieva*. ACM.

Chapelle, O., Schölkopf, B., & Zien, A. (Eds.). (2006). *Semi-supervised learning*. MIT Press.

Cormack, G. V., & Lynam, T. R. (2007). TREC public spam corpus. <http://plg.uwaterloo.ca/gvcormac/treccorpus07/> Last accessed 6.11.13.

Culp, M., & Michailidis, G. (2008). An iterative algorithm for extending learners to a semi-supervised setting. *Journal of Computational & Graphical Statistics, 17*, 545–571.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B, 39*, 1–38.

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*, 1–30.

de Sousa, C. A. R., Rezende, S. O., & Batista, G. E. A. P. A. (2013). Influence of graph construction on semi-supervised learning. In *Proceedings of the European conference machine learning and knowledge discovery in databases* (pp. 160–175).

Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. *Proceedings of the international conference on knowledge discovery and data mining* (pp. 269–274). ACM.

D.M. Research, (2010). Classic3 and classic4 datasets. <http://www.dataminingresearch.com/index.php/2010/09/classic3-classic4-datasets/> Last accessed 19.07.13.

Ferreira, R., de Freitas, F. L. G., de Souza Cabral, L., Lins, R. D., Lima, R., de Franca Pereira e Silva, G., et al. (2013). A four dimension graph model for automatic text summarization. *Proceedings of the web intelligence congress* (pp. 389–396). IEEE.

Forman, G. (2006). 19MclassTextWc dataset. <http://sourceforge.net/projects/weka/files/datasets/text-datasets/19MclassTextWc.zip/download>.

García, S., Fernández, A., Luengo, J., & Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences, 180*(10), 2044–2064.

Ghani, R. (2002). Combining labeled and unlabeled data for multiclass text categorization. *Proceedings of the international conference on machine learning* (pp. 187–194). Morgan Kaufmann Publishers Inc..

M.L. Group, (2009). Irish economic sentiment dataset. <http://mlg.ucd.ie/sentiment> Last accessed 12.07.12.

C.T.L., Group (1998). The 4 universities data set. <http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/> Last accessed 6.11.13.

Haffari, G., & Sarkar, A. (2007). Analysis of semi-supervised learning with the yarowsky algorithm. In *Conference on uncertainty in artificial intelligence, association for uncertainty in artificial intelligence* (pp. 159–166).

Han, E.-H., Boley, D., Gini, M., Gross, R., Hastings, K., Karypis, G., et al. (1998). Webace: A web agent for document categorization and exploration. *Proceedings of the international conference on autonomous agents* (pp. 408–415). ACM.

Hersh, W., Buckley, C., Leone, T. J., & Hickam, D. (1994). OHSUMED: An interactive retrieval evaluation and new large test collection for research. *Proceedings of the annual international conference on research and development in information retrieval* (pp. 192–201). Springer-Verlag New York, Inc..
He, Y., & Zhou, D. (2011). Self-training from labeled features for sentiment analysis. *Information Processing & Management, 47*(4), 606–616.
Jebara, T., Wang, J., & Chang, S.-F. (2009). Graph construction and b-matching for semi-supervised learning. *Proceedings of the international conference on machine learning* (pp. 441–448). ACM.
Ji, M., Sun, Y., Danilevsky, M., Han, J., & Gao, J. (2010). Graph regularized transductive classification on heterogeneous information networks. *Proceedings of the European conference on machine learning and knowledge discovery in databases* (pp. 570–586). Springer.
Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *Proceedings of international conference on machine learning* (pp. 200–209).
Karypis, G. (2006). Cluto – software for clustering high-dimensional datasets. <http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download> Last accessed 6.11.13.
Kim, S.-M., Pantel, P., Duan, L., & Gaffney, S. (2009). Improving web page classification by label-propagation over click graphs. *Proceedings of the international conference on information and knowledge management* (pp. 1077–1086). ACM.
Kong, X., Ng, M. K., & Zhou, Z.-H. (2013). Transductive multilabel learning via label set propagation. *IEEE Transactions on Knowledge and Data Engineering, 25*(3), 704–719.
Laguna, V., & de Andrade Lopes, A. (2010). Combining local and global KNN with cotraining, *19th European conference on artificial intelligence: Vol. 215. ECAI 2010* (pp. 815–820). Netherlands: IOS Press. doi:10.3233/978-1-60750-606-5-815.
Lewis, D. D. (2004). Reuters-21578. <http://www.daviddlewis.com/resources/testcollections/reuters21578/> Last accessed 6.11.13.
Lewis, D.D. (2005). Rcv1-v2/lyrl2004: The lyrl2004 distribution of the rcv1-v2 text categorization test collection. Last accessed 6.11.14.
Li, K., Meng, X., Cao, Z., & Sun, X. (2009). Multi-view learning for high dimensional data classification. *Proceedings of the international conference on Chinese control and decision conference* (pp. 3809–3813). IEEE Press.
Li, T., Zhu, S., & Ogihara, M. (2008). Text categorization via generalized discriminant analysis. *Information Processing & Management, 44*(5), 1684–1697.
Manning, C., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.
Markov, A., & Last, M. (2006). Model-based classification of web documents represented by graphs. *Proceedings of the workshop on web mining and web usage analysis on special interest group on knowledge discovery and data mining conference* (pp. 84–89). ACM.
Matsuo, Y., Sakaki, T., Uchiyama, K., & Ishizuka, M. (2006). Graph-based word clustering using a web search engine. *Prof. conference on empirical methods in natural language processing* (pp. 542–550). ACL.
Mei, Q., Cai, D., Zhang, D., & Zhai, C. (2008). Topic modeling with network regularization. *Proceedings of the international conference on world wide web* (pp. 101–110). ACM.
Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 404–411).
Nedjah, N., Mourelle, L. de Macedo, Kacprzyk, J., Frana, F. M. G., & de Souza, A. F. (2008). *Intelligent text categorization and clustering* (1st ed.). Springer.
Netscape. Dmoz – Open directory project. <http://www.dmoz.org/> Last accessed 19.07.13.
Nigam, K. (2000). The industry sector dataset. <http://www.cs.cmu.edu/TextLearning/sector-data.tar.gz> Last accessed 12.07.12.
Nigam, K., & Ghani, R. (2000). Analyzing the effectiveness and applicability of co-training. *Proceedings of the conference on information and knowledge management* (pp. 86–93). ACM.
Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning, 39*(2/3), 103–134.
Oh, H., Myaeng, S., & Lee, M. (2000). A practical hypertext categorization method using links and incrementally available class information. *Proceedings of the special interest group on information retrieval conference* (pp. 264–271). ACM.
Page, L., Brin, S., Motwani, R., & Winograd, T., (November 1998). *The PageRank citation ranking: Bringing order to the web*. Tech. rep. 1999-66, Stanford University. <http://ilpubs.stanford.edu:8090/422/>.
Palshikar, G. K. (2007). Keyword extraction from a single document using centrality measures. *Proceedings of the international conference on pattern recognition and machine intelligence* (pp. 503–510). Springer.
Pang, S. (2010). Csmining group – The r8 of reuters 21578 data set. <http://csmining.org/index.php/r52-and-r8-of-reuters-21578.html> Last accessed 10.03.14.
Pang, B., & Lee, L. (2004). Movie review data. http://www.cs.cornell.edu/People/pabo/movie-review-data/> Last accessed 6.11.13.
Pazzani, M. (1998). Syskill and webert web page ratings data set. <http://archive.ics.uci.edu/ml/datasets/Syskill+and+Webert+Web+Page+Ratings> Last accessed 6.11.13.
Porter, M. F. (1980). An algorithm for suffix stripping. *Readings in Information Retrieval, 14*(3), 130–137.
Rossi, R. G., Marcacini, R. M., & Rezende, S. O. (2013). *Benchmarking text collections for classification and clustering tasks*. Tech. rep. 395, Institute of Mathematics and Computer Sciences – University of Sao Paulo. <http://www.icmc.usp.br/CMS/Arquivos/arquivos_enviados/BIBLIOTECA_113_RT_395.pdf>.
Rossi, R. G., Faleiros, T. P., Lopes, A. A., & Rezende, S. O. (2012). Inductive model generation for text categorization using a bipartite heterogeneous network. *Proceedings of the international conference on data mining* (pp. 1086–1091). IEEE.
Rossi, R. G., Lopes, A. A., Faleiros, T. P., & Rezende, S. O. (2014). Inductive model generation for text classification using a bipartite heterogeneous network. *Journal of Computer Science and Technology, 3*(29), 361–375.
Rossi, R. G., Lopes, A. A., & Rezende, S. O. (2014). A parameter-free label propagation algorithm using bipartite heterogeneous networks for text classification. *Proceedings of the symposium on applied computing* (pp. 79–84). ACM.
Salton, G. (1989). *Automatic text processing: The transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc..
Salton, G., Singhal, A., Mitra, M., & Buckley, C. (1997). Automatic text structuring and summarization. *Information Processing & Management, 33*(2), 193–207.
Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys, 34*(1), 1–47.
Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management, 45*(4), 427–437.
Solé, R. V., Corominas-Murtra, B., Valverde, S., & Steels, L. (2010). Language networks: Their structure, function, and evolution. *Complexity, 15*(6), 20–26.
Steyvers, M., & Tenenbaum, J. B. (2005). The large-scale structure of semantic networks: Statistical analyses and a model of semantic growth. *Cognitive Science, 29*, 41–78.
Subramanya, A., & Bilmes, J. (2008). Soft-supervised learning for text classification. In *Proceedings of the conference on empirical methods in natural language processing, association for computational linguistics* (pp. 1090–1099).
Sun, Y., & Han, J. (2012). *Mining heterogeneous information networks: Principles and methodologies*. Morgan & Claypool Publishers.
Sun, Y., Han, J., Gao, J., & Yu, Y. (2009). iTopicModel: Information network-integrated topic modeling. *Proceedings of the international conference on data mining* (pp. 493–502). IEEE Computer Society.
Tan, P.-N., Steinbach, M., & Kumar, V. (2005). *Introduction to data mining*. Addison-Wesley.
Trawinski, B., Smetek, M., Telec, Z., & Lasota, T. (2012). Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms. *Applied Mathematics and Computer Science, 22*(4), 867–881.
TREC (2013). Text REtrieval Conference data. <http://trec.nist.gov/data.html> Last accessed 6.11.13.
Tseng, Y.-H., Ho, Z.-P., Yang, K.-S., & Chen, C.-C. (2012). Mining term networks from text collections for crime investigation. *Expert Systems with Applications, 39*(11), 10082–10090.
Uysal, A. K., & Gunal, S. (2014). The impact of preprocessing on text classification. *Information Processing & Management, 50*(1), 104–112.

Valin, J.-M., & Collings, I. B. (2007). Interference-normalized least mean square algorithm. *IEEE Signal Processing Letters, 14*(12), 988–991.
Vapnik, V. N. (1998). *Statistical learning theory* (1st ed.). Wiley.
Wang, W., Do, D. B., & Lin, X. (2005). Term graph model for text classification. *Proceedings of the international conference on advanced data mining and applications* (pp. 19–30). Springer.
Wang, F., & Zhang, C. (2006). Label propagation through linear neighborhoods. *Proceedings of the international conference on machine learning* (pp. 985–992). ACM.
Wan, X., Yang, J., & Xiao, J. (2007). Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. *Proc. annual meeting of the association for computational linguistics* (pp. 552–559). ACM.
Weiss, S. M., Indurkhya, N., & Zhang, T. (2012). *Fundamentals of predictive text mining*. Springer.
Widrow, B., & Hoff, M. E. (1960). Adaptive switching circuits. In *Institute of radio engineers, western electronic show and convention, convention record* (Part 4, pp. 96–104).
Xu, C., Tao, D., & Xu, C. (2013). A survey on multi-view learning. *Computing Research Repository*, 7 Available from abs/1304.5634 .
Yang, S.-Y., & Soo, V.-W. (2012). Extract conceptual graphs from plain texts in patent claims. *Engineering Applications of Artificial Intelligence, 25*(4), 874–887.
Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the annual meeting on association for computational linguistics, association for computational linguistics* (pp. 189–196).
Yin, Z., Li, R., Mei, Q., & Han, J. (2009). Exploring social tagging graph for web object classification. In *Proceedings of the international conference on knowledge discovery and data mining* (pp. 957–966).
Zhou, D., Bousquet, O., Lal, T. N., Weston, J., & Schölkopf, B. (2004). Learning with local and global consistency. In *Proceedings of the advances in neural information processing systems* (Vol. 16, pp. 321–328).
Zhu, X., Ghahramani, Z., & Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. *Proceedings of the international conference on machine learning* (pp. 912–919). AAAI Press.
Zhu, X., & Goldberg, A. B. (2009). *Introduction to semi-supervised learning*. Morgan and Claypool Publishers.