2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

Graph-Based Term Weighting for Text Categorization

Fragkiskos D. Malliaros*
École Polytechnique
France
Email: fmalliaros@lix.polytechnique.fr

Konstantinos Skianis*
École Polytechnique and ENS Cachan
France
Email: kskianis@lix.polytechnique.fr

***Abstract*—Text categorization is an important task with plenty of applications, ranging from sentiment analysis to automated news classification. In this paper, we introduce a novel graph-based approach for text categorization. Contrary to the traditional Bag-of-Words model for document representation, we consider a model in which each document is represented by a graph that encodes relationships between the different terms. The importance of a term to a document is indicated using graph-theoretic node centrality criteria. The proposed weighting scheme is able to meaningfully capture the relationships between the terms that co-occur in a document, creating feature vectors that can improve the categorization task. We perform experiments in well-known document collections, applying popular classification algorithms. Our preliminary results indicate that the proposed graph-based weighting mechanism is able to outperform existing frequency-based term weighting criteria, under appropriate parameter setting.**

I. Introduction

With the rapid growth of the social media and networking platforms, the available textual resources have been increased. Being able to automatically analyze and extract useful information from textual data is an important task with many applications. *Text categorization* or classification (TC) refers to the supervised learning task of assigning a document to a set of two or more pre-defined categories (or classes), based on learning models that have been trained using labeled data (i.e., documents with known class information) [35].

TC can be applied in several domains. A well-known application is the one of *opinion mining* (also known as sentiment analysis). The goal there is to use TC tools in order to identify subjective information (i.e., positive or negative opinions) from text corpora [8]. Opinion mining can be useful in several ways. For example, it can help a company to evaluate the success of an ad campaign or of a new product, leading to better risk assessment and management. Another very well-known application of TC is email filtering and more precisely, spam detection [2].

The pipeline that is followed in the TC problem is similar to any other supervised learning task. Most of the existing methods transform the textual data into a tabular representation that can later be used for the classification task. Typically, each document is modeled using the so-called *Vector Space Model* [4]. That way, a document is represented as a vector in the space defined by the different terms, and if a term occurs in the document, the corresponding value in the vector is non-zero. The main issue here is how to weight the importance of each term in the document. Typically, the *Bag-of-Words* (BoW) model is applied [4]; a text is represented as a multiset of its terms, disregarding the ordering of the words, but only keeping information about the frequency of appearance. Based on this modeling, two well-known and widely used term weighting schemes has been introduced, namely *Term Frequency* (TF) and *Term Frequency - Inverse Document Frequency* (TF-IDF) [35]. Although several variants and extensions of this modeling approach have been proposed (e.g., the n-gram model [4]), the main weakness comes from the underlying term independence assumption. The order of the terms within a document is completely disregarded and any relationship between terms is not taken into account in the categorization task.

In this paper, we introduce term weighting criteria for TC that go beyond the term independence assumption and consider relationships between the terms of a document. The notion of dependencies is introduced based on a graph-based document representation model. In this model, each term is represented as a node in the graph and the edges capture co-occurrence relationships of terms within a specified distance in the document. The importance of a term can be determined by the importance of the corresponding node in the graph, using node centrality criteria. By utilizing term ordering information in the features extracted by the document, the accuracy of categorization task can be improved.

The main contributions of this paper can be summarized as follows:

- *Graph-based term weighting schemes*: we adopt a graph-based representation of documents and derive novel term weighting schemes, by considering local and global node centrality criteria.
- *Exploration of model's parameter space and experimental evaluation*: since graphs are rich modeling structures, we discuss the main parameters of the graph representation approach for the TC problem. Then, we perform a preliminary experimental evaluation on well-known document collections, examining the performance of the different proposed weighting criteria.
- *Future research directions*: although the concept of graph-based document representation has already been used in other domains (such as, Information Retrieval), many features of this modeling approach are still unexplored. Here, we discuss how such problems can be addressed in order to further enhance the performance of graph-based term weighting schemes for TC.

The rest of the paper is organized as follows. Section II reviews part of the related work on text categorization

*Equal contribution.

ASONAM '15, August 25-28, 2015, Paris, France
 ISBN 978-1-4503-3854-7/15/08 $15.00
DOI: http://dx.doi.org/10.1145/2808797.2808872

techniques as well as graph-based models in text mining and Information Retrieval. Section III presents preliminary concepts for the task of TC. Then, in Sec. IV we introduce our approach and the proposed graph-based term weighting criteria for TC. Section V presents the experimental results and finally, in Sec. VI we provide further directions to the TC problem using the graph representation of documents.

II. Related Work

In this section, we briefly review the related work on TC, graph-based TC methods and graph-based techniques in text mining and IR.

a) Text classification: TC is one of the most fundamental and well studied tasks in text analytics and a number of diverse approaches have been proposed [21], [35], [14], [15], [22], [37], [26], [16], [19]. The first step of TC concerns the feature extraction task, i.e., which features will be used to represent the textual content. Typically, the straightforward *Bag-of-Words* approach is adopted, where every document is represented by a feature vector that contains boolean or weighted representation of unigrams or n-grams in general [11]. In the case of weighted feature vectors, various term weighting schemes have been proposed, with the most well-known ones being TF (Term Frequency), TF-IDF (Term Frequency - Inverse Document Frequency) and several variants of them [23], [32], [18]. Although these weighting schemes were initially introduced in the NLP and IR fields, they have also been applied in the TC task. Paltoglou and Thelwall [28] reported that, in the case of sentiment analysis, extensions of the TF-IDF weighting scheme introduced in the IR field can improve the classification accuracy. A comprehensive review of this area is offered in the article by Sebastiani [35].

b) Graph-based text classification: Close to our work are other graph-based text classification methods. Jiang et al. [13] and Rousseau et al. [31] proposed also to model documents as graphs; after that, graph mining algorithms are applied to extract frequent subgraphs, which are then used to produce feature vectors for classification. Wang et al. [38] introduced a term graph model, that contrary to our approach, it captures the relationships among terms using frequent itemset mining techniques. Aery and Chakravarthy [1] proposed InfoSift, a graph-matching based method for document classification. Close to our work is the approach followed in [12], where they proposed to use random walks in order to determine the importance of a term. In this paper, we provide evidences that we can rely on simpler graph-based criteria to achieve even better results.

c) Graph-based text mining and IR: Graph representation of documents is common in other text analysis task, including ad hoc Information Retrieval, keyword extraction and summarization [3], [34], [24], [7], [6], [29], [30], [10]. Our method moves on a similar axis as these techniques, but the application is on the TC task.

III. Background and Preliminaries

In this section, we discuss the basic components of the pipeline to deal with the TC problem. Initially, we introduce the traditional Bag-of-Words document representation and the corresponding frequency-based TF, TF-IDF term weighting criteria that can be derived by this model and have extensively been used in TC and most of the common text analytics tasks (e.g., IR, document clustering, keyword extraction). Then, we briefly describe the classification models that are used in the core of a TC system.

A. Bag-of-Words Document Representation and Term Weighting

Figure 1 depicts the basic pipeline of a TC system. The pipeline that typically is followed to deal with the problem is similar to the one applied in any classification problem; the goal is to learn the parameters of a classifier from a collection of training documents (with known class information) and then to predict the class of unlabeled documents.

The first step in text categorization is to transform documents, which typically are strings of characters, into a representation suitable for the learning algorithm and the classification task. The main approach here is to apply the *Vector Space Model*, a spatial representation of text documents. In this model, each document is represented by a vector in the n-dimensional space, where each dimension corresponds to a term (i.e., word) from the overall vocabulary of the given document collection. More formally, let $\mathcal{D} = \{d_1, d_2, \ldots, d_m\}$ denotes a collection of m documents, and $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$ be the dictionary, i.e., the set of terms in the corpus $\mathcal{D}$. The set of terms $\mathcal{T}$ can be obtained either directly from the documents of after applying some standard natural language processing techniques, such as tokenization, stop-words removal and stemming [4]. Each document $d_i \in \mathcal{D}$ is represented as a vector of term weights $d_i = \{w_{i,1}, w_{i,2}, \ldots, w_{i,n}\}$, where $w_{i,k}$ is the weight of term k in document d_i. That way, data can be represented by the *Document-Term matrix* of size $m \times n$, where the rows correspond to documents and the columns to the different terms (i.e., features) of set $\mathcal{T}$. Note that, due to the large number of features of this matrix, dimensionality reduction techniques, such as *Latent Semantic Analysis* (LSA) [39], can be applied [35].

The main question regarding the Vector Space Model is how to find appropriate *weights* for the terms within a document. In the traditional *Bag-of-Words* model for document representation, each document is represented as the bag (multiset) of its words, disregarding grammar and even word order. Under this model, the importance of a term for a document is mainly based on the frequency of this term. That is, the weight of a term $t \in \mathcal{T}$ within a document $d \in \mathcal{D}$ is based on the frequency $tf(t, d)$ of the term in the document (TF weighting scheme). Furthermore, terms that occur frequently in one document but rarely in the rest of the documents, are more likely to be relevant to the topic of the document. This is known as the inverse document frequency (IDF) factor, and is computed at the collection level. It is obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient, as follows:

$$idf(t, \mathcal{D}) = \log \frac{m + 1}{|\{d \in \mathcal{D} : t \in d|\}},$$

where m is the total number of documents in collection $\mathcal{D}$, and the denominator captures the number of documents that

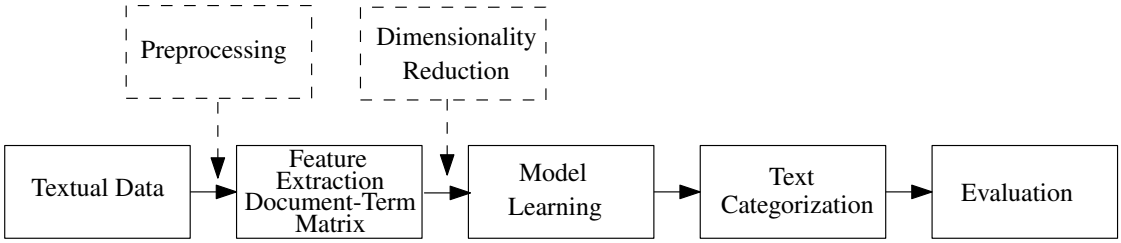

Fig. 1: Basic pipeline of the Text Categorization task.

term t appears. Here, we refer to the TF-IDF weighting scheme proposed in [36], also called pivoted normalization weighting:

$$tf\text{-}idf(t,d) = \frac{1+\ln\left(1+\ln(tf(t,d))\right)}{1-b+b\times\frac{|d|}{avgdl}} \times idf(t,\mathcal{D}),$$

where d is the length of the document, $avgdl$ is the average document length and parameter b is set by default to 0.20. This scoring function captures the intuitions that (i) the more often a term occurs in a document, the more it is representative of its content, and (ii) the more documents a term occurs in, the less discriminating it is[1]. Using the TF-IDF score of each term for each document, we can fill in the weights of the Document-Term matrix.

B. Classification Models

The core component of the TC task is the classification model. Each document $d_i \in \mathcal{D}$ is associated with a class label y_i, forming the class (or category) vector $\mathbf{Y}$. Then, the goal of the TC problem is to predict the class labels for a set of test documents. Based on this formulation, traditional classification algorithms (e.g., SVMs, Logistic Regression) can be applied to predict the category label of test documents [5]. As we will present later on, in this paper we consider the classification model as a black box. More precisely, in our experimental results we apply the *Support Vector Machine* (SVM) classification model [14] due to its superior performance in the TC task.

IV. Proposed Method

As we have already discussed, the traditional Bag-of-Words representation of documents and the corresponding scoring functions, do not retain information about the ordering and position of the terms in the document. Even if we consider the n-gram model, still information about the relationship between two different n-grams is ignored. Here, we propose to use a graph model for document representation, and we examine its performance in the TC task. After giving an overview of our approach, we provide details on (i) how to transform a document to a graph and (ii) how to weight the importance of terms under this model.

[1]Several variants of the TF-IDF score have been proposed. See also the description given in Ref. [35].

A. Overview

The proposed approach to TC follows the general pipeline depicted in Fig. 1. The novel point here concerns the way that the Document-Term matrix is weighed. Instead of using term frequency criteria, we show that by constructing a graph that captures co-occurrence relationships between the terms of a document, we retain structural information about the document that can improve the TC task. Algorithm 1 shows the basic steps of the proposed representation and weighting approach. Note that, the same representation and weighting mechanism is applied to both the training and test datasets.

Algorithm 1 Graph-Based Term Weighting Method

Input: Collection of documents $\mathcal{D} = \{d_1, d_2, \ldots, d_m\}$ and set (dictionary) of terms $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$
Output: Term weights $tw(t,d)$ for each term $t \in \mathcal{T}$ to each document $d \in \mathcal{D}$

1: **for** $d \in \mathcal{D}$ **do**
2: **(Graph Construction)** Construct a graph $G_d = (V, E)$. Each node $v \in V$ corresponds to a term $t \in \mathcal{T}$ of document d. Add edge $e = (u, v)$ between terms u and v if they co-occur within the same window of size w
3: **(Term Weighting)** Consider a node centrality criterion. For each term $t \in \mathcal{T}$, compute the weight $tw(t,d)$ based on the centrality score of node t in graph G_d and fill in the Document-Term matrix
4: **end for**

B. Graph Construction

Here, we describe how a document can be represented by a graph. We refer to this modeling approach as the Graph-of-Words (GoW) model [29]. As we noted above, the Graph-of-Words is a different document representation model that captures relationships between terms, questioning the term independence assumption [29], [7]. In general, each document $d \in \mathcal{D}$ is represented by a graph $G_d = (V, E)$, where the nodes correspond to the terms t of the document and the edges capture co-occurence relations between terms within a fixed-size sliding window of size w. That is, for all the terms that co-occur within the window, we add edges between the corresponding nodes (note that, the windows are overlapping starting from the first term of the document; at each step we simply remove the first term and add the new one from the document). As graphs constitute rich modeling structures, several parameters about the construction phase need to be specified.

1) Directed vs. undirected graph: One parameter of the GoW model, is if the graph representation of the document will be directed or undirected [9]. Directed graphs are able to preserve actual flow of a text, while in undirected ones, an edge captures co-occurrence of two terms whatever the respective order between them is. For example, in the field of Information Retrieval [29], directed graphs were selected to model documents. In this paper, we need to examine which representation is more appropriate for text categorization. As we will present in Sec. V, in our preliminary experimental evaluation we have considered both cases.

2) Weighted vs. unweighted graph: One approach is to consider the graphs as weighted. That is, the higher the number of co-occurences of two terms in the document, the higher the weight of the corresponding edge (i.e., the weight of each edge will be equal to the number of co-occurences of its endpoints). The second option is to consider the graphs as unweighted. In our preliminary experiments, we have focused on unweighted graphs, due to the simplicity of the model.

3) Size w of the sliding window: As we described above, according to the GoW model, we add edges between the terms of the document that co-occur within a sliding window of size w. The size of the window is yet another parameter of the model. As we will present shortly, for our experiments we have considered window of size $w = 3$, since this was performing well compared to other values. For example, in the Information Retrieval field, window of size $w = 4$ produced accurate results [29]. Note that, although by increasing the size of the window we are able to capture co-occurrence relationships between not necessarily nearby terms, the produced graph becomes relatively dense.

C. Term Weighting Criteria

After creating the graph of each document, we can proceed with the term weighing process. In the case where the document is represented by the Bag-of-Words model, the term frequency (TF) criterion (or TF-IDF) constitutes the basis for weighting the terms of each document. How this can be done in the Graph-of-Words model? The answer is given by utilizing node centrality criteria of the graph [25], [9]. That way, the importance of a term in a document can be inferred by the importance of the corresponding node in the graph. In graph theory and network analysis domains, several centrality criteria[2] have been proposed. Some of them have already been explored in graph-based Information Retrieval [29], [7] and keyword extraction [17], [30].

One important characteristic of these measures is that they consider either local information of the graph (e.g., degree centrality, in-degree/out-degree centrality in directed networks, weighted degree in weighted graphs, clustering coefficient) [9], or more global information, in the sense that the importance of a node is based on the properties of the node globally in the graph (e.g., PageRank centrality, eigenvector centrality, betweenness centrality, closeness centrality). In our preliminary experimental evaluation, we have considered both types of centrality criteria, depending also on the type of graph (i.e., directed or undirected) that we have chosen to construct (this is happening because some centrality criteria can be only defined on directed or undirected graphs).

Let $G_d = (V, E)$ be the graph representation of document $d \in \mathcal{D}$. Next, we briefly describe the centrality measures that were used in our preliminary experimental evaluation [25].

- *Degree centrality*. The degree centrality is one of the simplest node importance criteria, which captures the number of neighbors that each node has. Note that, the degree centrality is a local criterion. Let $\mathcal{N}(i)$ be the set of nodes connected to node i. Then, the degree centrality can be derived based on the following formula:

$$\text{degree\_centrality}(i) = \frac{|\mathcal{N}(i)|}{|V| - 1}.$$

- *In-degree and out-degree centrality*. Those two centrality measures constitute extensions of the degree centrality in directed networks, where we treat independently the in-degree (number of incoming edges) and out-degree (number of outgoing edges) of each node.

- *Closeness centrality*. In general, closeness centrality measures how close a node is to all other nodes in the graph. Let $dist(i, j)$ be the shortest path distance between nodes i and j. The closeness centrality of a node i is defined as the inverse of the average shortest path distance from the node to any other node in the graph [27]:

$$\text{closeness}(i) = \frac{|V| - 1}{\sum_{j \in V} dist(i, j)}.$$

  Contrary to degree centrality, the closeness score is a global metric, in the sense that it combines information from all the nodes of the graph. Here we compute the closeness centrality in the undirected graph.

The above list of centrality criteria is limited and describes only those that are used in our experimental evaluation. In fact, any other centrality criterion can be used to weight terms for the TC task.

After choosing a centrality criterion, we can directly assign a $tw(t, d)$ of the term t in document d. We refer to this weighting scheme as TW (in a similar way as the TF score in the BoW model). Furthermore, we can extend this weighting criterion, by considering information about the inverse document frequency of the term t in the collection $\mathcal{D}$. That way, we can derive the TW-IDF model as follows [29]:

$$tw\text{-}idf(t, d) = \frac{tw(t, d)}{1 - b + b \times \frac{|d|}{avgdl}} \times idf(t, \mathcal{D}).$$

In the experimental results, we set parameter $b = 0.003$ as suggested in [29].

Lastly, an important point concerns the computational complexity of the chosen centrality criteria. As we expect, some criteria are more efficient to be computed (e.g., degree centrality), while other no. This potential trade-off between

[2]Wikipedia's lemma for *network centrality*: http://en.wikipedia.org/wiki/Centrality.

classification accuracy and complexity of computing the features is also important, since it can affect the overall performance (in terms of time complexity) of the TC task.

V. Experimental Evaluation

In this section, we present our preliminary experimental evaluation of the graph-based term weighting criteria for TC. Before presenting our results, we describe the datasets used in the study and the experimental setup.

A. Description of the Datasets

We have performed experiments with the *Reuters-21578 R8* and *WebKB* datasets. The documents of the *Reuters-21578 R8* collection correspond to news that appeared on the Reuters newswire in 1987 and belong to one out of eight possible categories. The *WebKB* dataset corresponds to academic webpages that belong to four different categories. Both datasets have been split into the **Train** and **Test** parts[3]. Tables I and II give details about the datasets. The final evaluation of our method is done on the **Test** part and the goal will be to predict the class labels.

TABLE I: Description of the *Reuters-21578 R8* dataset.

Class Label	# of Train docs	# of Test docs	Total # of docs
acq	1,596	696	2,292
crude	253	121	374
earn	2,840	1,083	3,923
grain	41	10	51
interest	190	81	271
money-fx	206	87	293
ship	108	36	144
trade	251	75	326
Total	**5,485**	**2,189**	**7,674**

TABLE II: Description of the *WebKB* dataset.

Class Label	# of Train docs	# of Test docs	Total # of docs
project	336	168	504
course	620	310	930
faculty	750	374	1124
student	1097	544	1641
Total	**2,803**	**1,396**	**4,199**

The datasets given here (both **Train** and **Test**) have been preprocessed. Stopwords and words that are less than 3 characters long have been removed from the documents. Finally, Porter's stemmer has been applied, keeping only the root of each term.

B. Experimental Setup and Tools

We have implemented the proposed graph-based term weighting criteria in Python using the `NetworkX` library[4]. For the classification model, we have used SVM with linear kernel, implemented by the `scikit-learn` library[5].

We perform preliminary experiments on the **Test** part of the datasets, after training the classification model on the **Train** collection. The classification performance is evaluated using the macro-averaged Precision, Recall, F1-score measures and classification accuracy. Furthermore, since we deal with a multi-class classification problem, for the simplest setup of our approach, i.e., degree centrality, we also report the classification results for each category of the datasets. Our method is compared against the Bag-of-Words model using (i) TF and (ii) TF-IDF scores. After experimentation, we set window size $w = 3$, since it consistently produces good results.

[3]http://web.ist.utl.pt/acardoso/datasets/

[4]http://networkx.github.io/

[5]http://scikit-learn.org/

TABLE III: Classification results of the *Reuters-21578 R8* for different choices of term weighting criteria, with and without IDF penalization.

Weighting	F1-score	Accuracy
TF	0.9127	0.9634
TW, degree	0.8991	0.9611
TW, in-degree	0.8037	0.9438
TW, out-degree	0.8585	0.9546
TW, closeness	0.9125	0.9625
TF-IDF	0.8962	0.9616
TW-IDF, degree	**0.9175**	**0.9661**
TW-IDF, in-degree	0.8985	0.9629
TW-IDF, out-degree	0.8854	0.9625
TW-IDF, closeness	0.8846	0.9547

TABLE IV: Classification results of the *WebKB* dataset for different choices of term weighting criteria, with and without IDF penalization.

Weighting	F1-score	Accuracy
TF	0.8741	0.8853
TW, degree	**0.8962**	**0.9032**
TW, in-degree	0.8286	0.8545
TW, out-degree	0.8365	0.8603
TW, closeness	0.8960	0.9004
TF-IDF	0.8331	0.8538
TW-IDF, degree	0.8800	0.8882
TW-IDF, in-degree	0.7890	0.8381
TW-IDF, out-degree	0.8049	0.8474
TW-IDF, closeness	0.8505	0.8674

C. Results

We report preliminary experiments regarding the performance of the graph-based term weighting criteria. Although the traditional BoW model already performs well for the *Reuters-21578 R8*, here we are mostly interested to examine the capabilities of the new model to achieve results close to or superior to the ones of BoW.

Tables III and IV present our preliminary results. For each case, we are interested to compare the performance of the TF and TF-IDF mechanisms to the one of TW and TW-IDF respectively, using the centrality criteria presented earlier. As we can observe, the proposed weighting schemes produced by the GoW model, perform relatively well and in many cases outperform the TF and TF-IDF models. This is mainly evident in the *WebKB* dataset; as we can observe, for the *Reuters-21578 R8* dataset, the baseline methods already give high F1-score and accuracy results. Furthermore, in the case of the *Reuters-21578 R8* dataset, the best results were achieved with the TW-IDF scheme with degree centrality. Although we have considered only a few settings of the parameter space here, the best performance in both datasets is achieved by

TABLE V: Comparison of the TF weighting scheme vs. TW with degree centrality, for each category of the *Reuters-21578 R8* dataset.

Class Label	Precision		Recall		F1-score	
	BoW	GoW	BoW	GoW	BoW	GoW
acq	0.9617	0.9405	0.9741	0.9770	0.9679	0.9584
crude	0.9667	0.9649	0.9587	0.9091	0.9627	0.9362
earn	0.9844	0.9889	0.9917	0.9871	0.9880	0.9880
grain	1.0000	1.0000	0.9000	0.7000	0.9474	0.8235
interest	0.8857	0.9167	0.7654	0.8148	0.8212	0.8267
money-fx	0.8415	0.8506	0.7931	0.8506	0.8166	0.8506
ship	0.9375	1.0000	0.8333	0.7222	0.8824	0.8387
trade	0.8875	0.9114	0.9467	0.9600	0.9161	0.9351
Average	0.9629	0.9616	0.9635	0.9612	0.9629	0.9606

TABLE VI: Comparison of the TF-IDF weighting scheme vs. TW-IDF with degree centrality, for each category of the *Reuters-21578 R8* dataset.

Class Label	Precision		Recall		F1-score	
	BoW	GoW	BoW	GoW	BoW	GoW
acq	0.9743	0.9565	0.9799	0.9784	0.9771	0.9673
crude	0.9496	0.9492	0.9339	0.9256	0.9417	0.9372
earn	0.9800	0.9871	0.9945	0.9898	0.9872	0.9885
grain	1.0000	1.0000	0.9000	0.9000	0.9474	0.9474
interest	0.8243	0.9200	0.7531	0.8519	0.7871	0.8846
money-fx	0.7952	0.8721	0.7586	0.8621	0.7765	0.8671
ship	0.9630	1.0000	0.7222	0.6667	0.8254	0.8000
trade	0.9103	0.9241	0.9467	0.9733	0.9281	0.9481
Average	0.9608	0.9663	0.9616	0.9662	0.9608	0.9656

the degree centrality (undirected graph), with or without IDF normalization.

Furthermore, we have compared the performance of the BoW and GoW weighting schemes for each category of the datasets. In Tables V, VI and Tables VII, VIII we report the performance per category of the TF vs. TW schemes and TF-IDF vs. TW-IDF for each dataset. In all cases, we use degree centrality for the GoW weighting schemes. As we can observe, for the *Reuters-21578 R8* dataset, the performance of the GoW model is very close to the one of BoW weighting mechanisms; in the case where we apply IDF normalization, the TW-IDF scheme performs in general better than TF-IDF, especially in categories of small size (e.g., grain, interest). For the case of the *WebKB* dataset, in almost all categories, both TW and TW-IDF significantly outperform the baseline weighting mechanisms.

VI. Discussion and Future Work

The goal of this paper was to introduce a new paradigm for TC. The main idea is to represent each document as a graph, based on co-occurrence information between terms. Then, the importance of a term to a document can be specified using node centrality centrality criteria, such as degree and closeness centrality. We have performed preliminary experimental evaluation and the results are encouraging regarding the applicability of the proposed term weighting schemes to the TC task.

Due to the rich modeling properties of graphs, several parameters need to be specified for the problem. Here, we have only considered a small subset of them, but a better exploration of the parameter's space is needed in order to have a more complete picture of the capabilities of this model. Next, we describe some problems that are interesting to be addressed under this context.

TABLE VII: Comparison of the TF weighting scheme vs. TW with degree centrality, for each category of the *WebKB* dataset.

Class Label	Precision		Recall		F1-score	
	BoW	GoW	BoW	GoW	BoW	GoW
course	0.9233	0.9736	0.9323	0.9516	0.9278	0.9625
faculty	0.8534	0.8944	0.8717	0.8610	0.8624	0.8774
project	0.8259	0.8766	0.7857	0.8036	0.8049	0.8385
student	0.9039	0.8791	0.8989	0.9357	0.9014	0.9065
Average	0.8852	0.9039	0.8854	0.9033	0.8852	0.9029

TABLE VIII: Comparison of the TF-IDF weighting scheme vs. TW-IDF with degree centrality, for each category of the *WebKB* dataset.

Class Label	Precision		Recall		F1-score	
	BoW	GoW	BoW	GoW	BoW	GoW
course	0.8858	0.9639	0.9258	0.9484	0.9054	0.9561
faculty	0.8411	0.8368	0.8209	0.8503	0.8309	0.8435
project	0.7358	0.8889	0.6964	0.7619	0.7156	0.8205
student	0.8777	0.8818	0.8842	0.9191	0.8810	0.9001
Average	0.8526	0.8889	0.8539	0.8883	0.8531	0.8878

A. Exploration of Parameter's Space

As we discussed earlier, there are several ways to construct a graph from the document (e.g., directed or undirected). Additionally, depending on the graph construction, many diverse centrality criteria can be applied in order to weight the terms. A complete exploration of this space will give better insights about the ability of this method to outperform the traditional TF and TF-IDF scoring functions in the TC task, and our preliminary results support this argument.

B. Graph-Based Inverse Collection Weight

In the case of TF-IDF scheme, the frequency of each term in the document (TF factor) is penalized by the number of documents in which it appears (IDF factor). In our experiments, we simply penalize the TW graph-based weight by the IDF factor of the BoW model. How to derive a term penalization factor using the GoW model? For example, can we create a graph from all documents and consider centrality metrics (such as degree centrality) at the collection-level graph?. The last point has not been explored in the related literature and it would be interesting to examine it in the context of TC.

C. Graph-Based Dimensionality Reduction

A basic preprocessing step in any classification task is the one of dimensionality reduction. As we described earlier, in the context of TC we can apply dimensionality reduction in the Term-Document matrix (e.g., LSA). However, it would be interesting to examine if we can extend the task of dimensionality reduction to the graph representation of the documents. In the graph space, dimensionality reduction can be defined

either on the nodes of the graph (i.e., the terms), either on the edges or on both. For example, it would be an interesting research direction to examine if graph sampling [20] or edge sparsification [33] techniques can be applied in graphs that represent documents and if this can improve the TC task.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments. Fragkiskos D. Malliaros is a recipient of the Google Europe Fellowship in Graph Mining, and this research is supported in part by this Google Fellowship.

REFERENCES

[1] M. Aery and S. Chakravarthy, "Infosift: Adapting graph mining techniques for text classification," in *FLAIRS '05: Proceedings of the Eighteenth International Florida Artificial Intelligence Research Society Conference*, 2005, pp. 277–282.

[2] I. Androutsopoulos, J. Koutsias, K. Chandrinos, G. Paliouras, and C. Spyropoulos, "An Evaluation of Naive Bayesian Anti-Spam Filtering," in *Workshop on Machine Learning in the New Information Age*, 2000.

[3] R. Angelova and G. Weikum, "Graph-based text classification: Learn from your neighbors," in *SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006, pp. 485–492.

[4] R. A. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., 1999.

[5] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006.

[6] R. Blanco and C. Lioma, "Random walk term weighting for information retrieval," in *SIGIR '07: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2007, pp. 829–830.

[7] R. Blanco and C. Lioma, "Graph-based term weighting for information retrieval," *Inf. Retr.*, vol. 15, no. 1, pp. 54–92, 2012.

[8] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification," in *ACL '07: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, 2007.

[9] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[10] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *J. Artif. Int. Res.*, vol. 22, no. 1, pp. 457–479, 2004.

[11] J. Fürnkranz, "A study using n-gram features for text categorization," Austrian Research Institute for Artificial Intelligence, Tech. Rep. OEFAI-TR-98-30, 1998.

[12] S. Hassan, R. Mihalcea, and C. Banea, "Random walk term weighting for improved text classification," *Int. J. Semantic Computing*, vol. 1, no. 4, pp. 421–439, 2007.

[13] C. Jiang, F. Coenen, R. Sanderson, and M. Zito, "Text classification using graph mining-based feature extraction," *Knowl.-Based Syst.*, vol. 23, no. 4, pp. 302–308, 2010.

[14] T. Joachims, "Text categorization with suport vector machines: Learning with many relevant features," in *ECML '98: Proceedings of the 10th European Conference on Machine Learning*, 1998, pp. 137–142.

[15] T. Joachims, "Transductive inference for text classification using support vector machines," in *ICML '99: Proceedings of the Sixteenth International Conference on Machine Learning*, 1999, pp. 200–209.

[16] S. Kim, K. Han, H. Rim, and S. Myaeng, "Some effective techniques for naive bayes text classification," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 11, pp. 1457–1466, 2006.

[17] S. Lahiri, S. R. Choudhury, and C. Caragea, "Keyword and keyphrase extraction using centrality measures on collocation networks," *CoRR*, 2014.

[18] M. Lan, C.-L. Tan, H.-B. Low, and S.-Y. Sung, "A comprehensive comparative study on term weighting schemes for text categorization with support vector machines," in *WWW '05: Proceedings of the 4th International World Wide Web Conference*, 2005, pp. 1032–1033.

[19] M. Lan, C. L. Tan, J. Su, and Y. Lu, "Supervised and traditional term weighting methods for automatic text categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 721–735, 2009.

[20] J. Leskovec and C. Faloutsos, "Sampling from large graphs," in *KDD 06: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006, pp. 631–636.

[21] D. D. Lewis, "Representation and learning in information retrieval," Ph.D. dissertation, University of Massachusetts, Amherst, MA, USA, 1992.

[22] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text classification using string kernels," *J. Mach. Learn. Res.*, vol. 2, pp. 419–444, 2002.

[23] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[24] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *EMNLP '04: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004, pp. 404–411.

[25] M. Newman, *Networks: An Introduction*. Oxford University Press, Inc., 2010.

[26] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using em," *Mach. Learn.*, vol. 39, no. 2-3, pp. 103–134, May 2000.

[27] K. Okamoto, W. Chen, and X.-Y. Li, "Ranking of closeness centrality for large-scale social networks," in *FAW '08: Proceedings of the 2nd Annual International Workshop on Frontiers in Algorithmics*, 2008, pp. 186–195.

[28] G. Paltoglou and M. Thelwall, "A study of information retrieval weighting schemes for sentiment analysis," in *ACL '10: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010, pp. 1386–1395.

[29] F. Rousseau and M. Vazirgiannis, "Graph-of-word and TW-IDF: new approach to ad hoc IR," in *CIKM'13: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, 2013, pp. 59–68.

[30] F. Rousseau and M. Vazirgiannis, "Main core retention on graph-of-words for single-document keyword extraction," in *ECIR '15: Proceedings of the 37th European Conference on Information Retrieval*, 2015, pp. 382–393.

[31] F. Rousseau, E. Kiagias, and M. Vazirgiannis, "Text categorization as a graph classification problem," in *ACL '15: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, 2015.

[32] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, 1988.

[33] V. Satuluri, S. Parthasarathy, and Y. Ruan, "Local graph sparsification for scalable clustering," in *SIGMOD '11: Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2011, pp. 721–732.

[34] A. Schenker, M. Last, H. Bunke, and A. Kandel, "Classification of web documents using a graph model," in *ICDAR '03: Seventh International Conference on Document Analysis and Recognition*, 2003, pp. 240–244.

[35] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, 2002.

[36] A. Singhal, J. Choi, D. Hindle, D. D. Lewis, and F. Pereira, "At&t at trec-7," in *TREC '99: Proceedings of the 7th Text Retrieval Conference*, 1999, pp. 239–252.

[37] A. Sun and E.-P. Lim, "Hierarchical text classification and evaluation," in *ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining*, 2001, pp. 521–528.

[38] W. Wang, D. Do, and X. Lin, "Term graph model for text classification," in *Advanced Data Mining and Applications*. Springer Berlin Heidelberg, 2005, vol. 3584, pp. 19–30.

[39] B. Yu, Z.-b. Xu, and C.-h. Li, "Latent semantic analysis for text categorization using neural network," *Know.-Based Syst.*, vol. 21, no. 8, pp. 900–904, 2008.