Fusing Document, Collection and Label Graph-based Representations with Word Embeddings for Text Classification

Konstantinos Skianis
École Polytechnique
France
kskianis@lix.polytechnique.fr

Fragkiskos D. Malliaros
CentraleSupélec and Inria Saclay
France
fragkiskos.me@gmail.com

Michalis Vazirgiannis
École Polytechnique
France
mvazirg@lix.polytechnique.fr

Abstract

Contrary to the traditional *Bag-of-Words* approach, we consider the *Graph-of-Words* (GoW) model in which each document is represented by a graph that encodes relationships between the different terms. Based on this formulation, the importance of a term is determined by weighting the corresponding node in the document, collection and label graphs, using node centrality criteria. We also introduce novel graph-based weighting schemes by enriching graphs with word-embedding similarities, in order to reward or penalize semantic relationships. Our methods produce more discriminative feature weights for text categorization, outperforming existing frequency-based criteria. Code and data are available online[1].

1 Introduction

With the rapid growth of the social media and networking platforms, the available textual resources have been increased. *Text categorization* or classification (TC) refers to the supervised learning task of assigning a document to a set of two or more predefined categories (or classes) (Sebastiani, 2002). Well-known applications of TC include sentiment analysis, spam detection and news classification.

In the TC pipeline, each document is modeled using the so-called *Vector Space Model* (Baeza-Yates and Ribeiro-Neto, 1999). The main issue here is how to find appropriate weights regarding the importance of each term in a document. Typically, the *Bag-of-Words* (BoW) model is applied and a document is represented as a multiset of its terms, disregarding co-occurence between the terms; using this model, the importance of a term in a document is mainly determined by the frequency of the term. Although several variants and extensions of this modeling approach have been proposed (e.g., the n-gram model (Baeza-Yates and Ribeiro-Neto, 1999)), the main weakness comes from the underlying term independence assumption, where the order of the terms is also completely disregarded.

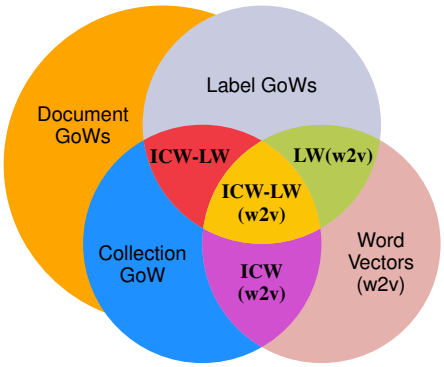

Figure 1: Blending different types of GoWs and word vector similarities in one framework.

After the introduction of deep learning models for TC (Blunsom et al., 2014; Kim, 2014), recent work by Johnson and Zhang (2015) shows how we could effectively use the order of words with CNNs (LeCun et al., 1995). In many cases though, space and time limitations may arise due to complex neural network architectures. As stated in work by Joulin et al. (2017), computation can still be expensive and prohibitive.

In this paper, we explore fast term weighting criteria for TC that go beyond the term independence assumption. The notion of dependencies between terms is introduced via a *Graph-of-Words* (GoW) representation model. Under this model, each term is represented as a node in the graph and the edges capture co-occurrence relationships

[1]Code and data: github.com/y3nk0/Graph-Based-TC

Proceedings of the Twelfth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-12), pages 49–58
New Orleans, Louisiana, June 6, 2018.

of terms with a specified distance in the document. We implicitly consider information about n-grams in the document as well as the collection of documents – expressed by paths in the graph – without increasing the dimensionality of the problem. Furthermore, we introduce word-embedding similarities as weights in the GoW approach, in order to further boost the performance of our methods. Finally, we successfully mix document, collection and label GoWs along with word vector similarities into a single powerful graph-based framework. An overview of our approach is shown in Fig. 1.

2 Related Work

Term weighting schemes. A core aspect in the *Vector Space Model* for document representation, is how to determine the importance of a term within a document. Many criteria have been introduced with the most prominent ones being TF, TF-IDF (Salton and Buckley, 1988; Singhal et al., 1996; Baeza-Yates and Ribeiro-Neto, 1999; Robertson, 2004) and Okapi BM25 (Robertson et al., 1995), while some recent ones include N-gram IDF (Shirakawa et al., 2015). Lan et al. (2005) conducted a comparative study of frequency-based term weighting criteria for text categorization; one of their outcomes was that, in many cases, the IDF factor is not significant for the categorization task, leading to no improvement of the performance. It is interesting to point out here that, more specialized approaches have been proposed for specific classification tasks, such as the Delta TF-IDF method that constitutes an extension of TF-IDF for sentiment analysis (Martineau and Finin, 2009). However, most of the previously proposed frequency-based weights consider the document as a *Bag-of-Words*; that way, any structural information about the ordering or in general, syntactic relationship of the terms, is ignored by the weighting process.

Text categorization. A number of diverse approaches have been proposed for TC (Joachims, 1998; McCallum and Nigam, 1998; Nigam et al., 2000; Sebastiani, 2002; Kim et al., 2006). 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. In the case of weighted feature vectors, various term weighting schemes have been used, with the most well-known ones being TF (Term Frequency), TF-IDF (Term Frequency - Inverse Document Frequency). 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 (2010) reported that, in the case of sentiment analysis, extensions of the TF-IDF weighting schemes introduced in the IR field, can further improve the classification accuracy. A comprehensive review of this area is offered in the article by Sebastiani (2002).

Deep Learning for TC. With the rise of deep learning models, CNNs were applied for text classification (Blunsom et al., 2014; Kim, 2014; Johnson and Zhang, 2015). Next, Zhang et al. (2015) presented Character-level CNNs for the task of TC. Finally, Joulin et al. (2017) proposed a novel text classifier which achieves equivalent performance to state-of-the-art TC models, with faster learning times. Our work does not focus on the classifier part, as the aforementioned methods, but on the extraction of better features.

Graph-based text categorization. In the related literature, most of the graph-based method for TC, rely on graph mining algorithms that are applied to extract frequent subgraphs, which are then used to produce feature vectors for classification (Deshpande et al., 2005; Jiang et al., 2010; Rousseau et al., 2015; Nikolentzos et al., 2017). The basic shortcoming of those methods stems from the computational complexity of the frequent subgraph mining algorithm. Furthermore, most of these methods require from the user to set the *support* parameter, which concerns the frequency of appearance of a subgraph. Close to our work are the approaches followed by Hassan et al. (2007) and Malliaros and Skianis (2015); they explored how random walks and other graph centrality criteria can be applied to determine the importance of a term.

Graph-based text mining, NLP and IR. Representing documents as graphs is a well-known approach in NLP and IR. TextRank algorithm, proposed by Mihalcea and Tarau (2004), was among the first works that considered a random walk model similar to PageRank, over a graph representation of the document, in order to extract representative keywords and sentences. Later, sev-

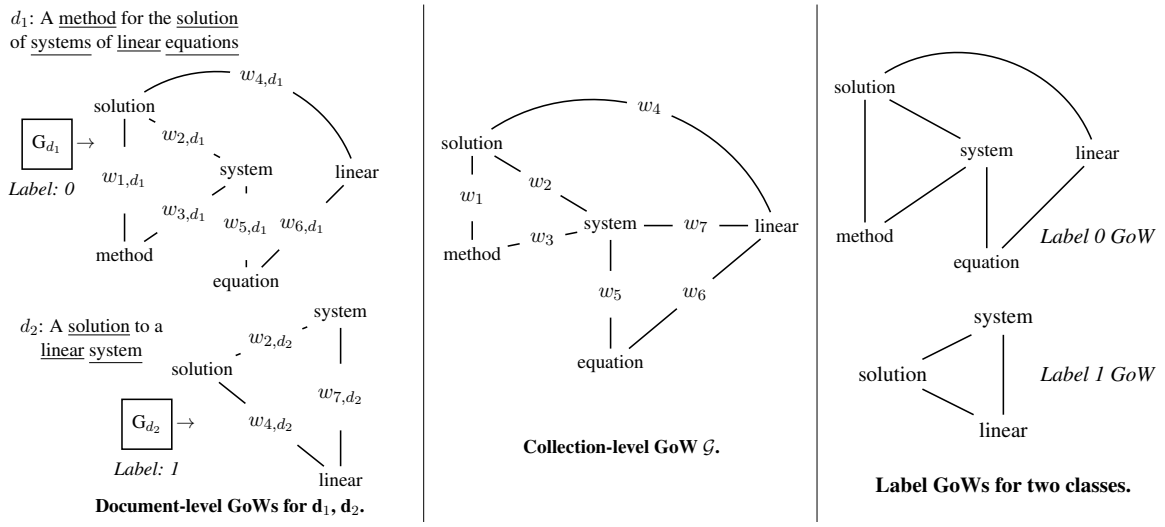

Figure 2: Example of document, collection-level and label GoWs for a collection composed by two documents and window size $w = 3$. Weights on the edges of G_{d_1} and G_{d_2} correspond to the similarity of two terms in the vector space. Here, Label GoWs are the same with Document GoWs (one document per class-label).

eral methods for those tasks were followed (Erkan and Radev, 2004; Litvak and Last, 2008; Boudin, 2013; Lahiri et al., 2014; Rousseau and Vazirgiannis, 2015). Another domain where graph-based term weighting schemes have been applied is the one of ad hoc Information Retrieval (Rousseau and Vazirgiannis, 2013). An interesting survey can be found in the work of Blanco and Lioma (2012) for a detailed description of graph-based methods in the text domain.

3 Preliminaries and Background

Let $\mathcal{D} = \{d_1, d_2, \ldots, d_m\}$ be a collection of documents and let $\mathcal{C} = \{c_1, c_2, \ldots, c_{|\mathcal{C}|}\}$ be the set of predefined categories. Text categorization is considered the task of assigning a boolean value to each pair $(d_i, c_i) \in \mathcal{D} \times \mathcal{C}$, i.e., assigning each document to one or more categories (Sebastiani, 2002). The main point here is how to find appropriate *weights* for the terms within a document. As we will present below shortly, our approach utilizes network centrality criteria.

Node Centrality Criteria. Centrality[2] represents a central notion in graph theory and network analysis in general; it constitutes of measures that capture the relative importance of the node in the graph based on specific criteria (Newman, 2010). One important characteristic of the centrality 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) (Newman, 2010), or more *global* information – in the sense that the importance of a node is determined by the properties of the node globally in the graph (e.g., PageRank, closeness). Let $G = (V, E)$ be a graph (directed or undirected), and let $|V|$, $|E|$ be the number of nodes and edges respectively. Next, we define basic centrality criteria that are used in the proposed methodology.

Degree centrality. The degree centrality is one of the simplest local node importance criteria, which captures the number of neighbors that each node has. 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: degree_centrality$(i) = \frac{|\mathcal{N}(i)|}{|V|-1}$.

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

PageRank centrality. PageRank counts the number and quality of edges to a node to determine a rough estimate of how important the node is: $\text{PR}(i) = \frac{1-\alpha}{|V|} + \alpha \sum_{\forall (j,i) \in E} \frac{\text{PR}(j)}{\text{out-deg}(j)}$, where α is the teleportation probability and out-deg(i) denotes the out degree on node i.

[2] en.wikipedia.org/wiki/Centrality.

4 Proposed Framework

In this section, we present the components of the proposed graph-based framework for TC.

4.1 Graph Construction

We model documents as graphs that capture dependencies between terms. More precisely, each document $d \in \mathcal{D}$ is represented by a graph $G_d = (V_d, E_d)$, where the nodes correspond to the terms t of the document and the edges capture co-occurrence relationships 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 of the graph. Note that, the windows are overlapping starting from the first term of the document; at each step, we simply remove the first term of the window and add the new one from the document. As graphs constitute rich modeling structures, several parameters about the construction phase need to be specified, including the directionality of the edges, the addition of edge weights, well as the size w of the sliding window. Fig. 2 gives a toy example of the construction of GoW for a collection composed by two documents.

To summarize, the key point of the graph-based representation for TC is the fact that it deals with the term independence assumption. Even if we consider the n-gram model, still information about the relationship between two different n-grams is not fully captured – as happens in the case of graphs. This has also been noted in other application domains (e.g., IR (Rousseau and Vazirgiannis, 2013)).

4.2 Term Weighting

Having the graph, the importance of a term in a document can be inferred by the importance of the corresponding node in the graph. In the previous section, we presented local and global centrality criteria that have been widely used for graph mining and network analysis purposes; here, we propose that those criteria can also be used for weighting terms in the TC task. That way, similar to TF, we can define the Term Weight (TW) weighting scheme as $\text{TW}(t, d) = \text{centrality}(t, d)$, where $\text{centrality}(t, d)$ corresponds to the score of term (node) t in the graph representation G_d of document d. The interesting point here is that TW can be used along with any centrality criterion in the graph, local or global.

Furthermore, we can extend this weighting scheme by considering information about the inverse document frequency (IDF factor) of the term t in the collection $\mathcal{D}$. That way, we can derive the TW-IDF model as follows:

$$\text{TW-IDF}(t, d) = \text{TW}(t, d) \times \text{IDF}(t, \mathcal{D}). \quad (1)$$

In fact, TW and TW-IDF constitute suites for graph-based term-weighting schemes and thus, can be applied in any text analytics task. Some of them have already been explored in graph-based IR (Rousseau and Vazirgiannis, 2013) and keyword extraction (Mihalcea and Tarau, 2004; Rousseau and Vazirgiannis, 2015).

The proposed weights are inferred from the interconnection of features (i.e., terms) – as suggested by the graph – and therefore information about n-grams is implicitly captured. That way, the feature space of the learning problem is kept to the one defined by the unique unigrams of our collection (instead of using simultaneously as features all the possible unigrams, bigrams, 3-grams, etc.), but the produced term weights incorporate n-gram information through the graph-based representation.

4.3 Inverse Collection Weight (ICW)

In this paragraph, we introduce the concept of *Inverse Collection Weight* (ICW) – a graph-based criterion to penalize the weight of terms that are "important" across the whole collection of documents. The main concept behind ICW is the *collection level* graph $\mathcal{G}$ – an extension of the *Graph-of-Words* in the collection of documents $\mathcal{D}$.

Definition 1 (Collection Level Graph $\mathcal{G}$) *Let $\{G_1, G_2, \ldots, G_d\}_{|\mathcal{D}|}$ be the set of graphs that correspond to all documents $d \in \mathcal{D}$. The collection level graph $\mathcal{G}$ is defined as the union of graphs $G_1 \cup G_2 \cup \ldots \cup G_d$ over all documents in the collection.*

The union of two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$ is defined as the union of their node and edge sets, i.e., $G \cup H = (V_G \cup V_H, E_G \cup E_H)$. The number of nodes in graph $\mathcal{G}$ is equal to the number of unique terms in the collection, while the number of edges is equal to the number of unique edges over all document-level graphs (see also Fig. 2).

This graph captures the overall dependencies between the terms of the collection; the relative overall importance of a term in the collection will

be proportional to the importance of the corresponding node in $\mathcal{G}$. Following similar methodological arguments as used for IDF (Robertson, 2004), we define a probability distribution over the nodes of $\mathcal{G}$ (or equivalently, the unique terms of $\mathcal{D}$), with respect to a centrality (term-weighting in our case) criterion; then, the probability of node (term) t will be:

$$\Pr(t) = \frac{\text{TW}(t, \mathcal{D})}{\sum_{v \in \mathcal{D}} \text{TW}(v, \mathcal{D})}. \quad (2)$$

Note that, in Eq. (2), we use $\mathcal{D}$ instead of $\mathcal{G}$; we consider that the space defined by the document collection $\mathcal{D}$ is equivalent to the one defined by graph $\mathcal{G}$ with respect to the unique terms of the collection. This way, the notion of $\text{TW}(t, \mathcal{D})$ used here is consistent with what was described earlier. Based on this, we define the ICW measure as:

$$\text{ICW}(t, \mathcal{D}) = \frac{\max_{v \in \mathcal{D}} \text{TW}(v, \mathcal{D})}{\text{TW}(t, \mathcal{D})}. \quad (3)$$

Instead of selecting the *maximum* centrality in the collection level (Eq. (3)), the *sum* of all centralities also yields good results.

ICW shares common intuition with the *inverse total term frequency* described in Robertson (2004). In fact, it can be considered as an extension of the total collection frequency of a term, to the graph-based document representation. Furthermore, similar to TW, it can be used along with any node centrality criterion.

Using ICW as a graph-based collection-level term penalization factor, we derive a new class of term-to-document weighting mechanism, namely TW-ICW. This weighting scheme is derived combining different local (i.e., document-level) and global (i.e., collection-level) criteria as follows:

$$\text{TW-ICW}(t, d) = \text{TW}(t, d) \times \log(\text{ICW}(t, \mathcal{D})).$$

In the case of TW and ICW, any centrality criterion can be applied. However, the computational complexity is a crucial factor that should be taken into account. Nevertheless, as we have noticed from the experimental evaluation, even using simple and easy-to-compute local criteria (e.g., degree), we achieve good classification performance.

4.4 Label Graphs

Shanavas et al. (2016) introduced supervised term weighting (TW-CRC) as a method to integrate class information with graphs. Similarly, we create a graph for each class (label), where we add all words of documents belonging to the respective class as nodes and their co-occurrence as edges. Our weighting scheme is a variant of TW-CRC; we define LW for a term t as:

$$\text{LW}(t) = \frac{\max(\deg(t, L))}{\max(\text{avg}(\deg(t, L)), \min(\deg(L))},$$

where the maximum degree of term t in all label graphs (L) is divided by the max of two values: the average degree of the term in all label graphs (except the one having the max degree) and the min degree of all the terms in all the label graphs. Then, we obtain ICW-LW as follows:

$$\text{ICW-LW}(t, d) = \log(\text{ICW}(t, \mathcal{D}) \times \text{LW}(t)),$$

and multiply it with $\text{TW}(t, d)$ to get TW-ICW-LW. Notice that, supervised frequency-based methods have also been proposed in previous work (Debole and Sebastiani, 2004; Huynh et al., 2011).

4.5 Edge Weighting using Word Embeddings

With our proposed framework, we can now use word embeddings (Bengio et al., 2003) in order to extract similarities between terms. Our goal is to integrate these similarities in the graph representation as weights on the edge between two words. The key idea behind our approach is that we want to reward semantically close words in the graph-document level (TW) and penalize them in the collection level (ICW).

The most commonly used similarity between two words t_1 and t_2 in the word-embedding space is cosine similarity, which ranges between -1 and 1. In order to have a valid distance metric, we need to bound this between 0 and 1. We use the angular similarity to represent the weight of an edge between two words, and since the vector elements may be positive or negative, the formula becomes:

$$\text{weight}(t_1, t_2) = 1 - \frac{\arccos(\text{sim}(t_1, t_2))}{\pi}. \quad (4)$$

The best performance was given by using Google's pre-trained word embeddings (Mikolov et al., 2013) and not by learning them by the datasets. Since the words included in the pre-trained version of word2vec are case sensitive and not stemmed, we did not apply any of these transformations. For words that do not appear in word2vec, we add a small value as similarity. Other distances (e.g. inverse euclidean, fractional) did not yield any further improvement.

Table 1: Datasets' statistics: #ICW shows the number of edges in the collection-level graph; #w2v: number of words that exist in pre-trained vectors.

	Train	Test	Voc	Avg	#w2v	#ICW
IMDB	1,340	660	32,844	343	27,462	352K
WEBKB	2,803	1,396	23,206	179	20,990	273K
20NG	11,293	7,528	62,752	155	54,892	1.7M
AMAZON	5,359	2,640	19,980	65	19,646	274K
REUTERS	5,485	2,189	11,965	66	9,218	163K
SUBJ.	6,694	3,293	8,639	11	8,097	58K

A similar approach for generic *keyphrase extraction* can be tracked in work by Wang et al. (2015). Providing more information in the weights, like number of co-occurrences between words, did not yield better results.

4.6 Classification Algorithms

Since the goal of this paper is to introduce new term weighting schemes, we rely on widely used classification algorithms. Specifically, we have used linear SVMs, due to their superior performance in TC (Joachims, 1998). Furthermore, as discussed in Leopold and Kindermann (2002), the choice of the kernel function of SVM is not very crucial, compared to the significance of the term weighting schemes.

5 Experiments

We have evaluated our method on six freely available standard TC datasets, covering multi-class document categorization, sentiment analysis and subjectivity. Specifically: (1) 20NG[3]: newsgroup documents belonging to 20 categories, (2) REUTERS[3]: 8 categories of Reuters-21578, (3) WEBKB[3]: 4 most frequent categories of webpages from Computer Science departments, (4) IMDB (Pang and Lee, 2004): positive and negative movie reviews; (5) AMAZON (Blitzer et al., 2007): product reviews acquired from Amazon over four different sub-collections; (6) SUBJECTIVITY (Pang and Lee, 2004): contains subjective sentences gathered from Rotten Tomatoes and objective sentences gathered from IMDB. A summary of the datasets can be found in Table 1.

In the experiments, linear SVMs were used with grid search cross-validation for tuning the C parameter. We also examined logistic regression, and observed similar performance. In the text preprocessing step, we have removed stopwords. No stemming or lowercase transformation was applied in order to match the words in `word2vec`.

For evaluation we use macro-average F1 score and classification accuracy on the test sets; that way, we deal with the skewed class size distribution of some datasets (Sebastiani, 2002). For the notation of the proposed schemes, we use TW (centrality measure) (e.g., TW (degree)) to indicate the centrality and TW-ICW (centrality at G, centrality at $\mathcal{G}$) (e.g., TW-ICW (degree, degree)) for the local and collection-level graphs respectively. In TW-IDF (w2v), we compute the weighted degree centrality on the document level, with word-embedding similarities as weights. Similarly, in TW-ICW (w2v) we compute both weighted centralities for document and collection graphs. Finally, we denote as TW-ICW-LW the blending of TW, ICW and label graphs (LW). In label graphs we only make use of the degree centrality, since it is fast and performs best.

5.1 Results

Table 2 presents the results concerning the categorization performance of the proposed schemes for the six datasets. As discussed previously, the size of the window considered to create the graphs is one of the model's parameters. From the extensive experimental evaluation that we have performed, we have concluded that small window sizes give the most persistent results across various datasets and weighting schemes. For completeness in the presentation, we report results for two window sizes. In order to capture more information, we need larger window sizes for small datasets (e.g. SUBJECTIVITY). Also, since for the baseline methods (TF, TF binary, TF-IDF, w2v, TF-IDF-w2v) there is no notion of window size, the results for $w = \{2, 3\}$ are the same. We have also examined several centrality criteria (using both undirected and directed graphs); undirected giving better results.

Comparing TF to the graph-based ones, namely TW (degree), in almost all cases TW gives higher F1 and accuracy results. Similar observations can be made in the case where the IDF penalization is applied. In most of the datasets, the TW-IDF (degree) scheme performs quite well. The interesting point here, which is confirmed by the related literature (Lan et al., 2005), is that TF-IDF is in general inferior to TF in TC. However, when

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

Table 2: Macro-F1 and accuracy for window size w. Bold shows the best performance on each window size and blue the best overall on each dataset. * indicates statistical significance of improvement over TF at $p < 0.05$ using micro sign test. MAX and SUM state the best nominator for ICW in Eq. (3).

	20NG (MAX)				IMDB (SUM)				SUBJECTIVITY (MAX)			
Methods	$w=3$		$w=4$		$w=2$		$w=3$		$w=6$		$w=7$	
	F1	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**
TF	80.88	81.55	-		84.23	84.24	-		88.42	88.43	-	
w2v	74.43	75.75	-		82.57	82.57	-		87.67	87.67	-	
TF-binary (ngrams)	81.64	82.11*	-		83.02	83.03	-		87.51	87.51	-	
TW (degree)	82.37	83.00*	82.21	82.83*	84.82	84.84	84.67	84.69	88.33	88.33	89.00	89.00*
TW (w2v)	81.88	82.51*	82.21	82.87*	84.66	84.69	84.52	84.54	87.75	87.57	87.66	87.67
TF-IDF	82.44	83.01*	-		83.33	83.33	-		89.06	89.06*	-	
TF-IDF-w2v	82.52	83.09*	-		82.87	82.87	-		89.91	89.91*	-	
TW-IDF (degree)	84.75	85.47*	84.80	85.46*	82.86	82.87	83.02	83.03	89.33	89.34*	89.33	89.34*
TW-IDF (w2v)	84.66	85.32	84.46	85.13	83.47	83.48	83.31	83.33	86.42	86.42	86.51	86.51
TW-ICW (deg, deg)	85.24	85.80*	**85.41**	**86.05***	84.98	85.00	85.13	85.15	89.30	89.31*	89.61	89.61*
TW-ICW (w2v)	**85.33**	**85.93***	85.29	85.90*	85.12	85.15	84.82	84.84	89.61	89.61*	87.30	87.30
TW-ICW-LW (deg)	85.01	85.66*	85.02	85.66*	85.73	85.75	85.28	85.30	**90.12**	**90.13***	**90.27**	**90.28***
TW-ICW-LW (w2v)	82.56	83.11*	82.24	82.81*	85.29	85.30	84.39	84.39	87.70	87.70	87.70	87.70
TW-ICW-LW (pgr)	83.92	84.66	83.80	84.54	84.97	85.00	85.73	85.75	86.60	86.60	86.45	86.45
TW-ICW-LW (cl)	84.61	85.22	84.71	85.27	**87.27**	**87.27***	**86.06**	**86.06**	89.97	89.97*	90.09	90.10*
	AMAZON (MAX)				WEBKB (SUM)				REUTERS (MAX)			
Methods	$w=2$		$w=3$		$w=2$		$w=3$		$w=2$		$w=3$	
	F1	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**
TF	80.68	80.68	-		90.31	91.91	-		91.51	96.34	-	
w2v	79.05	79.05	-		84.54	86.58	-		91.35	96.84	-	
TF-binary (ngrams)	79.84	79.84	-		91.22	92.85	-		86.33	95.34	-	
TW (degree)	80.07	80.07			91.69	92.64	91.45	92.49	93.58	97.53*	93.08	97.25*
TW (w2v)	80.07	80.07	79.54	79.54	91.70	92.64	91.00	92.06	93.09	97.35*	93.43	97.25*
TF-IDF	80.26	80.26	-		87.79	89.89	-		91.89	96.71	-	
TF-IDF-w2v	80.49	80.49	-		88.18	90.18	-		91.33	96.80	-	
TW-IDF (degree)	81.47	81.47*	81.55	81.55*	90.38	91.70	90.47	91.84	93.80	97.30*	93.13	97.35*
TW-IDF (w2v)	79.61	79.62	77.60	77.61	90.81	92.20	90.60	91.91	93.38	97.44*	**93.87**	**97.44***
TW-ICW (deg, deg)	82.08	82.08*	82.02	82.02*	91.72	92.78	91.42	92.49	92.91	97.35	93.59	97.39*
TW-ICW (w2v)	80.86	80.87*	78.82	78.82	91.58	92.64	91.84	92.85	93.57	97.30*	92.96	97.25
TW-ICW-LW (deg)	82.72	82.72*	82.91	82.91*	91.86	92.92	91.95	92.92	**93.88**	**97.53***	93.48	97.35*
TW-ICW-LW (w2v)	80.56	80.56	78.32	78.33	90.74	91.99	90.01	91.34	92.51	96.89	92.14	96.98
TW-ICW-LW (pgr)	82.23	82.23*	82.46	82.46*	91.18	92.20	92.23	93.07	93.38	97.35*	93.37	97.35*
TW-ICW-LW (cl)	**82.90**	**82.91***	**83.02**	**83.03***	**92.72**	**93.57***	**92.86**	**93.57***	93.12	97.25	92.87	97.21

the IDF penalization factor is applied on the TW term-to-document weighting, a powerful mechanism is derived. In the case of purely graph-based schemes, we can observe that some of them produce very good classification results. In almost all cases, TW-ICW-LW (degree or closeness) achieve the best performance.

Significant improvement is observed by adding the w2v similarities as weights in the document, collection level and label graphs in almost all datasets. In fact, we have obtained better results in 20NG (TW-ICW (w2v)), WebKB (TW-ICW (w2v)) and Reuters (TW-IDF(w2v)), by boosting semantically close words in the document level and penalizing them in the collection level.

TF n-gram binary scheme (TF binary) has also been examined, i.e., all the possible n-grams of the collection with binary weights (up to 6-grams in our experiments). For comparison reasons, the size of the unigram feature space considered by our framework is equal to the unique terms in the collections and much smaller compared to the $n-$grams ones. Moreover, graph-based weighting is able to outperform TF (binary) in all datasets.

We clearly see that by fusing document, collection and label graphs we obtain the best results in almost in 5 out of 6 datasets. Label graphs information consist a powerful weighting method, when combined with our proposed collection level graph approach. Adding word2vec similarities as weights, when label graphs are used, does not improve the accuracy. This implies that important

	20NG	IMDB	SUBJ.	AMAZON	WEBKB	REUTERS
CNN (no w2v, 20 ep.) (Kim, 2014)	83.19	74.09	88.16	80.68	88.17	94.75
FastText (100 ep.) (Joulin et al., 2017)	79.70	84.70	88.60	79.50	92.60	97.00
TextRank (Mihalcea and Tarau, 2004)	82.56	83.33	84.78	80.49	92.27	97.35
Word Attraction (Wang et al., 2015)	61.24	70.75	86.60	78.29	79.46	91.34
TW-CRC (Shanavas et al., 2016)	85.35	85.15	89.28	81.13	92.71	97.39
TW-ICW-LW (**ours**)	**86.05**	**87.27**	**90.28**	**83.03**	**93.57**	**97.53**

Table 3: Comparison in accuracy(%) to state-of-the-art deep learning and graph-based approaches.

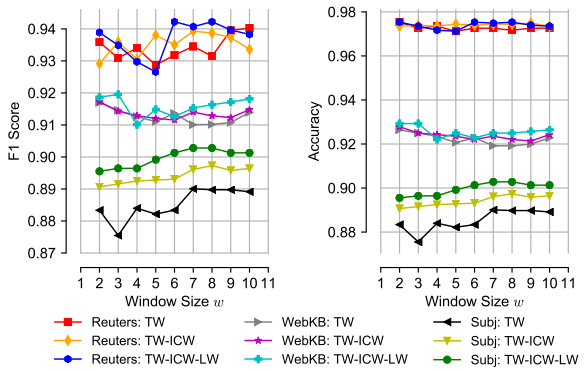

Figure 3: F1 score (left) and accuracy (right) of TW, TW-ICW and TW-ICW-LW (all degree) on REUTERS, WEBKB and SUBJECTIVITY, for window size $w = \{2, \ldots, 10\}$.

terms concerning different labels can be close in the word vector space. Choosing closeness in the document level GoW yields the best performance in 3 datasets. Closeness can only have an affect in larger document lengths and when used along with label graphs.

To further investigate the effectiveness of our approach, we have compared our results with current state-of-the-art graph-based and non graph-based methods. In Table 3 we compare against CNN for text classification ,without pre-trained word vectors (Kim, 2014), FastText (Joulin et al., 2017), TextRank (Mihalcea and Tarau, 2004), Word Attraction weights based on word2vec similarities (Wang et al., 2015) and Supervised Term Weighting (TW-CRC) by Shanavas et al. (2016). Our work produces comparable to state-of-the-art results. Since the implementation of most models is our own, their performance is not optimal.

Selecting the window size w is also important. As we observed, the maximum accuracy is achieved while using small window sizes. In any case, even if larger values of w were able to get slightly better results, a smaller window size would be preferable, due to the overall overhead that could be introduced (increase of the density of the graph). Figure 3 depicts the F1 score and accuracy on the WEBKB, REUTERS and SUBJECTIVITY datasets, using the TW, TW-ICW and TW-ICW-LW(deg) schemes for various window sizes. We notice also that larger sliding windows are only improving accuracy in datasets with small document length (e.g. SUBJECTIVITY).

6 Conclusion & Future Work

In this paper, we proposed a graph-based framework for TC. By treating the term weighting task as a node ranking problem of interconnected features defined by a graph, we were able to determine the importance of a term using node centrality criteria. Building on this formulation, we introduced simple-yet-effective weighting schemes at the collection and label level, in order to penalize globally important terms (as analogous to "globally frequent terms") and reward locally important terms respectively. We also incorporate additional word-embedding information as weights in the graph-based representations.

Our proposed methods could also be applied in IR. In fact, document-level graph-based term weighting has already been applied there, so it would be interesting to examine the performance of the proposed collection-level (ICW) penalization mechanism. In the unsupervised scenario, where label information is not available, community detection algorithms may be applied to identify clusters of words or documents in collection graphs. Graph-based representations of text could also be fitted into deep learning architectures following the idea of Lei et al. (2015). Lastly, one could examine a *Graph-of-documents* approach, in which we create a graph, where nodes represent documents and edges correspond to similarity between them. In this case, graph kernels could be utilized for graph comparison and/or Word Mover's distance (Kusner et al., 2015) between two documents as weights.

References

Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. 1999. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.* 3:1137–1155.

Roi Blanco and Christina Lioma. 2012. Graph-based term weighting for information retrieval. *Inf. Retr.* 15(1):54–92.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Association for Computational Linguistics*. Prague, Czech Republic.

Phil Blunsom, Edward Grefenstette, and Nal Kalchbrenner. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics.

Florian Boudin. 2013. A comparison of centrality measures for graph-based keyphrase extraction. In *IJCNLP*. pages 834–838.

Franca Debole and Fabrizio Sebastiani. 2004. Supervised term weighting for automated text categorization. In *Text mining and its applications*, Springer, pages 81–97.

Mukund Deshpande, Michihiro Kuramochi, Nikil Wale, and George Karypis. 2005. Frequent substructure-based approaches for classifying chemical compounds. *IEEE Trans. on Knowl. and Data Eng.* 17(8):1036–1050.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.* 22(1):457–479.

Samer Hassan, Rada Mihalcea, and Carmen Banea. 2007. Random-walk term weighting for improved text classification. In *ICSC*. pages 242–249.

Dat Huynh, Dat Tran, Wanli Ma, and Dharmendra Sharma. 2011. A new term ranking method based on relation extraction and graph model for text classification. In *Proceedings of the Thirty-Fourth Australasian Computer Science Conference-Volume 113*. Australian Computer Society, Inc., pages 145–152.

Chuntao Jiang, Frans Coenen, Robert Sanderson, and Michele Zito. 2010. Text classification using graph mining-based feature extraction. *Knowl.-Based Syst.* 23(4):302–308.

Thorsten Joachims. 1998. Text categorization with suport vector machines: Learning with many relevant features. In *ECML*. pages 137–142.

Rie Johnson and Tong Zhang. 2015. Effective use of word order for text categorization with convolutional neural networks. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*. pages 103–112.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. pages 427–431.

Sang-Bum Kim, Kyoung-Soo Han, Hae-Chang Rim, and Sung-Hyon Myaeng. 2006. Some effective techniques for naive bayes text classification. *IEEE Trans. Knowl. Data Eng.* 18(11):1457–1466.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International Conference on Machine Learning*. pages 957–966.

Shibamouli Lahiri, Sagnik Ray Choudhury, and Cornelia Caragea. 2014. Keyword and keyphrase extraction using centrality measures on collocation networks. *CoRR* .

Man Lan, Chew-Lim Tan, Hwee-Boon Low, and Sam-Yuan Sung. 2005. A comprehensive comparative study on term weighting schemes for text categorization with support vector machines. In *WWW*. pages 1032–1033.

Yann LeCun, Yoshua Bengio, et al. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* 3361(10):1995.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2015. Molding cnns for text: non-linear, non-consecutive convolutions. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* .

Edda Leopold and Jörg Kindermann. 2002. Text categorization with support vector machines. How to represent texts in input space? *Mach. Learn.* 46(1-3).

Marina Litvak and Mark Last. 2008. Graph-based keyword extraction for single-document summarization. In *MMIES*. pages 17–24.

Fragkiskos D. Malliaros and Konstantinos Skianis. 2015. Graph-based term weighting for text categorization. In *Proceedings of ASONAM*. pages 1473–1479.

Justin Martineau and Tim Finin. 2009. Delta tfidf: An improved feature space for sentiment analysis. In *ICWSM*.

Andrew McCallum and Kamal Nigam. 1998. A comparison of event models for naive bayes text classification. In *AAAI: Proceedings of the Workshop on Learning for Text Categorization*. pages 41–48.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *EMNLP*. pages 404–411.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Mark Newman. 2010. *Networks: An Introduction*. Oxford University Press, Inc.

Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Mach. Learn.* 39(2-3):103–134.

Giannis Nikolentzos, Polykarpos Meladianos, François Rousseau, Yannis Stavrakas, and Michalis Vazirgiannis. 2017. Shortest-path graph kernels for document similarity. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pages 1890–1900.

Georgios Paltoglou and Mike Thelwall. 2010. A study of information retrieval weighting schemes for sentiment analysis. In *ACL*. pages 1386–1395.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*.

Stephen Robertson. 2004. Understanding inverse document frequency: On theoretical arguments for idf. *Journal of Documentation* 60:2004.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp* 109:109.

François Rousseau and Michalis Vazirgiannis. 2013. Graph-of-word and TW-IDF: new approach to ad hoc IR. In *CIKM*. pages 59–68.

François Rousseau and Michalis Vazirgiannis. 2015. Main core retention on graph-of-words for single-document keyword extraction. In *ECIR*. pages 382–393.

Franois Rousseau, Emmanouil Kiagias, and Michalis Vazirgiannis. 2015. Text categorization as a graph classification problem. In *ACL*.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.* 24(5):513–523.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM Comput. Surv.* 34(1):1–47.

Niloofer Shanavas, Hui Wang, Zhiwei Lin, and Glenn Hawe. 2016. Centrality-based approach for supervised term weighting. In *Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference on*. IEEE, pages 1261–1268.

Masumi Shirakawa, Takahiro Hara, and Shojiro Nishio. 2015. N-gram IDF: A global term weighting scheme based on information distance.

Amit Singhal, Chris Buckley, and Mandar Mitra. 1996. Pivoted document length normalization. In *SIGIR*. pages 21–29.

Rui Wang, Wei Liu, and Chris McDonald. 2015. Corpus-independent generic keyphrase extraction using word embedding vectors.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28*, Curran Associates, Inc., pages 649–657.