ISSN 2007-9737

Text Analysis Using Different Graph-Based Representations

Esteban Castillo[1], Ofelia Cervantes[1], Darnes Vilariño[2]

[1] Universidad de las Américas Puebla,
Department of Computer Science, Electronics and Mechatronics,
Mexico

[2] Benemérita Universidad Autónoma de Puebla,
Faculty of Computer Science,
Mexico

{esteban.castillojz, ofelia.cervantes}@udlap.mx, darnes@cs.buap.mx

Abstract. This paper presents an overview of different graph-based representations proposed to solve text classification tasks. The core of this manuscript is to highlight the importance of enriched/non-enriched co-occurrence graphs as an alternative to traditional features representation models like vector representation, where most of the time these models can not map all the richness of text documents that comes from the web (social media, blogs, personal web pages, news, etc). For each text classification task the type of graph created as well as the benefits of using it are presented and discussed. In specific, the type of features/patterns extracted, the implemented classification/similarity methods and the results obtained in datasets are explained. The theoretical and practical implications of using co-occurrence graphs are also discussed, pointing out the contributions and challenges of modeling text document as graphs.

Keywords. Text modeling, graph-based representation, co-occurrence graphs, text classification, feature-vector approach, graph similarity approach.

1 Introduction

In recent years, the web has become an essential resource for obtaining information associated with any topic or domain. The amount of text produced by interactions on social media, blogs, URLs, etc., has made essential to use advanced techniques to be able to understand and obtain valuable patterns from these large volumes of data [10].

Considering the complexity and richness of the web information, the use of graph techniques for mining texts is a growing area of study [62]. Its aim is to discover novel and insightful knowledge from data that is represented as a graph. The use of this kind of graph techniques on text documents has a wide range of applications [21] in areas like social science, homeland security, finance, healthcare, climatology, web analysis, linguistics, etc. This is mainly because graphs are perceived as a natural way to represent the connections among data, as well as the increasing number of tools available to handle these types of structures [78, 45].

This paper presents the results of experimenting with co-occurrence graph-based representations over multiple text classification tasks. The contribution relies on the analysis of the relevance of these representations and the theoretical and practical implications with regard to the state of the art of graph-based representations [21, 49, 62]. Our hypothesis is that co-occurrence graphs are a good alternative to represent text documents, keeping in mind that graphs can map different levels of language into a richer data structure, which otherwise could not be used in an easy and integrated manner.

The motivation of this analysis is to show how co-occurrence graphs can be used to represent text documents in a practical manner independently of the text classification task and how this kind of representation can be a valuable asset to extract features/patterns that, due to its structural simplicity, other representations cannot show (like a vector representation [46]).

The remainder of this paper is structured as follows: in Section 2 we present existing approaches (related work) to deal with different text classification tasks using graphs. Section 3 provides details on design and implementation of proposed graphs used in different text classification tasks. In Section 4 a discussion about the relevance of graphs in these tasks is presented. Finally, implications and conclusions derived from this work so far are presented in Section 6.

2 Related Work

The goal underlying this paper is to highlight the importance of a richer data structure like graphs on different text classification tasks. Therefore, related work should be seen from the perspective of using graphs in trending text problems [61, 52] such as: Authorship Attribution, Authorship Verification, Author Profiling, and Sentiment Analysis. For each one of these problems, the task definition and recent works related in the context of graphs are presented and discussed.

2.1 Authorship Attribution

Authorship Attribution is the task of identifying the author of a given text document from a set of known authors, each one with a set of known documents [39]. This task is called a "close class task" because it is necessary to choose the author from a set of known authors which means that an unknown document must belong to one of them. Another point to take in mind is that each known author has many documents, each one with a great amount of textual information such as books, reviews, blogs, etc. So, when considering these facts, most of the state of the art see this task as a reasonably easy problem to address [72].

In this context, there are different works that used graphs to obtain the author's writing style. In [20, 32] different graph representations that integrate linguistic levels of text are proposed, the idea in these papers is to obtain text features by analyzing the lexical, syntactic and morphological relationships of texts on a richer data structure.

Other papers focused on the use of co-occurrence graphs [49] to represent the syntactic relationship of words to extract features based on recurrent interconnection of word patterns [48] (called motifs), extraction of topological properties in graphs like clustering coefficient [42] or to find the similarity among author's graphs [68]. Something similar to the extraction of recurrent information occurs in [25] where a graph based on revised text content (Wikis) is used to obtain the author who writes a document for the first time.

In addition to the use of graphs to extract relevant linguistic features or topological elements, other papers use graphs as classification algorithms to find the author of a document. In [44] a neural network based on grammatical structure and vocabulary of text documents is proposed, the main objective of this paper is to find vocabulary-based cues to determine the author of a document. On [75] a neural network that combines lexical and syntactic features as input layers is also used to classify an author.

2.2 Authorship Verification

The Authorship Verification task consists of determining whether or not an unknown document was written by a particular author, given some samples of the author's writing style [73]. This task is called an "open class task" because, unlike Authorship Attribution, it is necessary to verify if an author with a limited set of text documents wrote a text. One of the major differences with Authorship Attribution task is that there are no other authors to compare their writing style and the only author has a limited set of documents with fewer textual information, for instance, all social media text documents. This task is considered by the state of the art as a more difficult problem to address compared with the traditional Authorship Attribution task [41].

In the context of this task, there are a limited number of papers that uses a graph-based approach to solve it. In [80] a graph representation that captures lexical syntactic features is described, where the goal is to obtain representative words by means of graph mining tools [63] that can characterize the author's writing style.

Other papers [66, 6, 5] use neural networks and deep learning techniques to verify the authorship of

a document considering as input: lexical, syntactic, and semantic features.

2.3 Author Profiling

Authorship Analysis tasks[1] mentioned above propose finding the author of a text document considering their own writing style, in other words analyzing the individual's style. On the other hand, Author Profiling considers demographic aspects of text documents, such as age, gender or personality traits of authors.

This is used with the objective of analyzing how texts are shared by people and the type of text patterns observed in these interactions [30, 55]. Author Profiling is a task of growing importance [57] in applications of text forensics, security, and marketing. This makes this task a relevant problem considering a more traditional text classification task. There are some papers that deal with this task using graphs.

The majority proposed a supervised classification approach with a co-occurrence graph to detect age and gender. In [56] an emotion-labelled graph is proposed, where the objective is modelling the way people use emotional words to capture topological elements of graphs. In [4] a star topology graph is proposed to extract recurrent words without any kind of enrichment to be used as features. Other papers [58, 23] propose using a neural network to learn relevant features instead of searching for recurrent elements or the way words interact in text documents.

Although some papers extract the age and gender of people who write a text, there are few papers that extract other demographic aspects. For instance, in [79, 31] a graph-based representation is proposed to extract lexical, syntactic, and morphological features associated to different personality traits like *extroverted*, *stable*, *open*, *etc*.

[1] Authorship Attribution and Verification.

2.4 Sentiment Analysis

The Sentiment Analysis task consists of determining the polarity[2]/sentiment associated to a given text document considering a two-point scale (positive and negative) or a five-point scale (very positive, positive, neutral, negative and very negative) text classification [54]. This task has recently become an attractive research topic [52], particularly due to the growth of social media interactions on the web in platforms such as Twitter[3] or Facebook[4].

Some contributions associated to this task proposed a graph-based representation. In [83], a sentence-level graph is used to capture the relationship among opinion expressions and word polarities with the aim of characterizing the overall sentiment of a text document. In [82, 50] different graph representations are used to analyze the relationship between word senses and subjective/objective words to find the sentiment of a text. Other papers [29, 3] discussed the use of n-gram[5] graphs to capture different syntactic patterns associated to the polarity of texts.

Although most approaches proposed to extract features of a graph to characterize the sentiment of a text document, in some papers [70, 69, 77] the use of neural networks techniques are employed to classify a sentiment taking as input character and sentence level information on short text documents.

2.5 Summary

Considering the different methods used to represent text documents as graphs exposed above, it can be observed that the majority of them apply a supervised learning approach based on the extraction of relevant features from graph representations. In this context, co-occurrence graphs (simple or enriched), n-gram graphs, and other kind of graphs that use the lexical-syntactic information of texts have proven to be successful structures in the classification process of text tasks. The type of lexical features extracted using these

[2] The attitude of a writer with respect to some topic.
[3] https://twitter.com/
[4] https://www.facebook.com/
[5] An n-gram is a subsequence of one or more text tokens.

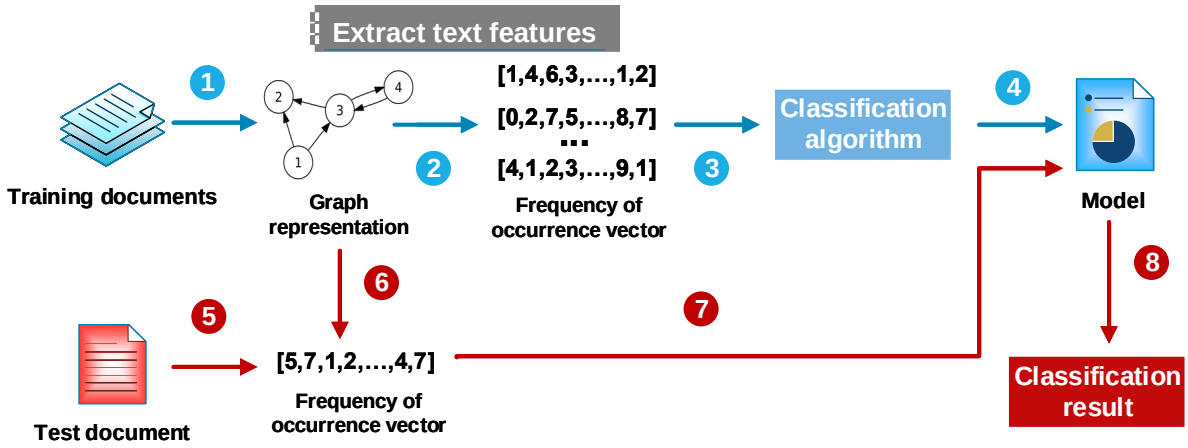

Fig. 1. Feature-vector approach

supervised learning approaches are mostly words with a high influence in sentences taking into account the topological structure of graphs. In the case of syntactic features, other components such as PoS tags [7], n-grams [38], and chunk phrases [8] can be extracted.

The alternative to the feature extraction methods based on graphs is the use of these structures as classification tools. Most of the state of the art approaches use neural networks, where different lexical and syntactic elements of texts are used as the input layer in a neural network. The kind of features used as input to the network are normally words or sentences which are ranked according to different techniques like the bag-of-words or TF-IDF [47].

In the case of Authorship Attribution task, the use of co-occurrence graphs with techniques to find recurrent patterns in text documents have proven to be effective, especially when having multiple examples of each author's writing style. For the Authorship Verification task, there are few works that employed graphs but, as can be seen in the related work section, the use of neural networks and co-occurrence graphs are a good alternative.

For the Sentiment Analysis and the Author Profiling tasks, the use of graphs that map different linguistic levels of text and neural networks have shown a competitive performance, although not at the same level of the authorship tasks. This could be because this tasks use social media pieces of text as full texts, which means that text could be unstructured, smaller and usually contains slang and gender-specific terminology [22, 43].

3 Proposed Graphs

In this section, different graph-based representations are presented and discussed. These representations propose the use of enriched/non-enriched co-occurence graphs in different text classification tasks. In this context, two main supervised learning approaches are proposed to classify text documents using graphs: one based on extracting feature vectors on a traditional classification method [34, 51] and another one based on calculating the similarity between graphs. For each classification task, the proposed graph as well as the benefits obtained with that representation and approach are described.

3.1 Feature-Vector Approach

Figure 1 shows the steps involved in a text classification process where text features are extracted from a graph using a vector representation. First, training text documents are used to

create a graph representation (steps 1-2). Then, a feature set is extracted and represented as a frequency of occurrence vector [67] for each training document (step 3). These vectors are then used alongside with a classification algorithm to create a model (step 4). Finally, a test document is also represented as a frequency of occurrence vector[6] (steps 5-6) and is tested using the previously created model with the aim of obtaining a classification result[7] (steps 7-8).

3.1.1 Authorship Attribution Graph

In the Authorship Attribution task, a directed graph representation (digraph) based on a star topology is proposed [19]. The purpose of using this form of representation is to find relevant words in text documents, by the appearance of recurrent words and PoS tags[8] in the syntactic structure of text sentences. Formally, the proposed graph is represented by $G = (V, E, L_V, L_E)$,where:

1. $V = \{v_0, v_1, ..., v_n\}$ is a finite set of vertices that consists of the words contained in one text document.
2. The vertex v_o is labeled as init and is considered as the central vertex in the star topology.
3. $E \subseteq V \times V$ is the finite set of edges which represent:
   - An edge between two vertices if their corresponding lexical units co-occur within a window of two words in the text (at least once).
   - An edge between init vertex and the other vertices/words in the graph.
4. $L_V = L_{V1} \bigcup L_{V2}$, the label set of V, where:
   - $L_{V1} = \{init\}$
   - $L_{V2} = \{etq : etq \in words\}$
5. $L_E = L_{E1} \bigcup L_{E2}$, the label set of E. Where PoS tags are used:
   - $L_{E1} = \{word\}$
   - $L_{E2} = \{etq_2 : etq_2 \in PoS\ tags\}$

As an example of this graph, consider the following sentence ζ extracted from a text T: "It may be only the second qualifier on the long road leading to the 1998 World Cup", which after a preprocessing[9] would be as follows: "second qualifier long road leading 1998 world cup". Based on the proposed representation, preprocessed sentence ζ can be mapped to a graph as shown in Figure 2.

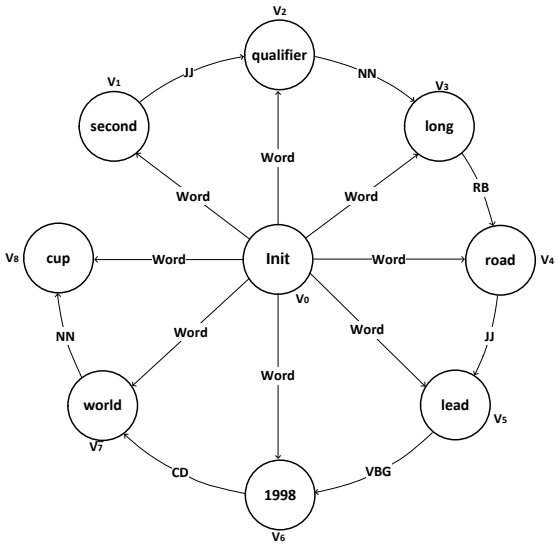

Fig. 2. Example of a star topology graph

The benefits associated with a star topology graph in the Authorship Attribution are the following:

- The creation of a graph per text document permits searching for recurring patterns in all documents written by an author.
- The use of co-occurrence windows of two words allows to map the natural relationship of words. This in turn facilitates the extraction of lexical-syntactic features that depend on

[6] Using the same features obtained in the training phase.
[7] Author or sentiment associated to a text document.
[8] The syntactic role of words in the text.
[9] this task includes lowercase all words in the texts and elimination of stopwords, punctuation symbols and all the elements that are not part of the ASCII encoding.

the interaction of words in the author's writing style.

— The addition of an **init** vertex and the edge direction allows to iterate over multiple paths in the graph using graph mining algorithms like SUBDUE [63].

— The use of PoS tags helps graph mining algorithms to differentiate the relationships between words in the graph structure.

— Words extracted from the graph can be used to obtain semantic features like synonyms, hyponyms, hyperonyms, etc [9]. Particularly, hyperonyms allow to concentrate a family of words semantically related into a single class, thus improving the proposed extracted set features.

— The classification results [19] obtained for an English dataset using this graph (accuracy of 79.2%) are near to the best result reported so far [24] (accuracy of 86.4%), which is an indicative that a star topology graph is a simple but very effective way to obtain good features in the Authorship Attribution task.

3.1.2 Sentiment Analysis Graph

For the Sentiment Analysis task, a non-directed graph representation based on a co-occurrence graph is proposed [16]. The goal behind this type of graph is to use centrality measures [26, 10] for obtaining relevant word features through the interaction/relevance of words on the lexical-syntactic structure of sentences. Formally, the proposed graph is represented by $G = (V, E, L_V, L_E)$, where:

1. $V = \{v_1, ..., v_n\}$ is a finite set of vertices that consists of all nouns, verbs and adjectives contained in one or several texts.
2. $E \subseteq V \times V$ is the finite set of edges which represent that two vertices are connected if their corresponding lexical units co-occur within a window of maximum three words in the text at least once.
3. L_V is the label set of V, where $L_V = \{etq : etq \in (nouns \bigcup verbs \bigcup adjectives)\}$
4. L_E is the label set of E, which consists of the number of times that two vertices co-occur in a text window of three words.

As an example of this graph, consider the following sentence ζ extracted from a text T: "They may have a SuperBowl in Dallas, but Dallas ain't winning a SuperBowl. Not with that quarterback and owner, they are really bad.", which after a preprocessing stage[10] would be as follows: "may have SuperBowl Dallas Dallas ain't winning SuperBowl quarterback owner are bad". Based on the proposed representation, preprocessed sentence ζ can be mapped to the co-ocurrence graph shown in Figure 3.

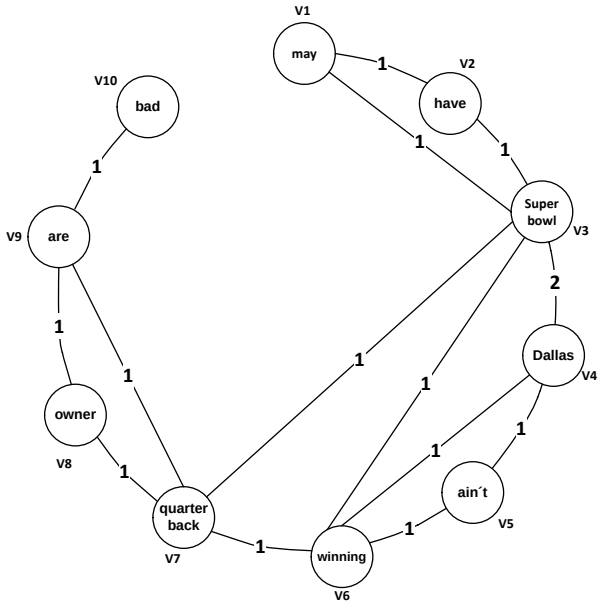

Fig. 3. Example of a co-occurrence graph with a window of three words.

The benefits associated with a co-occurrence graph in the Sentiment Analysis are the following:

[10] this task includes lowercase all words in the texts and elimination of punctuation symbols and all the elements that are not part of the ASCII encoding.

— The creation of a graph per sentiment (negative, neutral and positive) along with word co-occurrence, permits to extract text features strongly associated to each sentiment.

— The use of windows of three words allows to represent relationships of words that are together/separated in the syntactic sequence of texts. This presents an advantage for detecting relevant words where there is no syntactic order (usually the case of the Sentiment Analysis task).

— In the case of the Sentiment Analysis task, a weighted graph permits extracting features in a more effective way considering the use of centrality measures and the lack of syntactic order on texts.

— The use of degree centrality, which is defined as the number of edges incident upon a vertex in the graph, helps finding topologically representative words.

— The use of closeness centrality, defined as the average sum of the shortest paths from one vertex to the others in the graph, is an effective way to obtain the most accessible words in the graph, which are also syntactically relevant.

— The classification results [16] obtained for an English dataset using this graph (score 42.10) are not to far away to the best result reported so far [33] (score 64.84). This indicates that the use of a co-occurrence graph could be a good alternative to the traditional methods in the Sentiment Analysis task.

3.1.3 Authorship Verification Graph

In the Authorship Verification task an enriched co-occurrence graph (directed graph) is proposed [17]. The advantage of this type of graph compared to others is the creation of new edges between vertices that represent how phrases are used to create sentences (chunk tags) and the reinterpretation of centrality measures [26, 10] to extract word features based on the relevance/interaction of words. Formally, the proposed graph is represented by $G = (V, E, L_V, L_E)$, where:

1. $V = \{v_0, v_1, ..., v_n\}$ is a finite set of vertices that consists of the words contained in one or several texts.

2. $E \subseteq V \times V$ is the finite set of edges which represents the following connections:

   — Two vertices are connected by means of the sequence of the text if their corresponding lexical units co-occur within a window of two words in the text at least once (L_{E1} label).

   — Two vertices are connected if they are at the beginning and end of a phrase (L_{E2} label).

   — Two vertices are connected if they are at the beginning and end of a sentence (L_{E3} label).

3. L_V is the label set of V, where $L_V = \{etq : etq \in (word - PoS\, tag)\}$

4. $L_E = L_{E1} \bigcup L_{E2} \bigcup L_{E3}$, the label set of E. Where IOB prefixes[11], PoS tags, chunk tags[12] and a "sentence" label are used:

   — $L_{E1} = \{etq : etq \in (\underbrace{IOB - PoS}_{\substack{source \\ vertex}} / \underbrace{IOB - PoS}_{\substack{target \\ vertex}})\}$

   — $L_{E2} = \{etq : etq \in (chunktag - \#words)\}$

   — $L_{E3} = \{etq : etq \in (sentence - \#words)\}$

As an example, consider the following sentence ζ extracted from a text T: "I shot an Elephant in my pajamas.", which after a pre-processing stage[10] would be as follows: "I shot an elephant in my pajamas". Based on the proposed representation, preprocessed sentence ζ can be mapped to the proposed graph shown in Figure 4.

The benefits associated with using an enriched co-occurrence graph for the Authorship verification task are the following:

— The creation of a graph per author helps obtaining features strongly associated with their writing style.

[11] The position of words in phrases.

[12] The type of phrase on a text.

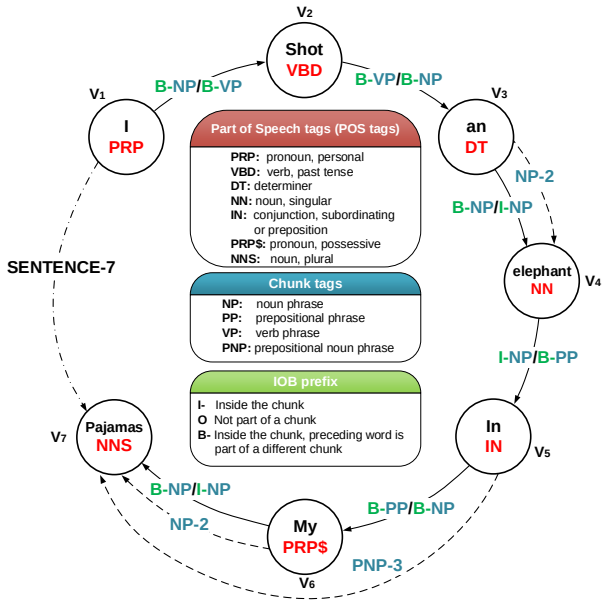

Fig. 4. Example of an Enriched co-occurrence Graph with a Window of two Words.

— The use of co-occurrence edges L_{E1} and chunk edges L_{E2} allows to map the natural relationship of words and the internal structure of sentences. This in turn helps obtaining relevant/important words using a reinterpretation of the most important centrality measures.

— In the case of degree centrality, vertices with a high rank are used to obtain highly interactive words, regardless of their syntactic relevance in texts. These words are used to obtain collocations[13].

— For the closeness centrality, the words that are reachable in the minimum number of steps are used to obtain relevant phrases associated to the texts (chunk phrases).

— The words with the highest betweenness centrality are employed to extract n-grams, considering that this centrality obtains words that generally act as a bridge to reach other words.

— The eigenvector centrality allows to obtain words that are important for the number connections they have (in a recursive way) instead of their own prestige.

— The classification results [17] obtained for the English-essays dataset subset (for instance) using this graph (score f0.46) are better than traditional approaches [18, 74] and near to the best result reported so far [27] (score 0.51). This is an indication that the use of an enriched co-occurrence graph can be used to verify the author of a given unknown document.

3.2 Similarity Approach

Figure 5 shows the steps involved in a text classification process considering the use of a similarity approach instead of a feature-vector approach. First, a graph representation is created over training document(s) (step 1), then a graph representation is also created for each test document (step 2). A graph similarity function is then applied taking as input the training and test graphs (step 3). Finally, the similarity between both graphs is used alongside with a heuristic to obtain a classification result[14] (step 4).

3.2.1 Author Verification Graph

For the Authorship Verification task, a direct co-occurrence graph is proposed [14]. The difference between this graph and the graph used to extract features for the same task, is that an edge weighted scheme based on the natural relationships of words is used in a context that considers the similarity between graphs. Formally, the proposed graph is represented by $G = (V, E, L_V, L_E)$, where:

1. $V = \{v_1, ..., v_n\}$ is a finite set of vertices that represents the words contained in one or many texts.

[13] Pairs of words that always appear together in text documents.

[14] A sentiment or a demographic aspect of an author.

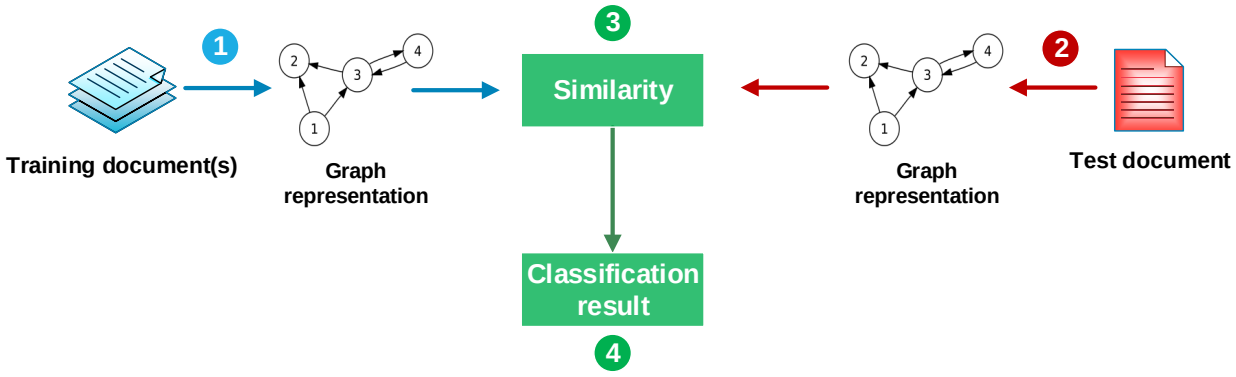

Fig. 5. Similarity approach

2. $E \subseteq V \times V$ is the finite set of edges which represent that two vertices are connected if their corresponding lexical units co-occur within a window of up to two words in the text at least once.
3. L_V is the label set of V, where $L_V = \{etq : etq \in words\}$
4. L_E is the label set of E, which consists of the number of times that two vertices co-occur in a text window of two words.

As an example of this graph, consider the following sentence ζ extracted from a text T: "The violence on the TV. The article discussed the idea of the amount of violence on the news.", which after a preprocessing stage[10] would be as follows: "the violence on the tv the article discussed the idea of the amount of violence on the news". Based on the proposed representation, preprocessed sentence ζ can be mapped to the co-occurrence graph shown in Figure 6.

The advantages associated to a co-occurrence graph in a graph similarity approach are the following:

— The use of a graph similarity approach helps in considering multiple graph construction scenarios in a straightforward way, which contrasts with supervised learning methods where most of the time the focus is on the extraction of features rather than the representation scheme:

  – One graph per document: for each author and unknown document create a graph representation. Then, the similarity between both is compared.
  – Multiple subgraphs for each document: for each author document, a new graph representation is created. From this major graph, many subgraphs can be obtained with the aim of extracting more examples of the writing style. Then, a graph for an unknown document is created and compared against existing subgraphs.
  – Multiple subgraphs for multiple documents: map all documents associated to an author to a graph representation. Then, obtain the most important subgraphs. Finally, create a graph for an unknown document and compared with existing subgraphs.

— The use of a graph representing the strict syntactic order of words (window of two words) helps obtaining the similarity between two graphs by using a vertex similarity function called dice similarity algorithm [1]. The purpose of this algorithm is to obtain the syntactic neighborhood patterns associated to

a vertex. This, in turn, allows to compare the topological structure of the vertices shared between two graphs.

— The proposed graph representation allows to evaluate the similarity of each unknown graph against an author graph in a faster way, given that the similarity function only evaluates the topological affinities of the vertices shared between both graphs without considering a weighting scheme. This means that dice similarity considers any kind of graph as a non-directed structure.

— The use of a weighting scheme in a graph where there are few samples of the author's writing style (Authorship Verification case) allows the extraction of more text samples (subgraphs) using the edge betweenness algorithm [2]. This algorithm is based on the use of the betweenness centrality. The main idea is that betweenness of the edges connecting two subgraphs is typically high. So, by gradually removing the edges with highest betweenness from a graph, and recalculating edge betweenness after every removal, sooner or later the graph can eventually be broken into smaller components.

— The classification results [14] obtained for an English and Spanish dataset using this graph (accuracy of 0.47 and 0.64) are near to the best results reported so far [74, 27, 40] (accuracy of 0.51 and 0.69). This shows that the combination of a similarity approach and a graph can be used in a very competitive way.

3.2.2 Sentiment Quantification Graph

The Sentiment Quantification task [52] is an extension of the classic Sentiment Analysis task. The objective consists of estimating the overall sentiment distribution of texts documents given a two point scale classification (positive and negative sentiments). In this task, a non-directed graph representation based on a co-occurrence graph is proposed [15]. The objective of this graph is to use the natural relationships of words (with a window of two words) to obtain the similarity of a test

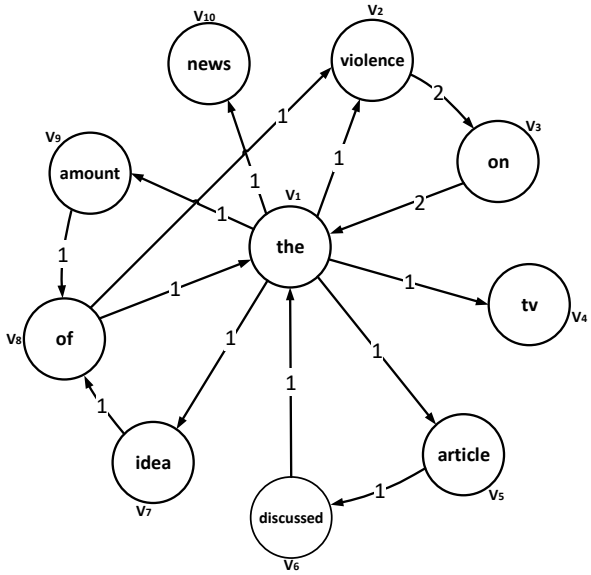

Fig. 6. Example of a co-occurrence Graph with a Window of two Words and Weighted Edges.

graph compared with positive and negative graphs. This graph, in turn, can be used to classify a text document in a novel way. Formally, the proposed graph is represented by $G = (V, E, L_V)$,where:

1. $V = \{v_1, ..., v_n\}$ is a finite set of vertices that consists of the words contained in many texts.
2. $E \subseteq V \times V$ is the finite set of edges which represent that two vertices are connected if their corresponding lexical units co-occur within a window of two words in the text at least once.
3. L_V is the label set of V, where $L_V = \{etq : etq \in words\}$

As an example, consider the following sentence ζ extracted from a text T: "Axel Rose needs to just give up. Now. Not later, not soon, not tomorrow.", which after the preprocessing stage[10] would be as follows: "axel rose needs to just give up now not later not soon not tomorrow". Based on the proposed representation, preprocessed sentence ζ can be mapped to the co-occurrence graph shown in Figure 7.

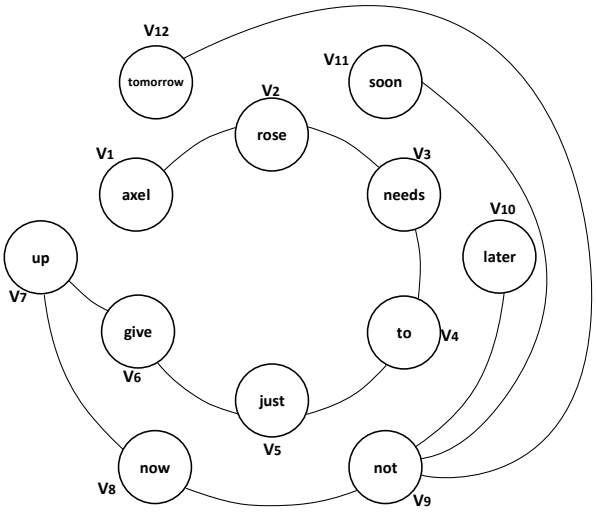

Fig. 7. Example of a co-occurrence graph with a window of two words and non weighted edges

The benefits associated with a co-occurrence graph in the Sentiment Quantification are the following:

— The creation of a graph per sentiment in the training documents and a graph for each text in the test documents, allows to evaluate the similarity of each test document against each sentiment in an intuitive way, rather than searching text features in a supervised learning approach.

— The use of windows of two elements in the graph allows to represent the natural relationship of words. This, in turn, permits to obtain the syntactic neighborhood associated with vertices using a fast function (dice similarity algorithm) that considers these patterns [1] for calculating similarity between graphs.

— Another key aspect of capturing the graph similarity of text documents using the dice similarity algorithm (and especially capturing the similarity of short text documents) is that this similarity function considers only the existence of word relations instead of their intensity (like in the bag-of-words model).

— The classification results [15] obtained for an English dataset using this graph (KLD[15] score **0.261**) are not far to the best results reported so far [76] (KLD score **0.034**). This again show the importance of a similarity approach and in specific of co-occurrence graphs to obtain competitive results regardless of the text classification task.

3.2.3 Author Profiling

For the Author Profiling task, the same graph used in the Sentiment Quantification task (figure 7) is applied in a context that consider the similarity between graphs to obtain demographic aspects (such as age, gender and personality traits) of text documents [13]. The additional benefits of this graph in the profiling task are the following:

— Similar to the Sentiment Quantification task, the creation of a graph for each demographic aspect (age, gender and personality traits) in the training documents and a graph per text in the test documents permits assessing similarity in a simple but effective way.

— The use of windows of two elements in the graph allows to represent the natural relationship of words. This, in turn, permits to obtain the syntactic neighborhood associated to vertices using a fast function (dice similarity algorithm) that considers these patterns [1] to obtain the similarity between graphs.

— The use of the link prediction theory [53, 64] helps to enriched co-occurrence graphs (especially from small text documents) based on gradually creating and adding new edges on a graph considering the interaction and proximity of shared interactions among vertices.

— The proposed graph with the aid of the link prediction theory permits to evaluate the similarity in two novel ways:

[15] Kullback-Leibler Divergence score.

  - Link prediction on demographic graph: Added new edges in the demographic graph during N iterations, in order to evaluate how similar are the edge patterns in the demographic and text graph shared vertices.
  - Link prediction between demographic graph and text graph: Added new edges to both graphs N iterations in order to compare how similar are the edge patterns in the demographic and text graph shared vertices.

— The classification results [13] obtained for an English dataset using this graph (accuracy for age 0.82, gender 0.85 and RMSE[16] for personality traits 0.30) are close to the best results reported [55, 12] (accuracy for age 0.83, gender 0.85 and RMSE for personality traits 0.14). This evidence the relevance of co-occurrence graphs to obtain the demographic aspects of text documents without considering a feature-vector approach and confirms the importance of a similarity approach in some text tasks.

4 Graph Representation Relevance

Different graph-based representations have been proposed to deal with text classification tasks (see Section 3). The majority of methods obtain competitive scores that are below the state of the art best results, given the same training and test corpora. In the case of Authorship Analysis tasks (Attribution and Verification), the obtained scores suggest that co-occurrence graphs (in all modalities) with a great deal of textual information can be used to accurately represent the author's writing style, regardless of using either a feature-vector or a similarity approach.

For the case of Sentiment Analysis tasks, where there is less textual information (e.g. Twitter texts), the scores obtained reveal that a co-occurrence graph with a window of two words following a similarity approach is a better option to obtain the sentiment associated to a text document. However, more experiments are needed in order to improve the obtained results using a feature-vector approach.

In the Author Profiling task we faced a similar effect to the one found for the Sentiment Analysis task. The use of a window of two words in a graph similarity approach helps effectively obtaining the age and gender of authors, but in the case of personality traits, the results are slightly below top results, which highlights the challenge of retrieving this kind of demographic aspects. Moreover, new experiments need to be performed for Author Profiling using a feature-vector approach.

In general, author's experiments show that the use of co-occurrence graphs is a reasonable alternative to the traditional vector representation approach, especially to map the natural and strict relationship of texts in a window of two words and all possible relationships of words in a window of three words. Further experiments will help to show the relevance of other types of enriched/non-enriched co-occurrence graphs in the proposed text tasks.

The use of a feature-vector approach helps to classify an author or sentiment in a unique way. In the case of Authorship Verification and Sentiment Analysis tasks, the use of centrality measures helps to obtain linguistic features (words) that do not depend entirely on stylistic aspects of texts documents. For the Authorship Attribution task, the extraction of recurrent word patterns helps to extract features considering the interaction of words rather than their frequency of occurrence.

In the case of a graph similarity approach, the use of a similarity function allows to find demographic aspects of text or a sentiment in a novel way. For the Author Profiling task, the application of the link prediction theory permits to add new edges to a graph where there is few textual information to characterize the age, gender and personality traits associated to a text. On the other hand, the extraction of subgraphs using edge betweenness permits to obtain more examples of a text document without searching for extra information in the Authorship Verification task. It is important to notice that the use of dice similarity (after link prediction or edge betweenness) enables the extraction of the similarity of shared vertices,

[16] Root Mean Squared error score.

which ultimately leads to the similarity of two graphs.

Finally, created graphs present different improvements compared to the approaches discussed in the related work (see Section 2). This helps illustrating the richness and flexibility of graphs to create representations that map relevant information associated to text documents. Further analysis over the modeling of graphs could lead to find more accurate text representations and text features that will be at the same level or outperform most of the state of the art techniques used for text classification tasks discussed in this paper.

5 Classification Results

In Table 1 the classification results of proposed approaches and the best results reported so far are summarized. The idea is to highlight the importance of presented ideas in contrast to the state of the art text classification results.

6 Conclusions and Future Work

In this paper, different approaches based on graphs to solve text classification tasks have been presented. The results obtained show the relevance of graphs compared with traditional classification approaches that use the same dataset. Considering the theoretical implications of these approaches, the contributions as well as the challenges associated to graphs created by the authors are the following:

— Challenges:

1. Considering the growing amount of information in social media, the creation of proposed graphs will require tools capable of handling this kind of large datasets (Big Data analytics) [81, 45].
2. Most of proposed methods need to be tested in other languages like Spanish, where there is more ambiguity and the syntactic structure of documents is more difficult to extract [28].
3. The use of co-occurrence graphs with windows of more than two words help to extract features where there is no strict syntactic order but there is a need to test the best window length in order to not extract irrelevant features.
4. In order to use centrality measures like closeness and betweenness, the algorithms need to calculate the shortest path among vertices. Therefore, it is necessary to evaluate in which other text classification tasks could be applied these centralities, without affecting performance.
5. It is necessary to explore different options to obtain similarity between graphs considering other types of enriched/non-enriched co-occurrence graphs with different window lengths.

— Contributions:

1. Different graphs representations have been proposed considering the specific features associated with each text classification task. However, these graphs representations can be used as well in the context of other text tasks without any change. (like for example in the Author Profiling task in Section 3.2.3).
2. A graph is generated only once, regardless of the number of training scenarios in a feature-vector or similarity approach. The same applies for text documents in a test phase.
3. Different feature extraction methods [49], classification algorithms [35, 36, 37] and similarity functions [1, 2] can be used without changing the created graphs.
4. Co-occurrence graphs with a window of two words work very well in the Authorship Analysis and Author Profiling tasks. In the case of Sentiment Analysis, the window of two words has a competitive performance.
5. In the feature-vector approach, the use of centrality measures is a good option to

Table 1. Classification results summary

Feature-vector approach				
Section	**Classification task**	**Approach**	**Classification result**	**Evaluation metric**
3.1.1	Authorship attribution English subset	Author's paper [19]	79.2	A [65]
		Best approach Escalante et al. [24]	86.4	
3.1.2	Sentiment Analysis English subset	Author's paper [16]	42.10	P, R and F [59]
		Best approach Hagen et al. [33]	64.84	
3.1.3	Authorship verification English subset	Author's paper [17]	0.46	C and RA [74]
		Best approach Fréry et al [27]	0.51	
Similarity approach				
Section	**Classification task**	**Approach**	**Classification result**	**Evaluation metric**
3.2.1	Authorship verification English subset	Author's paper [14]	0.47	C and RA [74]
		Best approaches Fréry et al [27]	0.51	
3.2.1	Authorship verification Spanish subset	Author's paper [14]	0.64	C and RA [74]
		Best approaches Khonji et al. [40]	0.69	
3.2.2	Sentiment quantification English subset	Author's paper [15]	0.261	K [52]
		Best approach Stojanovski et al [76]	0.034	
3.2.3	Author profiling English subset	Author's paper [13]	0.82, 0.85 and 0.30	age and gender: A personality: RM [55]
		Best approach Carmona et al [12]	0.83, 0.85 and 0.14	

A: Accuracy P: Presicion R: Recall
F: F1 C: C@1 RA: Roc-Auc
K: KLD RM:RMSE

obtain features that not depend entirely of stylistics aspects of text documents. In the case of the similarity approach, dice similarity helps to obtain the similitude of shared vertices which ultimately lead to obtain similarity between graphs.

6. Most obtained results show that graphs are an alternative that is in the same level as many other of the state of art techniques [74, 52, 57] using graph representations that are easy to build and have a relatively fast performance.
7. Proposed graphs also show to be a fast option considering many traditional techniques that search exhaustively for the best features like an n-gram approach [72].

Research on the use of graph representations continues in favor of improving obtained results, keeping in mind the complexity of the use of graphs. Ongoing and future work includes the following actions:

— Experimenting with other graph-based representations for texts that include semantic information related to texts.

— Exploring different supervised/unsupervised classification algorithms in order to improve results presented in this paper.

— Exploring different text datasets where there is not a predefined task like to find the author or the sentiment in order to test the behavior of proposed graphs when applied to real-world text documents belonging to specific topics [71].

— Experimenting with different visualization methods on graph structures to present and understand obtained information in a more natural and easy-to-understand manner [11].

— Extracting semantic text patterns based on proposed graphs in order to improve previous results.

— Exploring different similarity functions to obtain a more accurate similarity measure between graphs.

— Testing proposed approaches on new datasets (2016) for Authorship Analysis, Sentiment Analysis and Author Profiling tasks [60, 52].

Acknowledgements

This work has been supported by the CONACYT grant with reference #373269/244898 and the CONACYT-PROINNOVA project no. 0198881. The authors would also like to thank Daniel Macías Galindo and David Báez López for their invaluable help reviewing this manuscript.

References

1. **Adamic, L. A. & Adar, E. (2003).** Friends and neighbors on the web. *Social Networks*, Vol. 25, No. 3, pp. 211–230.
2. **Adamic, L. A. & Adar, E. (2004).** Finding and evaluating community structure in networks. *Physical Review E*, Vol. 69, No. 2, pp. 1–16.
3. **Aisopos, F., Papadakis, G., & Varvarigou, T. (2011).** Sentiment analysis of social media content uusing n-gram graphs. *Proceedings of the 3rd International Workshop on Social Media, Arizona, USA*, ACM, pp. 9–14.
4. **Alemán, Y., Loya, N., Vilariño, D., & Pinto, D. (2013).** Two methodologies applied to the author profiling task. *CLEF 2013 Evaluation Labs and Workshop, Online Working Notes, Valencia, Spain*, CEUR Workshop Proceedings, pp. 1–8.
5. **Bagnall, D. (2015).** Author identification using multi-headed recurrent neural networks. *CLEF 2015 Evaluation Labs and Workshop, Online Working Notes, Toulouse, France*, CEUR Workshop Proceedings, pp. 1–11.
6. **Bandara, U., Wijayarathna, G., Lee, M., Hirose, A., Hou, Z. G., & Kil, R. M. (2013).** Deep neural networks for source code author identification. *Proceedings of the International Conference of Neural Information Processing, Heidelberg, Berlin*, Springer, pp. 368–375.
7. **Bird, S., Klein, E., & Loper, E. (2009).** *Natural Language Processing with Python*, chapter Categorizing and Tagging Words. O'Reilly Media, Inc., pp. 179–189.

8. **Bird, S., Klein, E., & Loper, E. (2009).** *Natural Language Processing with Python*, chapter Extracting Information from Text. O'Reilly Media, Inc., pp. 264–277.
9. **Bird, S., Klein, E., & Loper, E. (2009).** *Natural Language Processing with Python*, chapter Accessing Text Corpora and Lexical Resources. O'Reilly Media, Inc., pp. 67–72.
10. **Bouadjenek, M. R., Hacid, H., & Bouzeghoub, M. (2016).** Social networks and information retrieval, how are they converging? a survey, a taxonomy and an analysis of social information retrieval approaches and platforms. *Information Systems*, Vol. 56, pp. 1–18.
11. **Brath, R. & Jonker, D. (2015).** *Graph analysis and visualization: Discovering business opportunity in linked data*, chapter Explore and explain. John Wiley & Sons, pp. 157–181.
12. **Carmona, M. A., López, A. P., Montes-y-Gómez, M., Villaseñor, L., & Escalante, H. J. (2015).** Inaoe's participation at pan'15: Author profiling task. *CLEF 2015 Evaluation Labs and Workshop, Online Working Notes, Toulouse, France*, CEUR Workshop Proceedings, pp. 1–9.
13. **Castillo, E., Cervantes, O., & Vilariño, D. (2017-Under revison).** Author profiling using a graph enrichment approach. *Journal of Intelligent & Fuzzy Systems*, pp. 1–13.
14. **Castillo, E., Cervantes, O., Vilariño, D., & Báez-López, D. (2015).** Author verification using a graph-based representation. *International Journal of Computer Applications*, Vol. 123, No. 14, pp. 1–8.
15. **Castillo, E., Cervantes, O., Vilariño, D., & Báez-López, D. (2016).** UDLAP at semeval-2016 task 4: Sentiment quantification using a graph based representation. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, ACL, pp. 109–114.
16. **Castillo, E., Cervantes, O., Vilariño, D., Báez-López, D., & Sánchez, J. A. (2015).** UDLAP: sentiment analysis using a graph-based representation. *Proceedings of the International Workshop on Semantic Evaluation, Colorado, USA*, ACL, pp. 556–560.
17. **Castillo, E., Cervantes, O., Vilariño, D., Báez-López, D., & Sánchez, J. A. (2017-Under revison).** Authorship verification using a graph knowledge discovery approach. *Computer Speech and Language*, pp. 1–44.
18. **Castillo, E., Cervantes, O., Vilariño, D., Pinto, D., & León, S. (2014).** Unsupervised method for the authorship identification task. *CLEF 2014 Evaluation Labs and Workshop, Online Working Notes, Sheffield, UK*, CEUR Workshop Proceedings, pp. 1035–1041.
19. **Castillo, E., Vilariño, D., Cervantes, O., & Pinto, D. (2015).** Author attribution using a graph based representation. *International Conference on Electronics, Communications and Computers, Puebla, Mexico*, IEEE, pp. 135–142.
20. **Castillo, E., Vilariño, D., Pinto, D., Olmos, I., Gonzalez, J. A., & Carrillo, M. (2012).** Graph-based and lexical-syntactic approaches for the authorship attribution task. *CLEF 2012 Evaluation Labs and Workshop, Online Working Notes, Rome, Italy*, CEUR Workshop Proceedings, pp. 1–7.
21. **Cook, D. J. & Holder, L. B. (2006).** *Mining graph data*, chapter Applications. John Wiley & Sons, pp. 345–468.
22. **Deng, S., Sinha, A. P., & Zhao, H. (2016).** Adapting sentiment lexicons to domain-specific social media texts. *Decision Support Systems*.
23. **Dichiu, D. & Rancea, I. (2016).** Using machine learning algorithms for author profiling in social media. *CLEF 2016 Evaluation Labs and Workshop, Online Working Notes, Évora, Portugal*, CEUR Workshop Proceedings, pp. 858–863.
24. **Escalante, H. J., Solorio, T., & Montes-y-Gómez, M. (2011).** Local histograms of character n-grams for authorship attribution. *Proceedings of the Human Language Technologies, Oregon, USA*, ACL, pp. 288–298.
25. **Flöck, F. & Acosta, M. (2014).** Wikiwho: precise and efficient attribution of authorship of revisioned content. *Proceedings of the 23rd international conference on World wide web*, ACM, pp. 1–11.
26. **Freeman, L. C. (1978).** Centrality in social networks conceptual clarification. *Social Networks*, Vol. 1, No. 3, pp. 215–239.
27. **Frèry, J., Largeron, C., & Juganaru, M. (2014).** Ujm at clef in author identification notebook for PAN at clef. *CLEF 2014 Evaluation Labs and Workshop, Online Working Notes, Sheffield, UK*, CEUR Workshop Proceedings, pp. 1042–1048.
28. **Gelbukh, A. & Sidorov, G. (2010).** *Procesamiento automático del español con enfoque en recursos léxicos grandes*, chapter Tareas y aplicaciones de PLN. IPN, pp. 37–85.

29. **Giannakopoulos, G., Karkaletsis, V., Vouros, G., & Stamatopoulos, P. (2008).** Summarization system evaluation revisited: n-gram graphs. *ACM Transactions on Speech and Language Processing*, Vol. 5, No. 3, pp. 1–38.
30. **Gollub, T., Potthast, M., Beyer, A., Busse, M., Rangel, F. M., Rosso, P., Stamatatos, E., & Stein, B. (2013).** Recent trends in digital text forensics and its evaluation - plagiarism detection, author identification, and author profiling. *CLEF 2013 Evaluation Labs and Workshop, Online Working Notes, Valencia, Spain*, CEUR Workshop Proceedings, pp. 1–20.
31. **Gómez-Adorno, H., Markov, I., Sidorov, G., Posadas, J. P., Sanchez, M. A., & Hernandez, L. C. (2016).** Improving feature representation based on a neural network for author profiling in social media texts. *Computational Intelligence and Neuroscience*, Vol. 2016, pp. 1–14.
32. **Gómez-Adorno, H., Sidorov, G., Pinto, D., Vilariño, D., & Gelbukh, A. F. (2016).** Automatic authorship detection using textual patterns extracted from integrated syntactic graphs. *Sensors*, Vol. 16, No. 9, pp. 1–19.
33. **Hagen, M., Potthast, M., Büchner, M., & Stein, B. (2015).** Webis: An ensemble for twitter sentiment detection. *Proceedings of the International Workshop on Semantic Evaluation, Colorado, USA*, ACL, pp. 582–589.
34. **Harrington, P. (2012).** *Machine Learning in Action*, chapter Machine learning basics. Manning Publications Co., pp. 3–17.
35. **Harrington, P. (2012).** *Machine Learning in Action*, chapter Classifying with probability theory: Naïve Bayes. Manning Publications Co., pp. 61–82.
36. **Harrington, P. (2012).** *Machine Learning in Action*, chapter Logistic regression. Manning Publications Co., pp. 83–100.
37. **Harrington, P. (2012).** *Machine Learning in Action*, chapter Support vector machines. Manning Publications Co., pp. 101–127.
38. **Ingersoll, G. S., Morton, T. S., & Farris, A. L. (2012).** *Taming text*, chapter Searching. Manning Publications Co., pp. 76–76.
39. **Juola, P. (2008).** Authorship attribution. *Foundations and trends in Information Retrieval*, Vol. 1, No. 3, pp. 233–334.
40. **Khonji, M. & Iraqi, Y. (2014).** A slightly-modified gi-based author-verifier with lots of features (ASGALF). *CLEF 2014 Evaluation Labs and Workshop, Online Working Notes, Sheffield, UK*, CEUR Workshop Proceedings, pp. 977–983.
41. **Koppel, M. & Schler, J. (2004).** Authorship verification as a one-class classification problem. *Proceedings of the Twenty-first International Conference on Machine Learning*, ACM, pp. 1–7.
42. **Lahiri, S. & Mihalcea, R. (2013).** Authorship attribution using word network features. Computing Research Repository, pp. 1–11.
43. **López, A. P., Montes-y-Gómez, M., Escalante, H. J., Pineda, L. V., & Stamatatos, E. (2016).** Discriminative subprofile-specific representations for author profiling in social media. *Knowledge-Based Systems*, Vol. 89, pp. 134–147.
44. **Macke, S. & Hirshman, J. (2015).** Deep sentence-level authorship attribution. Stanford University, pp. 1–7.
45. **Malak, M. S. & East, R. (2016).** *Spark GraphX in action*, chapter Two important technologies: Spark and graphs. Manning Publications., pp. 3–23.
46. **Manning, C. D., Raghavan, P., & Schütze, H. (2008).** *Introduction to information retrieval*, chapter The term vocabulary and postings lists. Cambridge University Press, pp. 19–45.
47. **Manning, C. D., Raghavan, P., & Schütze, H. (2008).** *Introduction to Information Retrieval*, chapter Scoring, term weighting and the vector space model. Cambridge University Press, pp. 117–120.
48. **Marinho, V. Q., Hirst, G., & Amancio, D. R. (2016).** Authorship attribution via network motifs identification. *5th Brazilian Conference on Intelligent Systems*, Computing Research Repository, pp. 1–6.
49. **Mihalcea, R. & Radev, D. (2011).** *Graph-based Natural Language Processing and Information Retrieval*, chapter Language networks. Cambridge University Press, pp. 78–80.
50. **Mihalcea, R. & Radev, D. (2011).** *Graph-based Natural Language Processing and Information Retrieval*, chapter Sentiment and subjectivity. Cambridge University Press, pp. 135–139.
51. **Müller, A. C. & Guido, S. (2016).** *Introduction to Machine Learning With Python: A Guide for Data Scientists*, chapter Supervised learning. O'Reilly Media, Inc., pp. 25–127.
52. **Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., & Stoyanov, V. (2016).** Semeval-2016 task 4: Sentiment analysis in twitter. *Proceedings*

of the 10th International Workshop on Semantic Evaluation (SemEval-2016), ACM, pp. 1–18.

53. **Nowell, D. L. & Kleinberg, J. (2007).** The link-prediction problem for social networks. *Journal of the Association for Information Science and Technology*, Vol. 58, No. 7, pp. 1019–1031.
54. **Pang, B. & Le, L. (2008).** Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, Vol. 60, No. 1-2, pp. 1–135.
55. **Rangel, F. M., Celli, F., Rosso, P., Potthast, M., Stein, B., & Daelemans, W. (2015).** Overview of the 3rd author profiling task at pan 2015. *CLEF 2015 Evaluation Labs and Workshop, Online Working Notes, Toulouse, France*, CEUR Workshop Proceedings, pp. 1–40.
56. **Rangel, F. M. & Rosso, P. (2016).** On the impact of emotions on author profiling. *Information Processing & Management*, Vol. 52, No. 1, pp. 73 – 92.
57. **Rangel, F. M., Rosso, P., Verhoeven, B., Daelemans, W., Potthast, M., & Stein, B. (2016).** Overview of the 4th author profiling task at pan 2016: Cross-genre evaluations. *CLEF 2016 Evaluation Labs and Workshop, Online Working Notes, Évora, Portugal*, CEUR Workshop Proceedings, pp. 1–35.
58. **Riemer, M., Krasikov, S., & Srinivasan, H. (2015).** A deep learning and knowledge transfer based architecture for social media user characteristic determination. *SocialNLP 2015 NAACL*, pp. 1–39.
59. **Rosenthal, S., Nakov, P., Kiritchenko, S., Mohammad, S., Ritter, A., & Stoyanov, V. (2015).** Semeval-2015 task 10: Sentiment analysis in twitter. *Proceedings of the International Workshop on Semantic Evaluation, Colorado, USA*, ACL, pp. 451–463.
60. **Rosso, P., Pardo, F. R., Potthast, M., Stamatatos, E., Tschuggnall, M., & Stein, B. (2016).** Overview of pan'16 - new challenges for authorship analysis: Cross-genre profiling, clustering, diarization, and obfuscation. *CLEF 2016 Evaluation Labs and Workshop, Online Working Notes, Évora, Portugale*, CEUR Workshop Proceedings, pp. 332–350.
61. **Rosso, P., Rangel, F. M., Potthast, M., Stamatatos, E., Tschuggnall, M., & Stein, B. (2016).** Overview of pan 2016 - new challenges for authorship analysis: Cross-genre profiling, clustering, diarization, and obfuscation. *Proceedings of CLEF PAN 2016*, Lecture Notes in Computer Science, pp. 332–350.
62. **Samatova, N. F., Hendrix, W., Jenkins, J., Padmanabhan, K., & Chakraborty, A. (2013).** *Practical graph mining with R*, chapter Introduction. Chapman & Hall/CRC, pp. 1–7.
63. **Samatova, N. F., Hendrix, W., Jenkins, J., Padmanabhan, K., & Chakraborty, A. (2013).** *Practical graph mining with R*, chapter Frequent subgraph mining. Chapman & Hall/CRC, pp. 180–186.
64. **Samatova, N. F., Hendrix, W., Jenkins, J., Padmanabhan, K., & Chakraborty, A. (2013).** *Practical graph mining with R*, chapter Link analysis. Chapman & Hall/CRC, pp. 110–121.
65. **Samatova, N. F., Hendrix, W., Jenkins, J., Padmanabhan, K., & Chakraborty, A. (2013).** *Practical graph mining with R*, chapter Performance metrics for graph mining tasks. Chapman & Hall/CRC, pp. 373–381.
66. **Savatic, A., Jamak, A., & Can, M. (2012).** Detecting the authors of texts by neural network committee machines. *Southeast Europe Journal of Soft Computing*, Vol. 1, No. 1, pp. 1–12.
67. **Sebastiani, F. (2002).** Machine learning in automated text categorization. *ACM Computing Surveys*, Vol. 34, No. 1, pp. 1–47.
68. **Segarra, S., Eisen, M., & Ribeiro, A. (2013).** Authorship attribution through function word adjacency networks. *International Conference on Acoustics, Speech and Signal Processing*, IEEE, pp. 5563–5567.
69. **Severyn, A. & Moschitti, A. (2015).** Twitter sentiment analysis with deep convolutional neural networks. *Proceedings of the 38th Conference on Research and Development in Information Retrieval,New York, USA*, ACM, pp. 959–962.
70. **Sharma, A. & Shubhamoy, D. (2012).** An artificial neural network based approach for sentiment analysis of opinionated text. *Proceedings of the ACM Research in Applied Computation Symposium, New York, USA*, ACM, pp. 37–42.
71. **Solorio, T., Hasan, R., & Mizan, M. (2013).** Sockpuppet detection in wikipedia: A corpus of real-world deceptive writing for linking identities. *Computing research repository*, pp. 1–4.
72. **Stamatatos, E. (2009).** A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, Vol. 60, No. 3, pp. 538–556.
73. **Stamatatos, E., Daelemans, W., Verhoeven, B., Juola, P., López, A., Potthast, M., & Stein,**

B. (2015). Overview of the author identification task at pan 2015. *CLEF 2015 Evaluation Labs and Workshop, Online Working Notes, Toulouse, France*, CEUR Workshop Proceedings, pp. 1–17.

74. **Stamatatos, E., Daelemans, W., Verhoeven, B., Potthast, M., Stein, B., Juola, P., Perez, M. S., & Cedeño, A. B. (2014).** Overview of the author identification task at PAN 2014. *CLEF 2014 Evaluation Labs and Workshop, Online Working Notes, Sheffield, UK*, CEUR Workshop Proceedings, pp. 877–897.
75. **Stańczyk, U. & Cyran, K. A. (2007).** Machine learning approach to authorship attribution of literary texts. *International journal of applied mathematics and informatics*, Vol. 1, No. 4, pp. 151–158.
76. **Stojanovski, D., Strezoski, G., Madjarov, G., & Dimitrovski, I. (2016).** Finki at semeval-2016 task 4: Deep learning architecture for twitter sentiment analysis. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, ACL, pp. 149–154.
77. **Tang, D., Qin, B., & Liu, T. (2015).** Deep learning for sentiment analysis: Successful approaches and future challenges. *Wiley. Data Mining and Knowledge Discovery*, Vol. 5, No. 6, pp. 292–303.
78. **Tsvetovat, M. & Kouznetsov, A. (2011).** *Social network analysis for startups*, chapter Introduction. O'Reilly Media, Inc., pp. 1–14.
79. **Verhoeven, B., Company, J. S., & Daelemans, W. (2014).** Evaluating content-independent features for personality recognition. *Proceedings of the Workshop on Computational Personality Recognition, Florida, USA*, ACM, pp. 7–10.
80. **Vilariño, D., Pinto, D., Gómez-Adorno, H., León, S., & Castillo, E. (2013).** Lexical-syntactic and graph-based features for authorship verification. *CLEF 2013 Evaluation Labs and Workshop, Online Working Notes, Valencia, Spain*, CEUR Workshop Proceedings, pp. 1–6.
81. **White, T. (2015).** *Hadoop: The definitive guide*, chapter Meet Hadoop. O'Reilly Media, Inc., pp. 3–15.
82. **Wiebe, J. & Mihalcea, R. (2006).** Word sense and subjectivity. *Proceedings of the 21st International Conference on Computational Linguistics*, ACL, pp. 1065–1072.
83. **Wu, Y., Zhang, Q., Huang, X., & Wu, L. (2011).** Structural opinion mining for graph-based sentiment representation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ACL, pp. 1332–1341.

Article received on 08/02/2017; accepted on 05/09/2017.
Corresponding author is Esteban Castillo.