







An Innovative Graph-Based Approach to Advance Feature Selection from Multiple Textual Documents

Nikolaos Giarelis , Nikos Kanakaris , and Nikos Karacapilidis  

Industrial Management and Information Systems Lab, MEAD, University of Patras,
26504 Rio, Patras, Greece

giarelis@ceid.upatras.gr, nkanakaris@upnet.gr, karacap@upatras.gr

Abstract. This paper introduces a novel graph-based approach to select features from multiple textual documents. The proposed solution enables the investigation of the importance of a term into a whole corpus of documents by utilizing contemporary graph theory methods, such as community detection algorithms and node centrality measures. Compared to well-tried existing solutions, evaluation results show that the proposed approach increases the accuracy of most text classifiers employed and decreases the number of features required to achieve ‘state-of-the-art’ accuracy. Well-known datasets used for the experimentations reported in this paper include *20Newsgroups*, *LingSpam*, *Amazon Reviews* and *Reuters*.

Keywords: Feature selection · Graph-based text representation · Document clustering · Text mining · Natural Language Processing

1 Introduction

Graph-based text representations are widely used in various Natural Language Processing, Text Mining and Information Retrieval tasks (Vazirgiannis et al. 2018). These representations exploit concepts and techniques inherited from graph theory (e.g. node centrality and subgraph frequency) to address limitations of the classical *bag-of-words* representation (Aggarwal 2018); in this way, they are able to capture structural and semantic information of a text, mitigate the effects of the ‘curse-of-dimensionality’ phenomenon, identify the most important terms of a text, and seamlessly incorporate information coming from external knowledge sources. However, existing graph-based representations concern a single document each time. In cases where one needs to analyze a corpus of documents, these approaches demonstrate a series of weaknesses, the main of them being that they are incapable to assess the importance of a word for the whole set of documents.

Recently, graph-based text representations have been used to facilitate and augment the feature selection process, i.e. the process of selecting a subset of relevant features when constructing a model. These approaches combine statistical tests and graph algorithms to uncover hidden correlations between terms and document classes. However,

while they take into account the co-occurrences between terms to identify the most representative features of a single document (something that is not the case in traditional statistical methods), they are not able to assess the importance of a term in a corpus of documents. To remedy the above weakness, this paper builds on a graph-based text representation model to introduce a novel approach to feature selection from multiple textual documents, namely *GraFS*. Contrary to existing approaches, the one introduced in this paper (i) enables the investigation of the importance of a term into a whole corpus of documents, (ii) incorporates the relationships between terms (co-occurrences) into the feature selection process, (iii) achieves state-of-the-art accuracy in ML tasks such as text classification using fewer features, and (iv) mitigates the effects of the ‘curse-of-dimensionality’ phenomenon. *GraFS* has been evaluated by using five datasets and five classifiers. Compared to four well-tried existing feature selection approaches, our initial experimental results show that *GraFS* increases the accuracy of most text classifiers and decreases the number of features required to achieve ‘state-of-the-art’ accuracy.

The remainder of the paper is organized as follows. Section 2 discusses related work issues. The proposed feature selection approach is presented in Sect. 3. Section 4 reports on the experiments carried out to assess the proposed approach against previous ones. Limitations of our approach, future work directions and concluding remarks are outlined in Sect. 5.

2 Background Work

The proposed feature selection approach builds on a graph-based representation of multiple textual documents and exploits advantages of contemporary graph databases. This section highlights related background work issues.

2.1 Graph-Based Text Representations

Graph-of-words is a well-known graph-based text representation method. Being similar to the bag-of-words approach that has been widely used in the NLP field, it enables a sophisticated keyword extraction and feature engineering process. In a graph of words, each node represents a unique term (i.e. word) of a document and each edge represents the co-occurrence between two terms within a sliding window of text. The utilization of a small sliding window size, due to the fact that larger ones produce heavily interconnected graphs where the valuable information is cluttered with noise, has been proposed in (Nikolentzos et al. 2017). In this direction, work described in (Rousseau et al. 2015) suggests that a window size of four is generally considered as the appropriate value, since it does not sacrifice either the performance or the accuracy of their approach.

2.2 Graph-Based Feature Selection

Several interesting graph-based feature selection approaches have been already proposed in the literature. For instance, (Rousseau et al. 2015) proposes various combinations and configurations of popular frequent subgraph mining techniques - such as *gSpan* (Yan and Han 2002), *Gaston* (Nijssen and Kok 2004) and *gBoost* (Saigo et al. 2009) - to

perform unsupervised feature selection exploiting the k-core subgraph. In particular, aiming to increase performance, Rousseau and his colleagues rely on the concept of k-core subgraph to reduce the graph representation to its densest part. The experimental results show a significant increment of the accuracy compared to common classification approaches. The work reported in (Henni et al. 2018) applies centrality algorithms (such as PageRank) to calculate the centrality score of a graph’s features and accordingly identify the most important ones. The approach presented in (Fakhraei et al. 2015) builds on combinations of several types of graph algorithms to discover highly connected features of a graph. Such algorithms include the Louvain Algorithm for community detection and the PageRank algorithm to discover influential nodes and other user-defined graph measures. This last approach combines PageRank and Coloring algorithms with the custom graph measures of in-degree and out-degree.

Other already proposed approaches rely on the recursive filtering of the existing feature space; for instance, one of them re-applies PageRank to find the most influential features (Ienco et al. 2008). These approaches use graph-connected features to include contextual information, as modelled implicitly by a graph structure, using relationships that describe connections among real data. They aim to reduce ambiguity in feature selection and improve accuracy in traditional Machine Learning methods.

2.3 Graph Databases

Compared to relational databases, graph databases provide a more convenient and efficient way to natively represent and store highly interlinked data. Moreover, they allow the retrieval of multiple relationships and entities with a single operation, thus avoiding the utilization of rigid join operations which are heavily used in relational databases (Miller 2013). An in-depth review of graph databases can be found in (Rawat and Kashyap 2017).

3 GraFS: Graph-Based Feature Selection

3.1 Graph-of-Docs Text Representation

To select the most representative features of a corpus of documents, we build on the *graph-of-docs* text representation, first proposed in (Giarelis et al. 2020). Aiming to represent multiple documents in a single graph, the graph-of-docs representation expands the well-known ‘graph-of-words’ model that produces a single graph for each individual document (Rousseau et al. 2013). Graph-of-docs allows diverse types of nodes and edges to co-exist in a graph, including nodes with types such as ‘document’ and ‘word’, and edges with types such as ‘is_similar’, ‘connects’, ‘includes’, and ‘feature’ (see Fig. 1).

Briefly, according to the graph-of-docs representation:

- each unique word node is connected to all the document nodes where it belongs to using edges of the ‘includes’ type;
- each unique word node selected as a feature is connected to document nodes using edges of the ‘feature’ type;

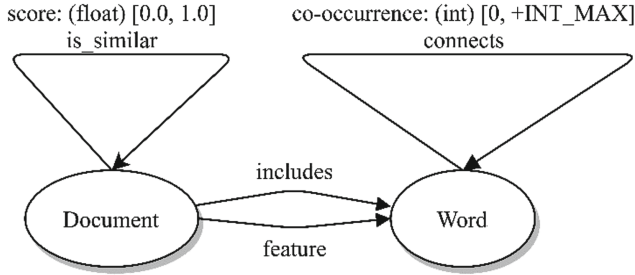


Fig. 1. The schema of the graph-of-docs representation model.

- edges of 'connects' type are only applicable between two word nodes and denote their co-occurrence within a specific sliding text window;
- edges of the 'is_similar' type link a pair of document nodes and indicate their contextual similarity.

Graph-of-docs enables us to investigate the importance of a term not only within a single document but also within a whole corpus of documents, which in turn augments the quality of the overall feature selection process.

3.2 Feature Selection

Our approach consists of four steps: (i) creation of a document similarity subgraph; (ii) detection of document communities; (iii) feature selection for each community, and (iv) feature selection for the whole corpus of documents.

Creation of a Document Similarity Subgraph. We argue that subgraphs from the graph-of-docs graph describing similar documents share common word nodes as well as similar structural characteristics. This enables us to calculate the similarity between two documents by using typical data mining similarity measures, which in turn facilitates the production of a similarity subgraph. The similarity subgraph consists of document nodes and edges of the 'is_similar' type.

Detection of Document Communities. By exploiting the document similarity subgraph, we detect communities of contextually similar documents using the 'score' property of the 'is_similar' type edges as a distance value. A plethora of community detection algorithms can be found in the literature, including *Louvain*, *Label Propagation* and *Weakly Connected Components*.

Feature Selection for Each Community. Since documents that are in the same community are contextually similar, we assume that it is also more likely that they share common features (see Fig. 2). Aiming to find the top-N most representative features of each community, GraFS ranks the terms of each community by their document frequency and their PageRank score.

Feature Selection for the Whole Corpus of Documents. The final step defines the feature space by merging the top-N features of each community. This reduces the number of the candidate features, something that (i) accelerates the feature selection process, (ii) mitigates the effects of the ‘curse-of-dimensionality’ phenomenon, and (iii) enables the training of more reliable ML models.

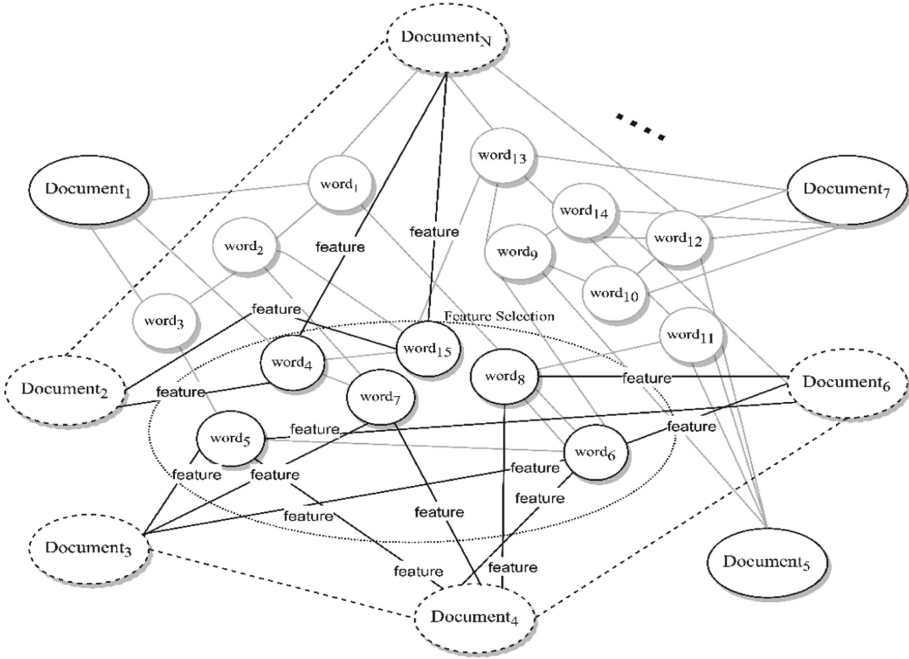


Fig. 2. Feature selection using the graph-of-docs text representation model. The selected features, shown within the circle, are linked to documents with edges of ‘feature’ type. Relationships between documents are denoted with dotted lines.

4 Experiments

For the implementation and evaluation of our approach, we used the Python programming language and the scikit-learn ML library (<https://scikit-learn.org>). The Neo4j graph database (<https://neo4j.com>) has been utilized for the needs of the graph-of-docs representation. The full code, the documentation and the evaluation results of our experiments are freely available at <https://github.com/NCODER/GraphOfDocs>.

4.1 Baseline Methods

This subsection presents the benchmarks used to evaluate the performance of GraFS. For the implementation of these methods, we used the scikit-learn ML library (implementation details can be found at https://scikit-learn.org/stable/modules/feature_selection.html).

Low Variance Feature Selection (LVAR). The first benchmark removes the features that do not meet a predefined variance threshold (Aggarwal 2018). This method is referred to as *LVAR* in the remainder of this paper (scikit-learn library, class: `sklearn.feature_selection.VarianceThreshold`).

Univariate Feature Selection (KBEST). The second benchmark relies on univariate statistical tests to select the k-best features (Aggarwal 2018). In particular, it attempts to find correlations between an individual feature and a document class. In this paper, we adopt the χ^2 test as our main statistical test. This method is referred to as *KBEST* in the remainder of this paper (scikit-learn library, classes: `sklearn.feature_selection.SelectKBest` and `sklearn.feature_selection.chi2`).

Feature Selection Using a Meta-Transformer Model (META). The third benchmark uses a meta-transformer model to retain only the features with significant importance. It is assumed that a statistical model (e.g. logistic regression) provides importance metrics for each feature to be considered as a candidate meta-transformer model. Available meta-transformer models include logistic regression, linear SVM and neural networks, as well as more sophisticated methods such as word embeddings (e.g. *word2vec* (Mikolov et al. 2013)). In this paper, we use the linear SVM model, since it performs well regardless of the number of samples or the number of unique features of a dataset. In the remainder of the paper, this method is referred to as *META* (scikit-learn library, classes: `sklearn.feature_selection.SelectFromModel` and `sklearn.svm.LinearSVC`).

4.2 Datasets

This subsection describes the datasets used in our experiments to evaluate the performance of GraFS. These datasets are available at <https://github.com/imis-lab/aiai-2020-datasets>.

20Newsgroups. We tested the proposed model by utilizing an already preprocessed version of the well-known *20Newsgroups* dataset, which is a collection of approximately 20,000 newsgroup documents, partitioned evenly across 20 different newsgroups. As far as the multi-class text classification task is concerned, this dataset fits well to the purposes of our experimentations since it provides a large volume of different documents on the same subjects.

Reuters. We tested the proposed model on a preprocessed version of the Reuters dataset, which includes 21,578 news stories; since almost half of them lack the class field, we used only the ones that came along with their class (i.e. 10,377). For each news story, certain attributes were retained; for instance, the ‘title’ attribute that contains the title of the story and the ‘body’ attribute that contains the main text of the news story. In this paper, we used this dataset to execute experiments related to the multi-class text classification task.

Amazon Reviews. We also tested the proposed model on a preprocessed version of the Amazon Reviews dataset, which contains labeled (positive or negative) reviews of products belonging to different categories (e.g. automotive, electronics, grocery etc.). We picked four product categories (i.e. books, DVD, electronics, kitchen), each having 1000 positive and 1000 negative reviews. We utilized this dataset to conduct experiments related to the opinion mining task.

LingSpam. The LingSpam dataset (Androustopoulos et al. 2000) contains 2,893 email messages, which are classified either as ‘spam’ or ‘not spam’. We utilized this dataset to conduct experiments related to the spam detection task.

JiraIssues. The JiraIssues dataset, concerns the development of 168 software projects including ‘Hadoop’, ‘Spark’ and ‘Airflow’. It contains information related to 228,969 Jira issues. Each Jira issue in this dataset has the attributes ‘description’, and ‘assignee’. The set of the document classes of the dataset corresponds to the names of the available employees (‘assignee’ attribute). This dataset was retrieved from the publicly accessible Jira instance of Apache Software Foundation (<https://issues.apache.org/jira>). We utilized it to execute experiments related to the multi-class text classification task.

Table 1. The hyper-parameters of each feature selection method per dataset.

Method	Dataset	Hyper-parameter	Values
LVAR	20Newsgroups, Reuters, Amazon, LingSpam, JiraIssues	Variance threshold	[0.0005, 0.001, 0.0015, 0.002, 0.003, 0.004, 0.005, 0.01]
GraFS	20Newsgroups, Reuters, Amazon, LingSpam, JiraIssues	top-N	[5, 10, 15, 20, 25, 50, 100, 250, 500]
KBEST	20Newsgroups	k	[1000, 2000, 3000, 5000, 10000, 15000, 20000, 25000, 30000]
KBEST	Reuters, Amazon, JiraIssues	k	[1000, 2000, 3000, 5000, 6000, 7000, 8000, 10000, 14000]
KBEST	LingSpam	k	[250, 500, 1000, 1500, 2000, 2500, 3000, 4000, 5000]

4.3 Experimental Setup

To identify the most important words in the entire corpus of documents, we selected to use the *PageRank* algorithm, since it performs well regardless of the topics of the documents. To identify similar documents needed for the generation of the document similarity subgraph, we used the *Jaccard* similarity measure since it deals only with the absence or the presence of a word, ignoring its document frequency. To form communities of similar documents, we used the *Louvain* community detection algorithm. Finally,

we executed several experiments with different hyperparameter values for the LVAR, KBEST and GraFS feature selection methods. Table 1 summarizes the values given to these hyperparameters per dataset.

4.4 Evaluation

To evaluate the effectiveness of our approach, we assess the contribution of GraFS in the accuracy of widely used text classifiers against the bag-of-words (BOW) text representation and the three domain-agnostic feature selection techniques described in Sect. 4.1 (see Table 2). The text classifiers considered are: *naive Bayes* (NB), *k-nearest neighbors* (5NN), *logistic regression* (LR), *neural networks* (NN100x50) and *linear support vector machines* (LSVM). It is noted that in the case of BOW, none of the feature selection techniques has been applied to the specific experiment. Results obtained show that GraFS (i) increases the accuracy in most cases, and (ii) decreases the number of features required to achieve ‘state-of-the-art’ accuracy (Fig. 3 – right part). Figure 3 illustrates the accuracy of the LSVM classifier per number of selected features for the GraFS, KBEST and LVAR feature selection techniques (additional comparisons can be retrieved from <https://github.com/imis-lab/aiai-2020-datasets>).

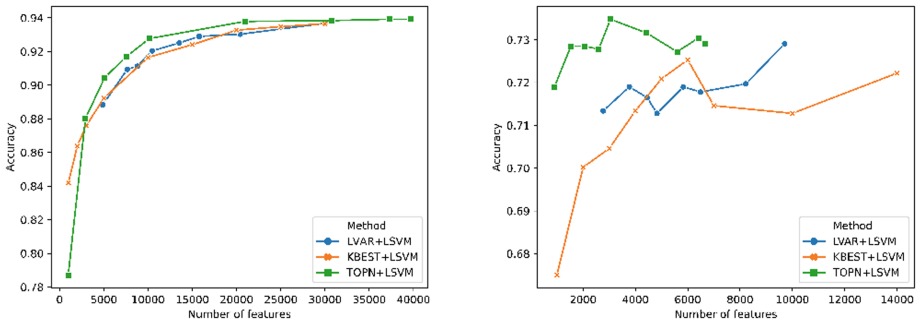


Fig. 3. Accuracy of the LSVM classifier per number of selected features for the GraFS (TOPN), KBEST and LVAR feature selection techniques on *20Newsgroups* (left) and *JiraIssues* (right) datasets.

Our approach differs from the existing ones in that it considers the whole corpus of documents (instead of each document separately) and the associated relationships between the words. Thus, the feature set selected using GraFS contains the most influential features of a document corpus. Hence, GraFS reduces the number of the selected features, which in turn mitigates the effects of the ‘curse-of-dimensionality’ phenomenon, i.e. the production of over-fitted ML models and sparse feature vectors. Contrary to our approach, common feature selection methods that are based on statistics ignore the interconnections between the terms (both within a single document and across the documents of a corpus), which has as effect that more features are required from the text classifiers to perform equally well (see the left graph in Fig. 3).

Table 2. Accuracy score (acc) and number of features (|f|) per text classifier for each feature selection technique on the five selected textual datasets. * and bold font highlights the best method for a specific dataset as far as the accuracy score and the number of features are concerned.

Method	20Newsgroups		Reuters		Amazon		LingSpam		JiraIssues	
	acc	f	acc	f	acc	f	acc	f	acc	f
GraFS+5NN	0.7192	20942	0.8272	809	0.6786	1065	0.9926	120	0.682	905
GraFS+NB	0.9421	20942	0.8399	7171	0.7273	1940	0.9963	2274	0.6958	6404
GraFS+LR	0.9402	37281	0.8782*	7171	0.776	4897	1.0*	120	0.7517	2598
GraFS+NN100x50	0.9575*	30793	0.8733	8187	0.7403	1940	1.0	758	0.7542*	6404
GraFS+LSVM	0.9392	39694	0.8737	7171	0.763	4897	0.9963	120	0.7348	3045
BOW+5NN	0.643	62384	0.7582	15514	0.6169	9771	0.8333	16695	0.6486	14539
BOW+NB	0.9361	62384	0.8191	15514	0.7208	9771	0.9963	16695	0.6989	14539
BOW+LR	0.9387	62384	0.8746	15514	0.763	9771	1.0	16695	0.7461	14539
BOW+NN100X50	0.9546	62384	0.8656	15514	0.7273	9771	0.9963	16695	0.741	14539
BOW+LSVM	0.9408	62384	0.8742	15514	0.763	9771	1.0	16695	0.7304	14539
LVAR+5NN	0.6942	4880	0.8209	1482	0.7013	1719	0.8926	5464	0.6493	2771
LVAR+NB	0.94	29992	0.8403	3356	0.737	2906	0.9963	8234	0.7373	5833
LVAR+LR	0.9384	29992	0.8755	7624	0.7792	3637	1.0	8234	0.7442	9706
LVAR+NN100X50	0.9541	29992	0.8724	4870	0.75	1719	1.0	11058	0.7398	6489
LVAR+LSVM	0.9368	29992	0.8746	7624	0.7825	1719	1.0	16695	0.7291	9706
KBEST+5NN	0.721	5000	0.7957	6000	0.724	350	0.9778	5000	0.6644	4000
KBEST+NB	0.9374	25000	0.8354	7000	0.75	500	0.9963	3000	0.6952	14000
KBEST+LR	0.9389	30000	0.8764	10000	0.7727	3000	1.0	1000	0.7461	6000
KBEST+NN100X50	0.9564	30000	0.8705	14000	0.7565	1000	1.0	1000	0.7423	10000
KBEST+LSVM	0.9366	30000	0.8737	14000	0.763	6000	1.0	1000	0.7253	6000
META+5NN	0.6542	14907	0.8002	2494	0.6623	2731	0.8704	2509	0.6329	2942
META+NB	0.9387	14907	0.8376	2494	0.737	2731	0.9963	2509	0.6989	2942
META+LR	0.9376	14907	0.876	2494	0.789*	2731	1.0	2509	0.7461	2942
META+NN100X50	0.952	14907	0.8701	2494	0.75	2731	1.0	2509	0.7467	2942
META+LSVM	0.9408	14907	0.8746	2494	0.7727	2731	1.0	2509	0.731	2942

5 Conclusions

This paper introduces a new approach for graph-based feature selection, namely *GraFS*. To test the proposed approach, we benchmarked *GraFS* against classical feature selection techniques. The evaluation outcome was very promising; state-of-the-art accuracy has been achieved in the classification of five well-known datasets using fewer features. In any case, our approach demonstrates two limitations: (i) it is unable to select features for outlier documents, i.e. documents that are not similar to any other document, and (ii) it requires significant time to generate the corresponding graph of documents in a disk-based graph database.

Aiming to address the above limitations as well as to integrate our approach into existing works on knowledge management systems, future work directions include: (i) the utilization and assessment of an in-memory graph database in combination with Neo4j; (ii) the exploitation of link prediction algorithms to deal with outlier documents; (iii) the application of graph and word embedding techniques, and (iv) the integration of our approach into collaborative argumentation environments where the underlying knowledge is structured through semantically-rich discourse graphs (e.g. integration with the approaches described in (Kanterakis et al. 2019) and (Karacapilidis et al. 2009)).

Acknowledgments. The work presented in this paper is supported by the OpenBio-C project (www.openbio.eu), which is co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE – INNOVATE (Project id: T1EDK- 05275).

References

- Aggarwal, C.C.: Machine Learning for Text. Springer, Cham (2018). <https://doi.org/10.1007/978-3-319-73531-3>
- Androutsopoulos, I., Koutsias, J., Chandrinou, K., Paliouras, G., Spyropoulos, C.: An evaluation of Naïve Bayesian anti-spam filtering. In: Proceedings of the Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning, pp. 9–17 (2000)
- Fakhraei, S., Foulds, J., Shashanka, M., Getoor, L.: Collective spammer detection in evolving multi-relational social networks. In: Proceedings of the 21 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1769–1778 (2015)
- Giarelis, N., Kanakaris, N., Karacapilidis, N.: On a novel representation of multiple textual documents in a single graph. In: Czarnowski, I., Howlett, R.J., Jain, L.C. (eds.) Intelligent Decision Technologies 2020 – Proceedings of the 12th KES International Conference on Intelligent Decision Technologies (KES-IDT-20), Split, Croatia, 17–19 June 2020. Springer (2020)
- Henni, K., Mezghani, N., Gouin-Vallerand, C.: Unsupervised graph-based feature selection via subspace and PageRank centrality. *Expert Syst. Appl.* **114**, 46–53 (2018)
- Inco, D., Meo, R., Botta, M.: Using PageRank in feature selection. In: SEBD, pp. 93–100 (2008)
- Kanterakis, A., et al.: Towards reproducible bioinformatics: the OpenBio-C scientific workflow environment. In: Proceedings of the 19th IEEE International Conference on Bioinformatics and Bioengineering (BIBE), Athens, Greece, pp. 221–226 (2019)
- Karacapilidis, N., et al.: Tackling cognitively-complex collaboration with CoPe_it! *Int. J. Web-Based Learn. Teach. Technol.* **4**(3), 22–38 (2009)

- Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3111–3119 (2013)
- Miller, J.J.: Graph database applications and concepts with Neo4j. In: *Proceedings of the Southern Association for Information Systems Conference*, vol. 2324, no. 36, Atlanta, USA (2013)
- Nijssen, S., Kok, J.N.: A quickstart in frequent structure mining can make a difference. In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 647–652. ACM Press (2004)
- Nikolentzos, G., Meladianos, P., Rousseau, F., Stavrakas, Y., Vazirgiannis, M.: Shortest-path graph kernels for document similarity. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1890–1900 (2017)
- Rawat, D.S., Kashyap, N.K.: Graph database: a complete GDBMS survey. *Int. J.* **3**, 217–226 (2017)
- Rousseau, F., Kiagias, E., Vazirgiannis, M.: Text categorization as a graph classification problem. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, vol. 1, pp. 1702–1712 (2015)
- Rousseau, F., Vazirgiannis, M.: Graph-of-word and TW-IDF: new approach to ad hoc IR. In: *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, pp. 59–68. ACM Press (2013)
- Saigo, H., Nowozin, S., Kadowaki, T., Kudo, T., Tsuda, K.: gBoost: a mathematical programming approach to graph classification and regression. *Mach. Learn.* **75**(1), 69–89 (2009). <https://doi.org/10.1007/s10994-008-5089-z>
- Vazirgiannis, M., Malliaros, F., Nikolentzos, G.: GraphRep: boosting text mining, NLP and information retrieval with graphs. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 2295–2296 (2018)
- Yan, X., Han, J.: gSpan: graph-based substructure pattern mining. In: *Proceedings of the IEEE International Conference on Data Mining*, pp. 721–724. IEEE Press (2002)