Graph-based term weighting scheme for topic modeling

Giannis Bekoulis
LIX
École Polytechnique
Palaiseau, France
giannis.bekoulis@gmail.com

François Rousseau
LIX
École Polytechnique
Palaiseau, France

***Abstract*—LSI and LDA are widely used techniques to uncover the underlying topical structure of text. They traditionally rely on bag-of-words representation of documents and term frequency-based (TF) weighting schemes. In this paper, we represent documents as graph-of-words to capture the relationships between close words and propose the number of contexts of co-occurrences as alternative term weights (TW). Experiments with a downstream supervised task show that counting the importance of a node inside the graph results in statistically significant higher accuracy and macro-averaged F_1-score than with TF-based LSI and LDA.**

I. Introduction

Document collections are difficult to organize, explore and search due to the ever growing amount of textual information. Latent Semantic Indexing (LSI) and Latent Dirichlet Allocation (LDA) can help compress multiple terms into latent dimensions to uncover the topics of large text collections and reduce the dimensionality of the feature space in downstream NLP tasks such as text categorization, information retrieval, topic discovery, etc. In all cases, the traditional bag-of-words document representation is used, which assumes term independence and only considers the document as the context of co-occurrence between all its terms, disregarding even grammar or word order. In practice, there are stronger relationships between neighboring terms, which should be reflected in the adopted term weighting scheme.

Graphs are the natural representation to represent complex information about entities and interaction between them and we think text makes no exception. Historically, following the traditional bag-of-words representation, unigrams have been considered as the natural features and later extended to n-grams to capture some word dependency and word order. Even if we consider the n-gram model, information about the relationship between two different n-grams is ignored, i. e. it fails to capture word inversion and subset matching (e. g., article about news vs. news article). We believe that the use of a graph saves more useful information than a standard vector of frequencies and can help us overcome these issues.

In this paper, we explored the *graph-of-words* representation to challenge the TF-based weighting schemes used in LSI and LDA. The difference is that instead of counting the frequency of the term inside the document (raw TF), we count the importance of the node inside the graph. This way we penalize terms that are not well-connected while we increase the weights for terms that do co-occur a lot. By doing so, we are able to augment the unigram feature space of the learning task with weights that implicitly consider information of n-grams (short and long ones) in the document - as expressed by hop-1 neighborhoods in the graph without increasing the dimensionality of the problem. We evaluated our approach using text categorization as the downstream supervised task and topic modeling as the dimensionality reduction pre-processing step.

The rest of this paper is structured as follows. Section II reviews the related work. Section III describes our approach. Section IV defines the experimental settings and presents the results we obtained on four standard datasets. Finally, Section V concludes our paper and mentions future work.

II. Related work

We present first the related work on topic modeling and graph-based text processing.

A. Topic modeling

Topic modeling aims at extracting the hidden topics of a document from its observable words. It has been used in a wide variety of NLP tasks, e. g., information retrieval [1], word sense disambiguation [2] and sentiment analysis [3].

Latent Semantic Indexing (LSI) [4] is the earliest major approach in topic modeling, relying on singular value decomposition to find dimensions along which sets of words tend to co-occur in the same contexts, i. e. documents when traditionally applied on the document-term matrix. In order to project documents in the same word vector space, the word order is ignored and the raw term frequency (TF) is used as matrix cell value.

Later, Hofmann [5] proposed probabilistic Latent Semantic Analysis (pLSA) to represent documents as a mixture of latent topics. As opposed to a simple mixture of unigrams model that assumes that the words of every document are drawn independently from a single multinomial distribution, i. e. each document is generated from a single topic, it introduces a fixed set of hidden topics. The joint distribution for pLSA can be expressed as:

$$\prod_{d} \mathrm{P}(d) \prod_{w} \left(\sum_{z} \mathrm{P}(z|d)\, \mathrm{P}(w|z) \right)^{\mathrm{TF}(w,d)} \tag{1}$$

where d are the documents, z the latent topics, w the terms of the vocabulary and $\mathrm{TF}(w, d)$ the observed raw term frequency of term w in document d. Although Hoffman's work is a significant step towards probabilistic topic modeling, there is no probabilistic formulation at the document level: the topic mixture $\mathrm{P}(z|d)$ is conditioned on each document, which leads to a large set of individual parameters (overfitting) and is undefined for unseen documents.

Blei et al. [6] then introduced Latent Dirichlet Allocation (LDA) to overcome these limitations, which is why we explored LDA rather than pLSA in our experiments. For each topic z, we pick a word multinomial distribution Φ_z from a Dirichlet distribution with parameter β. For each document d, we pick a topic multinomial distribution Θ_d from a Dirichlet distribution with parameter α. For each word w in a document d, we first draw a topic $z_{w,d}$ from the topic distribution of d and then draw the word itself from the word distribution of that topic. The joint distribution for LDA can be expressed as:

$$\begin{gathered}\prod_{z} \mathrm{P}(\Phi_z; \beta) \prod_{d} \mathrm{P}(\Theta_d; \alpha) \\ \prod_{w} \left[\mathrm{P}(z_{w,d}|\Theta_d)\, \mathrm{P}(w|\Phi_{z_{w,d}}) \right]^{\mathrm{TF}(w,d)}\end{gathered} \tag{2}$$

Note that in both Equation 1 and 2, we took the product over words thanks to the term independence assumption made in the bag-of-words model. We also expressed the product over all words of the vocabulary instead of just the sequence of words in each document so as to highlight the raw term frequency (TF). Hence, both LSI and LDA naturally consider bag-of-words as document representation and TF as term weighting scheme, but it does not have to be.

A lot of extensions have been inspired by the aforementioned models such as Transductive LSI [7], local LSI [8], Relevancy Weighted LSI [9], NetPLSA [10] as well as several variations of LDA [11], [3], [12], [13].

B. Graph-based text processing

Graph-modeling is an alternative way of representing information, which clearly highlights relationships of nodes among vertices. Text can be represented as a graph in various ways. We refer to Blanco and Lioma [14] for an in-depth review of all the graph representations of a document in the field of information retrieval. Graph representation of documents is common in other text analysis task, including text summarization [15], keyword detection [16], text categorization [17] and word sense disambiguation [18] among others. A vertex of a graph can represent a sentence, a word or even a character. Similarly, an edge represents a meaningful relation between two vertices, either linguistic (e. g., syntactic or semantic) or statistical (e. g., co-occurrence) depending on the use case. Edges can be directed or undirected, weighted (e. g., frequency or strength of the relation) or unweighted.

III. Our approach

In this section, we present our approach that consists in alternative document representation and term weighting scheme as input of LSI and LDA.

A. Graph-based document representation

We model documents as graph-of-words to capture some relationships between co-occurring words [17]. We first apply standard text preprocessing: stemming, stop word removal and retention of the 5,000 most frequent terms over the collection [3]. The remaining set of unique terms constitute the vertices of the graph. The edges are then drawn between terms co-occurring within a fixed sliding window over the processed text, ignoring self-loops. Formally, each document d is represented by a graph $\mathcal{G}_d = (\mathcal{V}_d, \mathcal{E}_d)$, where V_d are nodes that correspond to the remaining set of terms and E_d are edges that depict relationships between the terms within a fixed-size sliding window of size w. That is, for all the terms that co-occur within the window, we add edges between the corresponding nodes (note that, the windows are overlapping starting from the first term of the document; at each step we simply remove the first term and add the new one from the document). The underlying assumption is that all the words that exist inside a document have some relationships with the others, modulo a window size, outside of which the relationship is not taken into consideration.

We illustrate the process in Figure 1 above where the stop words in italic in the upper box are filtered out. We considered both undirected and directed edges, using the natural flow of the text, as well as unweighted and weighted edges, counting the number of times two terms co-occur. As opposed to a bag-of-words that considers the document as the sole context of co-occurrence between its words, this encodes local interactions as well. To still be able to re-use the existing framework for LSI and LDA, we chose to modify the term weighting scheme using the graph while still making the term independence assumption rather than include the graph structure in the term-document matrix or the generative model.

B. Degree-based term weighting scheme

Table I illustrates the node degree-based term weights (TW) in comparison to the traditional raw term frequency (TF) for the most frequent terms in the toy example from Figure 1. TW_u corresponds to the number of edges that a vertex is connected to, regardless of either the direction or the weight, i. e. the number of *distinct* contexts of co-occurrence. TW_{uin} and TW_{uout} are extensions where we only consider one direction in the co-occurrence. TW_w considers *all* contexts of co-occurrence. Even in the weighted case, the scaling is not linear (in the window size) because we do not encode self-loops, i. e. co-occurrence of a term with itself, which happens even more frequently after preprocessing. Therefore, TW can be interpreted as a dampened version of TF where

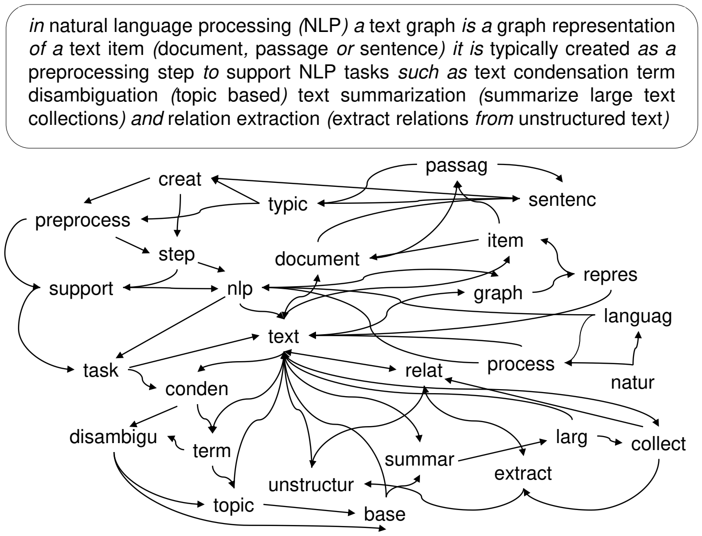

Fig. 1. Directed graph-of-words representation of the text in the upper box. Nodes correspond to unique terms and edges to co-occurrences within a window of size 3.

TABLE I
TERM FREQUENCY AND WEIGHTS FOR THE TOP TERMS (TF ≥ 2) OF THE DOCUMENT FROM FIGURE 1.

terms	TF	TW_u	TW_{uin}	TW_{uout}	TW_w
text	6	16	11	8	22
extract	2	3	2	2	6
graph	2	3	2	2	6
nlp	2	7	4	3	8
relat	2	4	3	3	8
summar	2	3	2	2	6

contexts of co-occurrence with other words matter more than just occurrence of a word.

For instance, let's consider the term "summar" from Figure 1. TW_{uin} and TW_{uout} are term weights that we consider in unweighted directed graphs. Table II depicts TW_{uin} that counts the number of incoming edges that a vertex is connected to, ignoring self-loops (e. g., excluding summar $\rightarrow$ summar), direction and frequency of the relation (weight) as well as TW_{uout}, counting only the number of outgoing edges of a vertex. We consider TW_u in unweighted and undirected graphs (see Table III). It is worth mentioning that $\text{TW}_{uin} + \text{TW}_{uout} \neq \text{TW}_u$, since we capture co-occurrence

TABLE II
TW_{uin} AND TW_{uout} FOR THE TERM "SUMMAR" OF THE DOCUMENT FROM FIGURE 1.

	left node	direction	right node	weight
TW_{uin}	based	$\rightarrow$	summar	1
	text	$\rightarrow$	summar	1
TW_{uout}	summar	$\rightarrow$	large	1
	summar	$\rightarrow$	text	1

TABLE III
TW_u AND TW_w FOR THE TERM "SUMMAR" OF THE DOCUMENT FROM FIGURE 1.

	left node	direction	right node	weight
TW_u	based	$\leftrightarrow$	summar	1
	text	$\leftrightarrow$	summar	1
	summar	$\leftrightarrow$	large	1
TW_w	based	$\leftrightarrow$	summar	1
	text	$\leftrightarrow$	summar	2
	summar	$\leftrightarrow$	large	2
	summar	$\leftrightarrow$	text	1

of two terms whatever the respective order between them is but without taking into account their frequency and self-loops. Specifically, the "text$\leftrightarrow$summar" and "summar$\leftrightarrow$text" relationships are considered as if only one edge exist between them in the TW_u case. Finally, in TW_w, the higher the number of co-occurrences of two terms in the document, the higher the weight of the corresponding edge, as we can see in Table III.

IV. EXPERIMENTS

In this section, we present how we evaluated our approach and discuss the results we obtained.

A. Dimensionality reduction and evaluation

We evaluated the performance of topic modeling with graph-of-words representation as input for text categorization, which is the task of assigning to a document a label among a set of predefined ones. The standard approach consists in representing documents as word vectors and learn a linear classifier in this vector space [19]. The feature space is potentially of very high dimension (the size of the vocabulary) and

TABLE IV

TEST ACCURACY AND MACRO-AVERAGE F_1 SCORES. BOLD FONT MARKS THE BEST PERFORMANCE IN A BLOCK COLUMN. * INDICATES STATISTICAL SIGNIFICANCE OF IMPROVEMENT IN ACCURACY OVER THE TF BASELINE OF THE SAME BLOCK USING THE MICRO SIGN TEST ($p < 0.05$).

	method \ dataset	20ng Acc	20ng F_1	R8 Acc	R8 F_1	R52 Acc	R52 F_1	BBC Acc	BBC F_1
LSI	TF (baseline)	0.7125	0.7032	0.9246	0.7582	0.8298	0.2696	0.8966	0.8953
LSI	TW_u (degree)	0.7055	0.6982	0.9223	**0.7718**	**0.8462***	**0.3944**	0.9326*	0.9331
LSI	TW_{uin} (in degree)	**0.7614***	**0.7503**	0.9278	0.7331	0.8166	0.1997	**0.9371***	**0.9353**
LSI	TW_{uout} (out degree)	0.7398*	0.7306	**0.9333***	0.7699	0.8368	0.2818	**0.9371***	0.9351
LSI	TW_w (weighted)	0.6869	0.6779	0.9141	0.7511	0.7960	0.3380	0.9191*	0.9145
LDA	TF (baseline)	0.7194	0.7031	0.7958	0.3594	0.6783	0.0504	0.8315	0.8267
LDA	TW_u (degree)	0.7388*	0.7248	0.7985	0.3778	0.6807	0.0551	0.8584	0.8583
LDA	TW_{uin} (in degree)	0.7325*	0.7229	0.7775	0.3085	0.6632	0.0439	0.8494	0.8461
LDA	TW_{uout} (out degree)	0.7198	0.7065	0.7967	0.3909	0.6791	0.0553	**0.8876***	**0.8852**
LDA	TW_w (weighted)	**0.7392***	**0.7272**	**0.8164***	**0.4327**	**0.6967***	**0.0673**	0.8607*	0.8599

also rather sparse. To reduce its dimension and also overcome text-specific issues such as synonymy, topic modeling has been proposed as a dimensionality reduction preprocessing step where the classification then happens in the denser topic space [9], [3].

Moreover, because text categorization is a supervised task, this allows us to quantitatively assess the effectiveness of a given topic modeling approach compared to another one or a change in the input term weighting scheme as opposed to the evaluation methods reviewed by Wallach et al. [20] for instance that estimate the probability of only tens of held-out documents given a trained model.

B. Datasets

We evaluated our approach on four multi-class text categorization datasets: (1) 20 Newsgroups (20ng) partitioned evenly across 20 different newsgroups, Reuters-21578 restricted to (2) its 52 most frequent categories (R52) and (3) its 8 most frequent categories (R8) [21] and (4) BBC news (BBC) [22]. Details are presented in Table V.

C. Implementation

We have developed our methods in *Python* using the *NetworkX* [23] library for the graph representation. For LDA topic models, we used the implementation of *LDA*[1]. Finally, for classification, LSI and standard term frequency we used *scikit-learn* [24], a standard Python machine learning library.

[1]https://pypi.python.org/pypi/lda

TABLE V

DESCRIPTION OF THE DATASET COLLECTIONS.

dataset	# of train docs	# of test docs	total # of docs
20ng	11,293	7,528	18,821
R8	6,532	2,568	9,100
R52	5,485	2,189	7,674
BBC	1,780	445	2,225

D. Experimental setup

Following Zelikovitz and Hirsh [25], [7] and Maas et al. [3], we used the largest collection of documents (20ng) to learn the topics, considering the whole dataset as unlabeled background knowledge, and then we used these topics on all datasets for dimensionality reduction. Because all the datasets are about news and somewhat related, learning the topics on the largest collection results in smoother probability estimates for LDA for instance, even if the topics are more general than if they were learnt on each separate dataset. Indeed, in our experiments, the effectiveness was improved for both the bag-of-words and the graph-of-words representation on all datasets without affecting the relative ranking of the various models.

Regarding the term weighting scheme, we considered only the raw term frequencies (TF) and node degrees (TW), without the additional IDF term as LDA is modeled using multinomial distributions and for LSI, the raw versions yielded to better performances on these datasets.

Following Wallach et al. [26], we trained LDA with 100 topics and 3,000 iterations and trained LSI so as to retain 85% of the variance. The assignment of words to topics remains invariant if we increase the number of topics for the size of our collections and they also suggest to use a larger number of topics and iterations rather than too few.

We then trained a linear SVM in each topic space. To prevent overfitting, we tuned the regularization hyper-parameter using cross-validation on the training set of 20ng (10^{-3} for TF, 10^{-5} for TW_u, 10^{-2} for TW_{uin}, 10^{-3} for TW_{uout} and 10^{-7} for TW_w). For our experiments we have considered window of size $w = 3$, since this was performing well compared to other values, although in the field of information retrieval as reported by Rousseau and Vazirgiannis [27], $w = 4$ was the best performing sliding window size.

To evaluate the effectiveness of our approach, we used the standard text categorization metrics: accuracy and macro-average F_1 score (which takes into account the skewed label class distribution). Statistical significance of improvement in accuracy over the TF baseline was assessed using the micro sign test ($p < 0.05$) [28].

E. Results

Table IV presents our results in text categorization on all four datasets. For both LSI and LDA, we compare the raw term frequency (TF) with all four versions of the node degree-based term weight (TW) as input term weighting scheme.

Overall, TW performs better than TF, which means that taking into account the co-occurrences is more beneficial to topic modeling than simply considering the occurrences of a word (the graph-of-words representation ignores self-loops, i. e. self co-occurrences in the processed text). Moreover, by encoding these local interactions in the term weighting scheme, we can re-use the traditional LSI and LDA definitions. Note that LSI seems to produce better results than LDA, but this is not the purpose of our work, which focused on improving each model separately. Similarly, text categorization is just a proxy to quantitatively assess improvements.

For LSI, unweighted node degrees generally yield better performances, which means that it is the number of *distinct* contexts of co-occurrences that matters the most, in particular the number of distinct words co-occurring *after* a given word (the next and the next but one words), i. e. the out degree (TW_{uout}).

For LDA, the weighted node degree (TW_{w}) consistently yields the best performances, which means that weighting the co-occurrences by their frequency is better taken into account by LDA than LSI.

V. Conclusions and future work

In this paper, we explored an alternative term weighting scheme for topic modeling to challenge the traditional TF-based approach. Using the graph-of-words representation that encodes local interactions between terms, we proposed to use the number of contexts of co-occurrences instead of the plain term frequency, counting the number of distinct contexts for LSI and weighting by the frequency of each context for LDA.

In future work, we might explore topic coherence for LDA in the graph-of-words context and consider a graph normalization scheme over the whole document collection (IDW) equivalent to the IDF variant in the bag-of-words model for LSI.

References

[1] X. Wei and W. B. Croft, "Lda-based document models for ad-hoc retrieval," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006, pp. 178–185.

[2] J. L. Boyd-Graber, D. M. Blei, and X. Zhu, "A topic model for word sense disambiguation." in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, ser. EMNLP-CoNLL '07, 2007, pp. 1024–1033.

[3] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, ser. HLT '11, 2011, pp. 142–150.

[4] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[5] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '99, 1999, pp. 50–57.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[7] S. Zelikovitz and H. Hirsh, "Transductive lsi for short text classification problems." in *Proceedings of FLAIRS conference*, 2004, pp. 556–561.

[8] H. Schütze, D. A. Hull, and J. O. Pedersen, "A comparison of classifiers and document representations for the routing problem," in *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '95, 1995, pp. 229–237.

[9] T. Liu, Z. Chen, B. Zhang, W.-y. Ma, and G. Wu, "Improving text classification using local latent semantic indexing," in *Proceedings of the 4th IEEE International Conference on Data Mining*, ser. ICDM '04, 2004, pp. 162–169.

[10] Q. Mei, D. Cai, D. Zhang, and C. Zhai, "Topic modeling with network regularization," in *Proceedings of the 17th international conference on World Wide Web*, 2008, pp. 101–110.

[11] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '09, 2009, pp. 248–256.

[12] R. Das, M. Zaheer, and C. Dyer, "Gaussian lda for topic models with word embeddings," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 2015, pp. 795–804.

[13] D. Q. Nguyen, R. Billingsley, L. Du, and M. Johnson, "Improving topic models with latent feature word representations," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 299–313, 2015.

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

[15] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research*, vol. 22, pp. 457–479, 2004.

[16] J. Liu, J. Wang, and C. Wang, "A text network representation model," in *Proceedings of the 5th International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 4, 2008, pp. 150–154.

[17] F. Rousseau, E. Kiagias, and M. Vazirgiannis, "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*, ser. ACL-IJCNLP '15, vol. 1, 2015, pp. 1702–1712.

[18] R. Mihalcea and P. Tarau, "TextRank: Bringing Order into Texts," in *Proceedings of the 9th Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '04, 2004, pp. 404–411.

[19] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of the 14th International Conference on Machine Learning*, ser. ICML '97, 1997, pp. 412–420.

[20] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno, "Evaluation Methods for Topic Models," in *Proceedings of the 26th International Conference on Machine Learning*, ser. ICML '09, 2009, pp. 1105–1112.

[21] F. Debole and F. Sebastiani, "An Analysis of the Relative Hardness of Reuters-21578 Subsets: Research Articles," *Journal of the American Society for Information Science and Technology*, vol. 56, no. 6, pp. 584–596, 2005.

[22] D. Greene and P. Cunningham, "Practical solutions to the problem of diagonal dominance in kernel document clustering," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06, 2006, pp. 377–384.

[23] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using networkx," in *Proceedings of the 7th Python in Science Conferences*, ser. SciPy 2008, 2008, pp. 11–15.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[25] S. Zelikovitz and H. Hirsh, "Using lsi for text classification in the presence of background text," in *Proceedings of the 10th International Conference on Information and Knowledge Management*, ser. CIKM '01, 2001, pp. 113–118.

[26] H. M. Wallach, D. M. Mimno, and A. McCallum, "Rethinking lda: Why priors matter," in *Advances in Neural Information Processing Systems 22*, ser. NIPS '09, 2009, pp. 1973–1981.

[27] F. Rousseau and M. Vazirgiannis, "Graph-of-word and tw-idf: New approach to ad hoc ir," in *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, ser. CIKM '13, 2013, pp. 59–68.

[28] Y. Yang and X. Liu, "A re-examination of text categorization methods," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '99, 1999, pp. 42–49.