More Discriminative Sentence Embeddings via Semantic Graph Smoothing

Chakib Fettal[1,2] and **Lazhar Labiod**[1] and **Mohamed Nadif**[1]
[1]Centre Borelli, Université Paris Cité, 75006 Paris, France
[2]Informatique Caisse des Dépôts et Consignations, 75013 Paris, France
{firstname.lastname}@u-paris.fr

Abstract

This paper explores an empirical approach to learn more discriminantive sentence representations in an unsupervised fashion. Leveraging semantic graph smoothing, we enhance sentence embeddings obtained from pretrained models to improve results for the text clustering and classification tasks. Our method, validated on eight benchmarks, demonstrates consistent improvements, showcasing the potential of semantic graph smoothing in improving sentence embeddings for the supervised and unsupervised document categorization tasks.

1 Introduction

Text categorization, also known as document categorization, is a natural language processing (NLP) task that involves arranging texts into coherent groups based on their content. It has many applications such as spam detection (Jindal and Liu, 2007), sentiment analysis (Melville et al., 2009), content recommendation (Pazzani and Billsus, 2007), etc. There are two main approaches to text categorization: classification (supervised learning) and clustering (unsupervised learning). In text classification, the process involves training a model using a labeled dataset, where each document is associated with a specific category. The model learns patterns and relationships between the text features and the corresponding categories during the training phase. Text clustering, however, aims to group similar documents together without prior knowledge of their categories. Unlike text classification, clustering does not require labeled data. Instead, it focuses on finding inherent patterns and similarities in the text data to create clusters.

In the field of NLP, pretrained models have attained state-of-the-art performances in a variety of tasks (Devlin et al., 2019; Liu et al., 2019; Reimers and Gurevych, 2019), one of which is text classification. In spite of that, text clustering using such models did not garner significant attention. To this day most text clustering techniques use the representations of texts generated by some pretrained model such as Sentence-BERT (Reimers and Gurevych, 2019) and often use classical clustering approaches such as k-means to obtain a partition of the texts. This is done without any fine-tuning due to the unsupervised nature of the clustering problem.

Recently, graph filtering has appeared as an efficient and effective technique for learning representations for attributed network nodes. The effectiveness of this technique has made it a backbone for popular deep learning architectures for graphs such as the graph convolutional network (GCN) (Kipf and Welling, 2016). Simplified versions of this deep architecture have been proposed wherein the learning of large sets of weights has been deemed unnecessary. Their representation learning scheme works similar to Laplacian smoothing and, by extension, graph filtering. We can give as examples of these simplified techniques the simple graph convolution (SGC) (Wu et al., 2019), and the simple spectral graph convolution (S²GC) (Zhu and Koniusz, 2020). Some researchers used GCNs for the task of text classification. Yao et al. (2019) proposed TextGCN which is GCN with a custom adjacency matrix built from word PMI and the TF-IDF of the documents with the attributes being word count vectors. Lin et al. (2021) proposed BertGCN which is similar to TextGCN with the difference that they use BERT representation for the GCN and combine their training losses. The issue is that these approaches are not suitable for learning unsupervised representations since labels are needed. This is a significant limitation towards their use in unsupervised tasks. Recently some graph-based unsupervised approaches were proposed to deal with text data represented using document-term matrices (Fettal et al., 2022, 2023).

In this paper, we propose to use the concept of graph smoothing/filtering, which is the main

component accredited with the success of GCNs (Defferrard et al., 2016; Kipf and Welling, 2016; Li et al., 2018), to semantically "fine-tune" the representations obtained via sentence embedding models to help traditional clustering and classification algorithms better distinguish between semantically different texts and group together texts which have similar meanings, all in an unsupervised manner. To do this, we build a graph with respect to the text which describes the semantic similarity between the different documents based on the popular cosine similarity measure. Our approach yields almost systematic improvement when using filtering on the textual representations as opposed to using them without filtering in both facets of document categorization: classification and clustering. Experiments on eight popular benchmark datasets support these observations.

The code for the experiments is available at [1].

2 Background: Graph Filtering and Smoothing

Graph Signal Processing (Shuman et al., 2013; Ortega et al., 2018) provides a framework to analyze and process signals defined on graphs, by extending traditional signal processing concepts and tools to the graph domain. This allows for the representation and manipulation of signals in a way that is tailored to the specific structure of the graph. In what follows we refer to matrices in boldface uppercase and vectors in boldface lowercase.

Graph Signals Graph signals are mappings from the set of vertices to the real numbers. A graph signal for a given graph $\mathcal{G}$ can be represented using vector $\mathbf{f} = [f(v_1), \ldots, f(v_n)]^\top$ such that $f : \mathcal{V} \to \mathbb{R}$ is a real-valued function on the vertex set. The smoothness of a signal $\mathbf{f}$ over graph $\mathcal{G}$ can be characterized using the Laplacian quadratic form associated with Laplacian $\mathbf{L}$:

$$\mathbf{f}^\top \mathbf{L}\mathbf{f} = \frac{1}{2}\sum_{i,j} a_{ij}(\mathbf{f}_i - \mathbf{f}_j)^2. \tag{1}$$

These signals can be high dimensional and can represent many kinds of data. In our case, signals will represent text embeddings.

Graph Filters Smoother graph signals can be obtained by minimizing the quantity described in Formula (1). That is the goal of graph filters and the filtering is generally done from a spectral perspective. A specific class of filters that additionally has an intuitive interpretation from a vertex perspective is that of the polynomial filters. When the filter is a P-th order polynomial of the form $\hat{h}(\mathbf{L}) = \sum_{m=0}^{p} \theta_m \mathbf{L}^m$, the filtered signal at vertex i, is a linear combination of the components of the input signal at vertices within a P-hop local neighborhood of vertex i:

$$\mathbf{f}_i^{\text{out}} = \alpha_{ii}\mathbf{f}_i^{\text{in}} + \sum_{j \in N(i,p)} \alpha_{ij}\mathbf{f}_j^{\text{in}} \tag{2}$$

where $N(i,p)$ is the P-th order neighborhood of vertex i. It is possible to then make the correspondence with a polynomial filter (from a spectral perspective) as follows:

$$\alpha_{ij} = \sum_{m=d_{\mathcal{G}}(i,j)}^{p} \theta_m (\mathbf{L}^m)_{ij} \tag{3}$$

where $d_{\mathcal{G}}$ is the shortest distance between node i and j. Several polynomial filters have been proposed in the literature such as the ones associated with Simple Graph Convolution (SGC) (Wu et al., 2019), simple spectral Graph Convolution (S²GC) (Zhu and Koniusz, 2020), approximate personalized propagation of neural predictions (APPNP) (Gasteiger et al., 2018) and Decoupled Graph Convolution (DGC) (Wang et al., 2021).

3 Proposed Methodology: Smoothing Sentence Embeddings

In this paper, we theorize that smoothing sentence embeddings with a semantic similarity graph can help supervised and unsupervised categorization models better differentiate between the similar and dissimilar documents, leading to performance gains. A common choice for quantifying semantic similarity of text is the cosine similarity; given two sentence embedding vectors $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^d$ we have

$$\cos(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i^\top \mathbf{x}_j}{\|\mathbf{x}_i\|\|\mathbf{x}_j\|}.$$

We build a k-nearest neighbors connectivity graph which we denote $\mathcal{G}$ based on this similarity measure i.e. a graph for which each node has exactly k neighbors and whose edge weights are all equal to one. We characterize the graph $\mathcal{G}$ using its adjacency matrix $\mathbf{A}$, we denote its Laplacian as $\mathbf{L}$.

[1] https://github.com/chakib401/smoothing_sentence_embeddings

Given the adjacency matrix, a standard trick to obtain better node representations consists in adding a self-loop

$$\hat{\mathbf{A}} = \mathbf{A} + \lambda \mathbf{I} \tag{4}$$

where λ is a hyperparameter controlling the number of self-loops. As such in what follows we consider the symmetrically normalized version of $\hat{\mathbf{A}}$, that is

$$\mathbf{S} = \hat{\mathbf{D}}^{-1/2} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-1/2}. \tag{5}$$

Now given a node embedding matrix $\mathbf{X}$ and the previous semantic similarity graph. We consider four polynomial graph filters whose propagation rules we describe in Table 1.

Table 1: The propagation rules associated with the different polynomial filters. $\mathbf{H}^{(0)}$ is the $\mathbf{X}$. P is the propagation order. α and T are filter-specific hyperparameters.

Filter	Propagation Rule
F_{SGC}	$\mathbf{H}^{(p+1)} \leftarrow \mathbf{S}\mathbf{H}^{(p)}$
$F_{\text{S}^2\text{GC}}$	$\mathbf{H}^{(p+1)} \leftarrow \mathbf{H}^{(p)} + \mathbf{S}\mathbf{H}^{(p)}$
F_{APPNP}	$\mathbf{H}^{(p+1)} \leftarrow (1-\alpha)\mathbf{S}\mathbf{H}^{(p)} + \alpha\mathbf{H}^{(0)}$
F_{DGC}	$\mathbf{H}^{(p+1)} \leftarrow (1-\frac{T}{P})\mathbf{H}^{(p)} + \frac{T}{P}\mathbf{S}\mathbf{H}^{(p)}$

4 Experiments

In this section we evaluate our semantically smoothed representations obtained through four filters on two tasks, clustering and classification, with respect to the original representations obtained from SentenceBERT (Reimers and Gurevych, 2019) as well as two large language models baselines: BERT and RoBERTa.

4.1 Datasets and Metrics

We use eight benchmark datasets of varying sizes and number of clusters, and we report their summary statistics in Table 2. For the metrics, in the supervised context, we use the F1 score as the quality metric while in the unsupervised context we use the adjusted rand index (ARI) (Hubert and Arabie, 1985) and the adjusted mutual information (AMI) (Vinh et al., 2009).

4.2 Experimental Settings

For the classification task, we use a random stratified 64%-16%-20% train-val-test split. We also tune the hyperparameters k of the k-nn graph, order of propagation P, the parameter λ and the filter specfic parameters α and T. For the clustering task, we use $k = 10$ for the k-nn graph, set $P = 2$ as the propagation order, $\lambda = 1$, $\alpha = 0.1$ and $T = 5$. We report the averages of the metrics as well as their standard deviations over 10 runs (for the classification task, we omit standard deviation due to them being insignificant).

Table 2: Summary statistics of the datasets. Balance refers to the ratio of the most frequent class over the least frequent class. Length refers to the average sentence length in the corpus.

Dataset	Docs	Classes	Balance	Length
20News	18,846	20	1.6	221
DBpedia	12,000	14	1.1	46
AGNews	8,000	4	1.1	31
BBCNews	2,225	5	1.3	384
Classic3	3,891	3	1.4	152
Classic4	7,095	4	3.9	107
R8	7,674	8	76.9	65
Ohsumed	7,400	23	61.8	135

4.3 Experimental Results

Clustering Results We compare the results of the k-means algorithm (kM) applied on Sentence-BERT (we refer to it as SBERT or SB) embeddings with and without the different filtering operations. Note that instead of using kM we can use any other clustering algorithms including variants of kM such as k-means++ (Arthur and Vassilvitskii, 2007) and entropy kM (Chakraborty et al., 2020). In addition to this, we add a baseline which uses an ensemble technique (Ait-Saada et al., 2021) on the layer outputs of the word embedding of BERT and RoBERTa, this method improves over considering a single layer or taking the mean. We report the clustering results in Table 3. The filtering operation systematically leads to better results on the benchmark with respect to the filterless clustering scheme on all datasets we have used. These increases are statistically significant in most cases. It also significantly beats the ensemble approach on most datasets.

Classification Results Similar to the clustering setting, we compare results from a Logistic Regression (LR) applied on the original sentence embeddings with and without the filtering operation we introduced. We also use fine-tuned BERT and RoBERTa (2 epochs) as baselines; we use the base versions due to computational restrictions. We report the results in Table 4. We see that this op-

Table 3: Clustering results in terms of AMI and ARI on the eight datasets. The best results are highlighted in bold. If our best performing variant outperforms the best comparative method in a statistically significant matter (t-test at a confidence level of 95%), we highlight it in blue.

	20News		AGNews		BBCNews		Classic3	
	AMI	ARI	AMI	ARI	AMI	ARI	AMI	ARI
$ENS_{BERT\text{-}base}$	37.5 ±2.5	15.3 ±1.7	54.1 ±3.6	51.4 ±5.8	81.0 ±5.5	80.0 ±8.5	98.6 ±0.1	99.4 ±0.0
$ENS_{BERT\text{-}large}$	46.1 ±0.7	21.4 ±0.6	58.5 ±2.8	58.2 ±5.9	86.0 ±3.5	86.5 ±6.3	98.4 ±0.2	99.3 ±0.1
$ENS_{RoBERTa\text{-}base}$	37.5 ±1.4	15.9 ±1.8	55.9 ±4.1	52.1 ±4.1	80.0 ±5.3	77.2 ±9.4	98.4 ±0.1	99.3 ±0.1
$ENS_{RoBERTa\text{-}large}$	48.0 ±0.8	23.2 ±1.2	56.7 ±4.6	52.8 ±5.1	85.8 ±3.8	85.1 ±7.2	98.7 ±0.1	99.4 ±0.1
SBERT+kM	62.9 ±0.3	47.4 ±1.0	57.9 ±0.1	60.5 ±0.1	90.8 ±0.2	93.0 ±0.1	96.0 ±0.1	97.6 ±0.1
SB+F_{SGC}+kM	65.4 ±0.4	49.1 ±1.1	**60.6 ±0.1**	62.4 ±0.3	90.6 ±0.1	92.9 ±0.1	98.8 ±0.0	99.5 ±0.0
SB+F_{S^2GC}+kM	64.9 ±0.4	49.0 ±1.1	60.1 ±0.2	62.2 ±0.2	**90.9 ±0.1**	**93.1 ±0.1**	98.3 ±0.0	99.2 ±0.0
SB+F_{APPNP}+kM	65.4 ±0.4	**49.8 ±1.2**	**60.6 ±0.0**	**62.5 ±0.0**	90.6 ±0.1	92.9 ±0.1	98.5 ±0.0	99.3 ±0.0
SB+F_{DGC}+kM	**65.6 ±0.7**	48.8 ±1.0	60.5 ±1.5	60.5 ±2.2	90.2 ±0.1	92.5 ±0.1	**99.1 ±0.0**	**99.6 ±0.0**

	Classic4		DBpedia		Ohsumed		R8	
	AMI	ARI	AMI	ARI	AMI	ARI	AMI	ARI
$ENS_{BERT\text{-}base}$	71.4 ±3.5	49.0 ±4.0	73.4 ±2.5	51.0 ±4.0	15.2 ±1.0	9.1 ±1.2	35.3 ±2.0	22.7 ±2.4
$ENS_{BERT\text{-}large}$	73.0 ±1.8	51.1 ±3.2	72.4 ±2.1	47.2 ±4.2	16.1 ±0.9	9.3 ±0.7	35.7 ±3.5	22.8 ±3.1
$ENS_{RoBERTa\text{-}base}$	72.1 ±4.7	51.0 ±4.1	74.2 ±2.6	52.5 ±4.7	17.5 ±0.7	11.4 ±0.8	25.6 ±1.0	13.6 ±1.2
$ENS_{RoBERTa\text{-}large}$	74.1 ±3.5	52.5 ±3.9	72.5 ±2.5	49.0 ±4.4	19.4 ±0.7	12.7 ±0.7	42.4 ±5.6	32.9 ±9.2
SBERT+kM	84.5 ±0.1	86.2 ±0.1	86.0 ±1.4	80.0 ±3.1	39.3 ±0.7	23.5 ±1.2	63.1 ±1.8	45.5 ±3.7
SB+F_{SGC}+kM	85.8 ±2.8	85.6 ±7.4	85.6 ±1.0	78.5 ±2.7	**41.8 ±0.5**	**25.2 ±1.0**	**65.6 ±0.5**	49.0 ±0.6
SB+F_{S^2GC}+kM	86.0 ±0.0	86.9 ±0.0	**86.6 ±1.2**	**80.4 ±2.8**	41.0 ±0.8	24.5 ±1.5	64.8 ±1.1	47.8 ±0.7
SB+F_{APPNP}+kM	86.2 ±0.0	87.0 ±0.0	85.8 ±1.0	78.9 ±2.7	41.6 ±0.7	24.9 ±1.5	65.1 ±1.6	48.5 ±1.0
SB+F_{DGC}+kM	**86.9 ±0.0**	**87.7 ±0.0**	85.4±1.0	78.4 ±2.2	**41.8 ±0.7**	24.8±1.7	**65.6 ±0.5**	**49.3 ±0.4**

Table 4: Classification results in terms of F1 score on the eight data sets.

	20News	R8	AGNews	BBCNews	Classic3	Classic4	DBpedia	Ohsumed
$BERT_{base}$	80.7	89.94	**89.78**	95.51	**100.0**	**98.58**	97.84	56.48
$RoBERTa_{base}$	85.48	89.42	88.06	96.73	99.16	96.47	**98.22**	58.11
SBERT+LR	83.35	90.22	86.25	98.62	99.61	98.19	97.33	62.87
SB+F_{APPNP}+LR	**87.54**	90.9	87.9	**99.06**	99.75	98.36	97.14	**67.6**
SB+F_{DGC}+LR	87.11	90.08	87.59	98.19	99.61	98.52	97.38	67.09
SB+F_{S^2GC}+LR	87.36	**91.19**	88.33	98.62	99.75	98.19	97.26	67.42
SB+F_{SGC}+LR	87.26	89.22	88.05	**99.06**	99.61	98.32	97.01	67.05

eration leads to better performances on the classification task on the majority of the datasets with respect to the filterless Sentence-BERT but this performance increase is not as pronounced as for the clustering task. We also see that the representations we learn lead to competitive results with respect to BERT and RoBERTa despite Sentence-BERT not being suited to classification.

Statistical Significance Testing Using the Bonferroni-Dunn post-hoc mean rank test (Demšar, 2006), we analyze the average ranks of the clustering and classification over the Sentence-BERT representations with and without filtering in terms of AMI and ARI, for the clustering task, as well as the F1 score for the classification task on the eight datasets. Figure 1 shows that the clustering and classification results when using the pro-

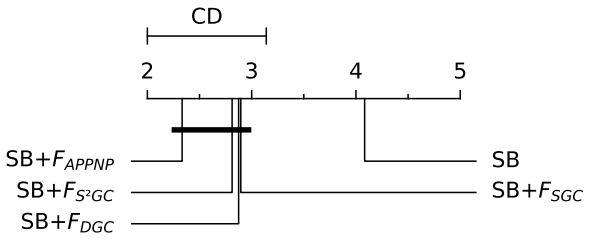

Figure 1: Bonferroni-Dunn average rank test at a confidence level of 95%.

posed semantically smoothed representations are statistically similar and that they all outperform the Sentence-BERT variant with no filtering in a statistically significant manner at a confidence level of 95%.

5 Conclusion

We proposed a simple yet effective empirical approach that consists in using similarity graphs in an unsupervised manner to smooth sentence embeddings obtained from pretrained models in a semantically aware manner. The systematic improvements in performance on both clustering and classification tasks on several benchmark datasets of different scales and balance underscore the effectiveness of using semantic graph smoothing to improve sentence representations.

6 Limitations

The main limitation of our approach is the additional computational complexity entailed by creating the k-nn graph from the data, performing the smoothing. Add to that, the hyperparameter tuning that is necessary for the classification task. While this increase is in no way prohibitive even for large datasets, a performance-speed compromise is to be considered.

References

Mira Ait-Saada, François Role, and Mohamed Nadif. 2021. How to leverage a multi-layered transformer language model for text clustering: an ensemble approach. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2837–2841.

David Arthur and Sergei Vassilvitskii. 2007. K-means++ the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035.

Saptarshi Chakraborty, Debolina Paul, Swagatam Das, and Jason Xu. 2020. Entropy weighted power k-means clustering. In *International conference on artificial intelligence and statistics*, pages 691–701. PMLR.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29.

Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, 7:1–30.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Chakib Fettal, Lazhar Labiod, and Mohamed Nadif. 2022. Subspace co-clustering with two-way graph convolution. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 3938–3942.

Chakib Fettal, Lazhar Labiod, and Mohamed Nadif. 2023. Boosting subspace co-clustering via bilateral graph convolution. *IEEE Transactions on Knowledge and Data Engineering*.

Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Predict then propagate: Graph neural networks meet personalized pagerank. In *International Conference on Learning Representations*.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification*, 2:193–218.

Nitin Jindal and Bing Liu. 2007. Review spam detection. In *Proceedings of the 16th international conference on World Wide Web*, pages 1189–1190.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Yuxiao Lin, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Jiwei Li, and Fei Wu. 2021. Bertgcn: Transductive text classification by combining gcn and bert. *arXiv preprint arXiv:2105.05727*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019.

Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Prem Melville, Wojciech Gryc, and Richard D Lawrence. 2009. Sentiment analysis of blogs by combining lexical knowledge with text classification. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1275–1284.

Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. 2018. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828.

Michael J Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In *The adaptive web: methods and strategies of web personalization*, pages 325–341. Springer.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2009. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th annual international conference on machine learning*, pages 1073–1080.

Yifei Wang, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. 2021. Dissecting the diffusion process in linear graph convolutional networks. *Advances in Neural Information Processing Systems*, 34:5758–5769.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7370–7377.

Hao Zhu and Piotr Koniusz. 2020. Simple spectral graph convolution. In *International conference on learning representations*.