INDUCING DOCUMENT REPRESENTATIONS WITH GRAPH-BASED METHODS: A BLUEPRINT

Boshko Koloski[1,2*] **Marko Pranjić** [1,2] **Nada Lavrač** [1,2] **Blaž Škrlj**[1*], **Senja Pollak** [1]
* equal contribution
[1] Jozef Stefan Institute, Ljubljana, Slovenia, Ljubljana, Slovenia
[2] Jozef Stefan International Postgraduate School, Ljubljana, Slovenia
[3] School of Engineering and Management, University of Nova Gorica, Vipava, Slovenia
{name.lastname@ijs.si}

ABSTRACT

Representing documents in continuous numerical spaces is one of the key tasks in NLP. Contemporary state-of-the-art techniques leverage large neural networks and learn the document representations self-supervised. However, while these approaches excel at learning contextual word representations, they often overlook implicit document-to-document relations that can arise in real-world settings. We propose a blueprint method for constructing document representations that explicitly accounts for such implicit relations to address this issue.

1 INTRODUCTION

In contemporary NLP, researchers usually model documents as sequential collections of words using techniques such as (recurrent) neural networks Mikolov et al. (2013b;a); Peters et al. (2018)) or attention-based transformer NNs (Devlin et al. (2018); Liu et al. (2019)) to capture contextual relationships between words. Alternative methods for constructing document representations involve building graphs from co-occurring words and using them to represent the documents (Bunke & Riesen (2011); Sonawane & Kulkarni (2014); Yao et al. (2019); Osman & Barukub (2020)). These graph-based document representations have shown promise in capturing not only the sequential relationships between words but also the global structure of the document, performing on par with language model approaches (Zhang et al. (2020); Ragesh et al. (2021); Wang et al. (2023)), allowing for more effective and interpretable representations that can be useful in various downstream NLP tasks (like document similarity (Paul et al. (2016)), topic modeling (Xie et al. (2021)) and document understanding (Gemelli et al. (2023))). Huang et al. (2022) presented a method that enhances the language model's representation by incorporating information from a sub-word co-occurrence graph using a shared loss function. For tasks that involve analyzing documents within a network structure, such as identifying fake news (Han et al. (2020)) or recommending items based on textual reviews Fan et al. (2019), Graph Neural Networks (GNNs) have proven to be highly effective (Wu et al. (2023)). However, real-world documents often lack explicit connections between them, making it difficult to apply GNNs. We present a framework for constructing a document-to-document (D2D) network (available here `https://github.com/bkolosk1/doc2doc.git`) and evaluate various GNNs.

2 GRAPHS OF DOCUMENTS AND WHERE TO FIND THEM?

Let $G = (V, E)$ be a graph, where V is the set of nodes (in our case, the set of documents) and E is the set of edges. In our case the set of edges E is not given and the goal is to construct it with the examination of the potential of documents to form links. The first step of our method is to transform the documents from raw texts to d-dimensional continuous vector space $L \in \mathbb{R}^d$. Next, we calculate the the edge weight as follows:

$$e_{i,j} = sim(L(v_i), L(v_j)) \text{ where } L \text{ denotes the latent document embedding}$$

Here, v_i denotes a document, and $L(v_i)$ the latent embeddings of that document. The function sim measures similarity between two embeddings, representing the edge weight $e_{i,j} \in \{0, 1\}$ between v_i

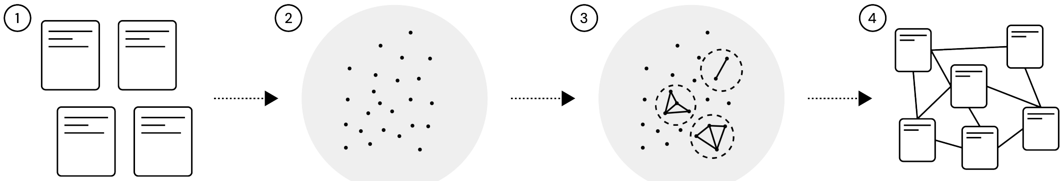

Figure 1: The proposed methodology aims to robustly create a network of related documents.

and v_j. Finally, we apply the thresholding function $thresh$ and keep only the edges that the function discriminates. We examine a transductive and an inductive scenario for obtaining a representation of a new document. The approach is presented in Figure 1 and the algorithm in Appendix 1.

Everything and all at once. In this scenario, we construct a graph between all the documents in the corpus. The graph is built by first computing their latent embeddings and later calculating their similarities. We hypothesize that a given document will have strong connections only with a small subset of the documents. To address this, we relax the constraints and compute the closest *k* documents for each document in *D*. We normalize the documents and construct a KDTree Qiu et al. (2018) of distances, wherethe closest *k* documents are the most similar in terms of cosine distance.

The whole is the same as the sum of its parts. In this scenario, we attempt to induce representations for new, previously unseen documents using the existing graph structure. We begin by constructing the graph using the aforementioned approach only from the training set of documents. Next, we use the NetMF embedding to obtain node (document) representations. To induce a representation for a new document, we first query the KDTree to retrieve the *k* closest neighbors of the paper. We build the final document representation by averaging over the retrieved neighbour node embeddings.

3 Experimental setup and evaluation

To evaluate our method, we choose six different text classification datasets. We benchmark our method against two SOTA language models, LinkBERT (Yasunaga et al. (2022)) and RoBERTa (Liu et al. (2019)), both fine-tuned for 10 $epochs$ with default parameters. We use MPNet as the latent space representation (Reimers & Gurevych (2019)). We explore a grid of hyper-parameters for our method with $thresh \in \{0.1, 0.9\}$, and $k \in \{5, 100\}$. For the transductive models, we use the Spektral implementation of GCN (Kipf & Welling (2016)) and GAT (Veličković et al. (2017))) with default parameters. For NetMF (Qiu et al. (2018)), we use the implementation by Škrlj et al. (2020) and train a LogisticRegression classifier with C=1. Table 1 summarizes the evaluation results (we select the models that performed the best on the dev. set with respect to the F1-Macro.

Table 1: Dataset information and averaged test data evaluation on the 10 runs with different seeds.

	Dataset description				**Language Models**		**Transductive D2D**		**Inductive D2D**
Dataset	**Train Size**	**Dev Size**	**Test Size**	**Classes**	**LinkBERT**	**RoBERTa**	**GAT**	**GCN**	**NetMF3**
BBC (Greene & Cunningham (2006))	790	264	352	4	**0.9863**	0.9815	0.6404	0.7937	0.9460
MBTI (Myers (1962))	4879	1627	2169	16	**0.5695**	0.3468	0.5072	0.1341	0.2116
AAAI-FN (Patwa et al. (2021))	6420	2140	2140	2	0.9802	**0.9819**	0.8229	0.7380	0.9120
PAN-Age (Rangel et al. (2016))	225	76	101	5	0.3881	0.4356	**0.7335**	0.3445	0.4554
PAN-Gender (Rangel et al. (2017))	2024	675	900	4	**0.7523**	0.7444	0.6823	0.5669	0.6078
PAN-FakeNews (Rangel et al. (2020))	270	30	200	2	0.6475	**0.6725**	0.6697	0.5629	0.6450

4 Conclusion and further work

Our method performs comparably with some SOTA techniques, especially with numerous classes and small samples. When more data is available, the method's performance falls compared to the LLMs. To improve our method, we plan to improve the thresholding step, add more layers to the network (either via different embedding methods or via metadata), and speed-up the similarity search with fuzzy-search (e.g. FAISS). We believe the induced graph structure can transform NLP tasks into graph-theoretic ones and vice versa (like clustering with community detection).

ACKNOWLEDGMENTS

We acknowledge financial support from several sources: the Slovenian Research Agency via research core funding for the programme Knowledge Technologies (P2-0103) and the projects CANDAS (Computer-assisted multilingual news discourse analysis with contextual embeddings, J6-2581) and SOVRAG (Hate speech in contemporary conceptualizations of nationalism, racism, gender and migration, J5-3102). A Young Researcher Grant PR-12394 supported the work of the first author. We thank Mihajlo Tunev for the improved scheme illustration.

URM STATEMENT

All authors meet the URM criteria of ICLR 2023 Tiny Papers Track.

REFERENCES

Horst Bunke and Kaspar Riesen. Recent advances in graph-based pattern recognition with applications in document analysis. *Pattern Recognition*, 44(5):1057–1067, 2011.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The World Wide Web Conference*, WWW '19, pp. 417–426, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450366748. doi: 10.1145/3308558.3313488. URL https://doi.org/10.1145/3308558.3313488.

Andrea Gemelli, Sanket Biswas, Enrico Civitelli, Josep Lladós, and Simone Marinai. Doc2graph: A task agnostic document understanding framework based on graph neural networks. In *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*, pp. 329–344. Springer, 2023.

Derek Greene and Pádraig Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd international conference on Machine learning*, pp. 377–384, 2006.

Yi Han, Shanika Karunasekera, and Christopher Leckie. Graph neural networks with continual learning for fake news detection from social media. *arXiv preprint arXiv:2007.03316*, 2020.

Yen-Hao Huang, Yi-Hsin Chen, and Yi-Shin Chen. ConTextING: Granting document-wise contextual embeddings to graph neural networks for inductive text classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 1163–1168, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. URL https://aclanthology.org/2022.coling-1.100.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013b.

Isabel Briggs Myers. The myers-briggs type indicator: Manual (1962). 1962.

Ahmed Hamza Osman and Omar Mohammed Barukub. Graph-based text representation and matching: A review of the state of the art and future challenges. *IEEE Access*, 8:87562–87583, 2020.

Parth Patwa, Shivam Sharma, Srinivas Pykl, Vineeth Guptha, Gitanjali Kumari, Md Shad Akhtar, Asif Ekbal, Amitava Das, and Tanmoy Chakraborty. Fighting an infodemic: Covid-19 fake news dataset. In Tanmoy Chakraborty, Kai Shu, H. Russell Bernard, Huan Liu, and Md Shad Akhtar (eds.), *Combating Online Hostile Posts in Regional Languages during Emergency Situation*, pp. 21–29, Cham, 2021. Springer International Publishing. ISBN 978-3-030-73696-5.

Christian Paul, Achim Rettinger, Aditya Mogadala, Craig A Knoblock, and Pedro Szekely. Efficient graph-based document similarity. In *The Semantic Web. Latest Advances and New Domains: 13th International Conference, ESWC 2016, Heraklion, Crete, Greece, May 29–June 2, 2016, Proceedings 13*, pp. 334–349. Springer International Publishing, 2016.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL `https://aclanthology.org/N18-1202`.

Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, pp. 459–467, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355810. doi: 10.1145/3159652.3159706. URL `https://doi.org/10.1145/3159652.3159706`.

Rahul Ragesh, Sundararajan Sellamanickam, Arun Iyer, Ramakrishna Bairi, and Vijay Lingam. Hetegcn: heterogeneous graph convolutional networks for text classification. In *Proceedings of the 14th ACM international conference on web search and data mining*, pp. 860–868, 2021.

Francisco Rangel, Paolo Rosso, Ben Verhoeven, Walter Daelemans, Martin Potthast, and Benno Stein. Overview of the 4th author profiling task at pan 2016: cross-genre evaluations. In *Working Notes Papers of the CLEF 2016 Evaluation Labs. CEUR Workshop Proceedings/Balog, Krisztian [edit.]; et al.*, pp. 750–784, 2016.

Francisco Rangel, Paolo Rosso, Martin Potthast, and Benno Stein. Overview of the 5th author profiling task at pan 2017: Gender and language variety identification in twitter. *Working notes papers of the CLEF*, 48, 2017.

Francisco Rangel, Anastasia Giachanou, Bilal Hisham Hasan Ghanem, and Paolo Rosso. Overview of the 8th author profiling task at pan 2020: Profiling fake news spreaders on twitter. In *CEUR workshop proceedings*, volume 2696, pp. 1–18. Sun SITE Central Europe, 2020.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Blaž Škrlj, Jan Kralj, and Nada Lavrač. Embedding-based silhouette community detection. *Machine Learning*, 109:2161–2193, 2020.

Sheetal S Sonawane and Parag A Kulkarni. Graph based representation and analysis of text document: A survey of techniques. *International Journal of Computer Applications*, 96(19), 2014.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Yizhao Wang, Chenxi Wang, Jieyu Zhan, Wenjun Ma, and Yuncheng Jiang. Text fcg: Fusing contextual information via graph learning for text classification. *Expert Systems with Applications*, pp. 119658, 2023.

Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, Bo Long, et al. Graph neural networks for natural language processing: A survey. *Foundations and Trends® in Machine Learning*, 16(2):119–328, 2023.

Qianqian Xie, Jimin Huang, Pan Du, Min Peng, and Jian-Yun Nie. Graph topic neural network for document representation. In *Proceedings of the Web Conference 2021*, pp. 3055–3065, 2021.

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

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. Linkbert: Pretraining language models with document links. In *Association for Computational Linguistics (ACL)*, 2022.

Yufeng Zhang, Xueli Yu, Zeyu Cui, Shu Wu, Zhongzhen Wen, and Liang Wang. Every document owns its structure: Inductive text classification via graph neural networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 334–339, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.31. URL https://aclanthology.org/2020.acl-main.31.

APPENDIX 1: ALGORITHM

We present the pseudocode that outlines the steps involved in constructing the D2D network. By following these steps, we can establish a network that captures the relationships between different documents based on a $SimilarityFunction$, from a $LattentEmbedding$ space based on the $ThresholdingFunction$, and a $NearestNeighbourAlgorithm$ for search of the target $kNeighbors$. The algorithmic approach we use is detailed below.

Algorithm 1 D2D network construction algorithm

```
Require: Documents, SimilarityFunction, LatentEmbedding, NearestNeighbourAlgorithm,
  ThresholdingFunction, kNeighbours
  documentNetwork ← Empty
  documentEmbeddings ← LatentEmbedding(Documents)
  nnSearch ← NearestNeighbourAlgorithm(documentEmbeddings)
  for document ∈ Documents do
      TargetDocuments ← nnSearch(document, kNeighbours)
      for targetDocument ∈ TargetDocuments do
          edgeWeight ← SimilarityFunction(document, targetDocument)
          if ThresholdingFunction(edgeWeight) then
              documentNetwork.makeEdge(document, targetDocument, edgeWeight)
          end if
      end for
  end for
  return documentNetwork
```

Time complexity In our approach, first, we use MPNet for *LatentEmbedding*, which infers the embedding of an article in constant time c_{latent}, in total $\mathcal{O}(D \cdot c_{latent}) = \mathcal{O}(D)$. To enable efficient search, we incorporate a KDTree as *NearestNeighbourAlgorithm*, which can be constructed in $\mathcal{O}(D \log D)$ time. The core component of our algorithm is the construction of the adjacency matrix which is based on the number of target neighbours k. The overall complexity of our approach in the worst-case scenario where $k = D$ becomes $\mathcal{O}(D^2)$.