

Comparative Analysis of Community Detection and Transformer-Based Approaches for Topic Clustering of Scientific Papers

Daniel Bretsko[1(✉)], Alexander Belyi[1], and Stanislav Sobolevsky[1,2,3]

[1] Department of Mathematics and Statistics, Faculty of Science, Masaryk University, Kotlarska 2, Brno 61137, Czech Republic
bretsko@math.muni.cz

[2] Center for Urban Science and Progress, New York University, 370 Jay Street, Brooklyn 11201, NY, USA

[3] Institute of Law and Technology, Faculty of Law, Masaryk University, Veveri 70, Brno 61180, Czech Republic

Abstract. We are solving the topic clustering problem, where we need to categorize papers with initially available subjects into more consistent and higher-level topics. We approach the task from two perspectives, one is the traditional network science, where we perform community detection on a subject network with the use of *Combo* algorithm, and the second is the transformer-based *top2vec* algorithm which uses sentence-transformer to embed the content of the papers. The comparison between the two approaches was conducted using a dataset of scientific papers on computer science and mathematics collected from the SCOPUS database, and different coherence scores were used as a measure of performance. The results showed that the community detection *Combo* algorithm was able to achieve a similar coherence score to the transformer-based *top2vec*. The findings suggest that community detection may be a viable alternative for topic clustering when one has predefined topics, especially when a high coherence score and fast processing time are desired. The paper also discusses the potential advantages and limitations of using *Combo* for topic clustering and the potential for future work in this area.

Keywords: Network analysis · NLP · Topic clustering · Community detection · Sentence-transformers

1 Introduction

Topic modeling is often used in natural language processing (NLP) and text mining to group documents or texts that are similar in content. In most cases, topic classification is built exclusively based on the document content. However, in some cases, initial topic attachments may already be provided along with

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023, LNCS 13956, pp. 648–660, 2023.
https://doi.org/10.1007/978-3-031-36805-9_42

the documents. But such initial classification may not be satisfactory for the particular applied purposes due to its granularity, heterogeneity, and, often, lack of clarity on the criteria of how it was defined. At the same time, it may still carry helpful information not to be discarded. Topic clustering can help reduce the number of provided topics and create a higher-level balanced document category representation that can be useful for arranging and grouping the documents or for supervised learning tasks in NLP. The main difference from the more general topic modeling is that in topic clustering, a certain classification attached to the documents is already given and can be used to derive higher-level topics or clusters rather than building them solely based on the content of the documents. It also enables the application of a broader spectrum of techniques, including traditional network science tools.

Different algorithms and approaches are available for topic clustering and topic modeling, each with its strengths and limitations. Some papers describe the approach with classical clustering methods applied to transformer-network embeddings [8,24] or probabilistic and frequency models [21], but without access to external data, they cannot leverage network science approaches that in our case showed great results in terms of coherence. There are also papers employing community detection for topic clustering [5], especially on the basis of social networks [2,16], but they lack the comparison with current state-of-the-art transformer-based models as well as the state-of-the-art *Combo* [19] algorithm for community detection.

To fill these gaps, we compare approaches from network science and modern deep learning applied to the problem of topic clustering. We use two state-of-the-art techniques from each discipline: *Combo* [19] algorithm for community detection in networks, and *top2vec* [4] algorithm, which utilizes *Sentence-BERT* [17] transformer model. We evaluated two methods using a dataset of 10,000 scientific papers related to computer science and mathematics collected from the SCOPUS database. The results are compared using coherence scores [18] as the measure of performance. To ensure the robustness of our results, we use several coherence score algorithms.

2 Topic Modeling

2.1 Problem Setup

The text data for this study was collected from ten thousand *SCOPUS* scientific papers on computer science and mathematics domains. They contain information about the papers' *titles, abstracts*, and *subjects* they belong to. There are three hundred unique subjects, and each paper may belong to several subjects. Our task is to assign each document to one of several subjects, providing a higher-level document classification of desired granularity. To create such a classification based on given data, we employ *Combo* [19] algorithm for community detection in networks, and *top2vec* [4] algorithm, which utilizes *Sentence-BERT* [17] transformer model.

Combo is a state-of-the-art optimization algorithm for community detection in complex networks, which is perfectly suitable for analyzing small to medium size networks, particularly in research tasks that require the quality of resulting partitioning to be as high as possible and was successfully evaluated on partitioning human mobility and interaction networks [6,9,20,23]. The algorithm uses three types of operations to find the best partitions: merging two communities, splitting a community into two, and moving nodes between two distinct communities. After selecting an initial partition made of a single community, the following steps are iterated as long as any gain in terms of the modularity score can be obtained: (1) for each source community, the best possible redistribution of every source nodes into each destination community (either existing or new) is calculated; this also allows for the possibility that the source community entirely merges with the destination; (2) the best merger/split/recombination is performed.

Top2Vec algorithm is a popular and reliable [3,5,7,11] method for performing topic modeling and semantic search, which jointly embeds topic, document, and word vectors in such a way that their proximity represents semantic similarity. Firstly, the *Sentence-BERT* is utilized to create jointly embedded document and word vectors, ensuring that documents are placed close to other similar documents and to the most distinguishing words. Subsequently, the *UMAP* [13] algorithm is applied to reduce the dimensionality of the document vectors, as the high-dimensional document vectors can be sparse. The *HDBSCAN* [12] clustering technique is then employed to find dense areas in the documents and detect anomalies. For each dense area, *top2vec* calculates the topic vector, which is the centroid of the document vectors belonging to the area in the original dimension. Finally, the n-closest word vectors to the topic vector are determined and ordered by proximity, with the closest word vectors serving as the topic words.

2.2 Preprocessing

Text preprocessing was performed with *spaCy* [10] and *Gensim* [15] libraries and regex to clean and prepare the data for both approaches. First, we preprocess the given *subjects* by lowercasing and manual editing. Second, we preprocess the 'title' + 'abstract' concatenation by getting rid of unexpected URLs, extending the contractions, lowercasing the text, and lemmatizing it while deleting the stop words and leaving just nouns(NOUN), proper nouns(PROPN), and adjectives(ADJ), as they are sufficient for text analysis in this case and well-suited as topic cluster keywords. After that, we perform bigram and trigram detection for better keywords and key phrases. Worth noting that developers of *top2vec* point out that it does not require stop-word elimination or lemmatizing for good topic modeling, but during the experiments, we observed that it gave much better coherence score results as well as more human-interpretable keywords and key phrases for cluster description.

2.3 Evaluation Metrics

For the evaluation of our clusters of subjects, we use two different measures of topic coherence [1,18]. The coherence score is a widely used metric for evaluating the quality of topic models [5,24]. It calculates the score of a single topic by measuring the degree of semantic similarity between high-scoring words (keywords) in the topic. These measurements help distinguish between topics that are semantically interpretable and topics that are artifacts of statistical inference. Two types of coherence measures used in our experiments are:

- C_{uci} measure is based on a sliding window and the point-wise mutual information (PMI) of all word pairs of the given top words.
- C_v measure is based on a sliding window, one-set segmentation of the top words, and an indirect confirmation measure that uses normalized point-wise mutual information (NPMI) and the cosine similarity.

While both measures have shown a high correlation with human assessments of topic quality, the correlation of C_v was stronger [18]. As our coherence scores depend on the amount of describing keywords and key phrases for each topic, we will use different amounts of them to describe both algorithms' performance. We have chosen 5, 10, 15, and 20 keywords per topic because fewer amounts do not give a good representation of a topic, and for more than 20 keywords, the scores do not change.

2.4 Modeling

Top2Vec. We use our dataset to obtain embedding for each document in 384-dimensional space (which will also be used to obtain keywords for clusters found by *Combo*) with the use of the *all-MiniLM-L6-v2* sentence transformer model, which is a smaller version of *all-MiniLM* [22]. Next, we create embedding for each unique subject by averaging the embeddings of documents that belong to that subject. After that, we follow the steps described in Sect. 2.1. We reduce the dimensionality of 384-dimensional vectors with the *UMAP* algorithm to perform *HDBSCAN* clustering. We tried several dimensions and found that clustering of 2-dimensional vectors gave the best coherence scores. And we also used them for visualization. Then we find a centroid for each cluster by averaging the embeddings of the subjects that belong to that centroid (in the original 384-dimensional space). We also embed the words in the dictionary to find the closest keywords to the clusters' centroid. We tweak the hyperparameters of HDBSCAN to conduct the comparison of the results for different numbers of clusters. All other hyperparameters were chosen in a way to maximize coherence scores.

Combo. For our community-detection-based topic clustering, we create a graph where subjects correspond to nodes connected by weighted edges, where the weights are the counts of how often papers belong to both topics. On that graph, we run the *Combo* algorithm using *PyCombo* library, which provides the cluster

assignment for each node. After we find the partitions, we follow the same steps as for **Top2Vec** 2.4, but instead of clustering with the *HDBSCAN* algorithm, we use our received communities as a clustering result and then proceed to build the clusters' centroids on their basis. Worth noting that Combo is capable of automatically finding the optimal amount of communities, but for research purposes, we tweaked the *modularity resolution* hyperparameter to receive results with different numbers of found clusters. We ran the algorithm 10 times with different random seeds and selected the result with the highest modularity. All other hyperparameters were chosen in a way to maximize coherence scores.

3 Results

To provide a comprehensive comparison and show the capabilities of both approaches, we conducted the experiments with:

- different numbers of clusters (by tweaking the hyperparameters);
- different numbers of keywords describing the newly-created clusters (for coherence scores pipelines);
- different coherence scores.

To evaluate performance, we measured the execution time of both algorithms and compared coherence scores. After choosing the best number of clusters for each algorithm, we create a 2-D representation of clustering and show wordclouds to facilitate human-interpretable evaluation (qualitative analysis).

3.1 Performance Speed

We measured the performance of both *top2vec* and *Combo* for creating clusters. Since we used preprocessed text and *all-MiniLM-L6-v2* embeddings to analyze both approaches, for the community detection approach, we only measured the performance of *Combo* for the already created subjects-graph, and for *top2vec* only the *UMAP* and *HDBSCAN* steps, where they receive ready-to-use embedding from *all-MiniLM-L6-v2* model. We obtained the following results:

- *Combo*: 334 ms ± 71.9 ms per loop (mean ± std. dev. of 7 runs);
- *UMAP+HDBSCAN*: 1.16 s ± 13.3 ms per loop (mean ± std. dev. of 7 runs).

As we can see, *Combo* is 3 times faster than the *UMAP+HDBSCAN* combination. Worth noting that both approaches have their own preprocessing steps, like creating a graph before applying *Combo* or calculating embeddings for *top2vec*. And those steps are the most time-consuming parts of the pipeline. However, creating a graph takes fractions of a second, while calculating embeddings takes more than 20 min for our dataset.

3.2 Scores Plots and Evaluation

In Fig. 1, we present the results of both algorithms in terms of different coherence scores. Since coherence scores depend on the number of keywords selected for cluster description, we evaluate our results for 5, 10, 15, and 20 keywords. We also adjust the parameters of clustering algorithms to obtain different numbers of clusters.

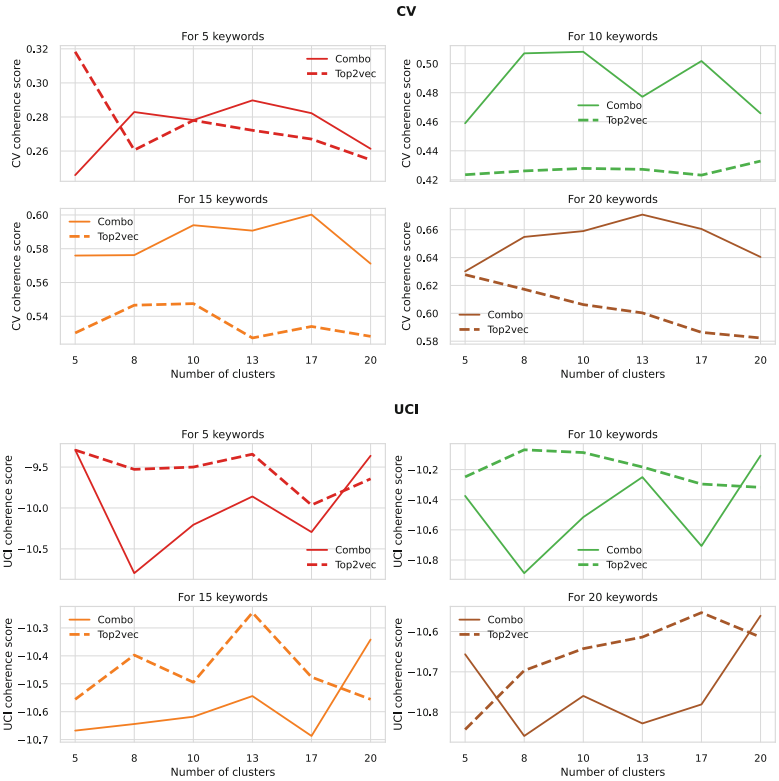

Fig. 1. Coherence scores for a different number of keywords and clusters. *Combo*'s results are represented with solid lines, while *top2vec*'s with dashed lines

It could be seen from Fig. 1 that *Combo* outperforms *top2vec* in terms of C_v coherence (in 96% of the cases), while in terms of C_{UCI} coherence *top2vec* has an advantage over the *Combo* (in 75% of the cases). However, for 20 clusters, Combo demonstrates consistent advantage in all cases. To explore the distribution of scores, in Fig. 2, we show obtained coherence scores in box-plots. As one can see, neither approach shows clear superiority in both topic coherences, so we can conclude that on the quantitative level, they perform similarly.

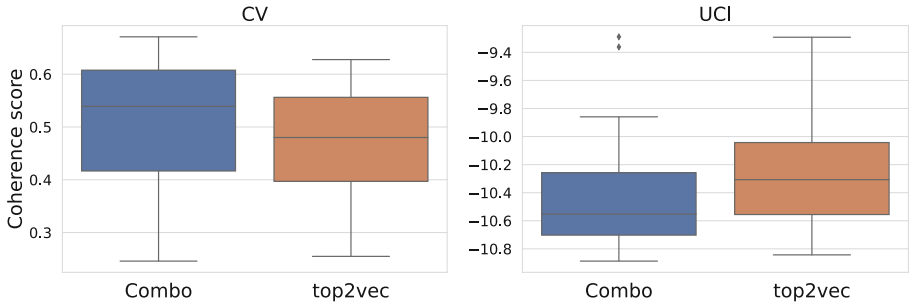

Fig. 2. Box-plots of different coherence scores achieved by *Combo* and *top2vec*

To explore obtained clustering results further for both algorithms, we chose the number of clusters where the results provided the best trade-off between granularity and interpretability based on keywords (see Sect. 3.4) for both algorithms. These numbers are:

- 10 clusters for *top2vec*;
- 9 clusters for *Combo*.

In the following subsections, we will use these numbers of clusters. In Table 1, we show the highest achieved coherence scores for the selected number of clusters.

Table 1. The coherence scores obtained by *Combo* for 9 clusters and *top2vec* for 10 clusters for different numbers of keywords describing the topic (better results are in **bold**)

No Keywords	C_v		C_{uci}	
	Combo	top2vec	Combo	top2vec
5	**0.2782**	0.2780	−10.20	**−9.500**
10	**0.508**	0.427	−10.51	**−10.087**
15	**0.593**	0.547	−10.61	**−10.494**
20	**0.659**	0.606	−10.75	**−10.642**

3.3 2-D Clustering Plot

To visualize the clustering results of both methods and see how different subjects are located relative to each other, we reduce the embedding dimensionality to two dimensions using *UMAP* and show 2-D plots in Figs. 3 and 4. As described in previous sections, for both algorithms, we found the closest keywords for each cluster's centroid. These keywords are used to describe obtained clusters and calculate coherence scores, but we also use the best keyword as the name of each cluster.

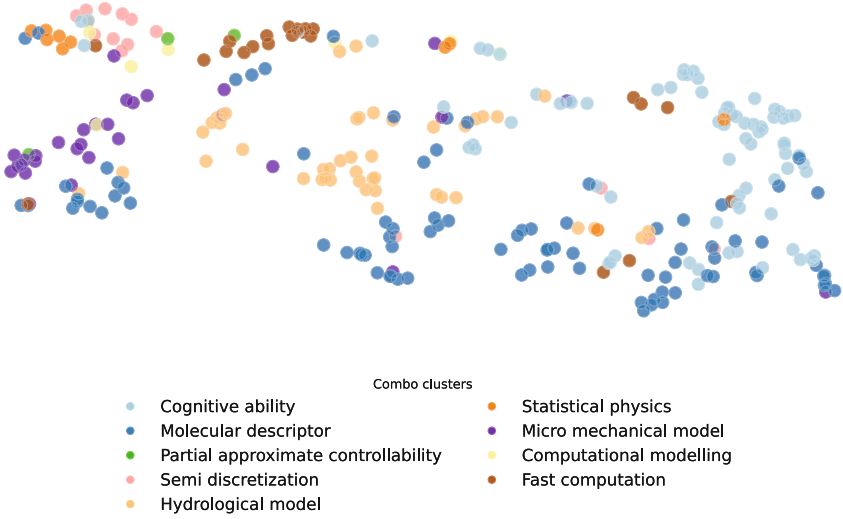

Fig. 3. *Combo* topic clustering result with top-1 keywords as a name

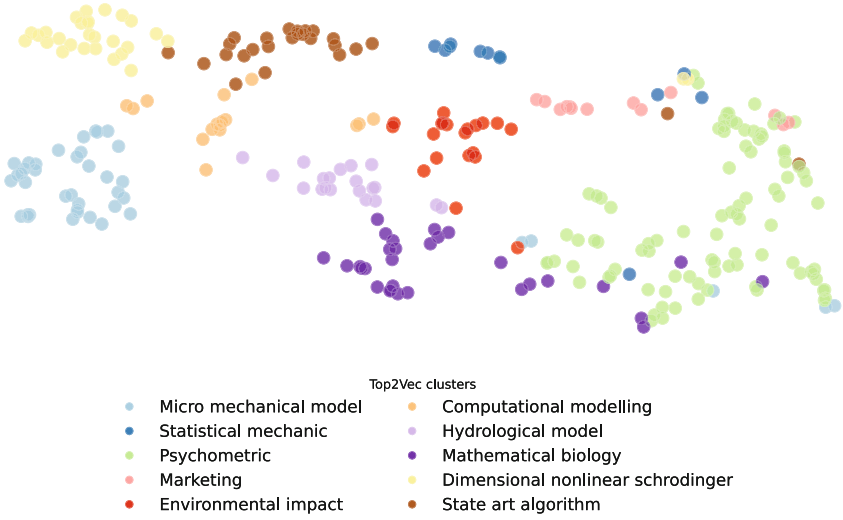

Fig. 4. *top2vec* topic clustering result with top-1 keywords as a name

Since *Combo* uses network topology to find meaningful clusters, while *top2vec* uses subjects' embeddings, 2-D visualization better reflects clusters obtained by *top2vec*.

3.4 Wordclouds

To explore obtained clusters further, we take a deeper look at keywords that describe them. We present these keywords as *wordclouds* [14], which represent our clusters as sets of keywords and key phrases that are gathered within images in such a way that the bigger they are, the shorter their distance to the cluster's centroid, and the better they describe the cluster/topic. With the wordclouds, we can see what keywords and key phrases were used to calculate the coherence scores. Also, they can be used for the qualitative analysis of the results, allowing us to interpret them and see how good the algorithms and their built clusters are. In Fig. 5 and Fig. 6, we present wordclouds of all clusters found by *Combo* and *top2vec* when parameters were adjusted to obtain nine and ten clusters respectively.

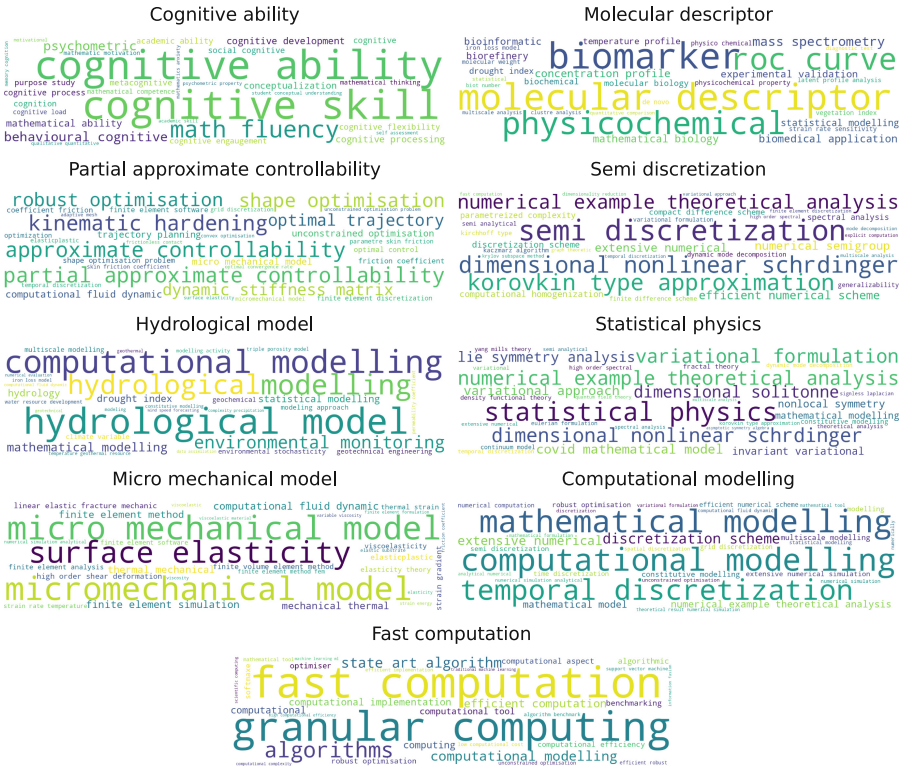

Fig. 5. Wordclouds for clusters obtained by *Combo*

Even though all papers were related to computer science and mathematics, *Combo* clearly distinguished various subtopics related to these fields. We can also observe that even if topics concentrate more on physics or cognitive skills, each of them contains keywords related to computer science and mathematics

like ‘mathematical modeling’, ‘computational modeling’, ‘statistical modeling’, ‘modeling’, ‘computation’, etc., which gives us the sanity-check on created clusters for community detection approach.



Fig. 6. Wordclouds for clusters obtained by *top2vec*

Top2Vec was also able to find interpretable clusters. Topics like ‘Micro mechanical model’, ‘Hydrological model’, and ‘Computational modeling’ were found by both *Combo* and *top2vec*. Topics like ‘Dimensional nonlinear Schrodinger’ and ‘Psychometric’ represent topics similar to ‘Semi discretization’ and ‘Cognitive ability’ found by *Combo* but with different topic names (top-1 keywords). ‘Mathematical biology’ topic is very close to *Combo’s* ‘Molecular descriptor’, and ‘Environmental impact’ and ‘Hydrological model’ look like two separate topics made from *Combo’s* ‘Hydrological model’, while *top2vec’s* ‘Marketing’ and ‘Statistical mechanic’, and *Combo’s* ‘Partial approximate controllability’ and ‘Statistical physics’ are unique for each algorithm.

Based on build wordclouds and their analysis, we conclude that both community detection and transformer-network approaches have found well-interpretable clusters of topics.

4 Discussion

Based on the previous results, we can conclude that for both methods, we were able to achieve comparable coherence scores and receive well-interpretable topics. We can notice that *Combo* performed better in C_v coherence metric and *top2vec* was superior in C_{UCI}, so we concluded that both algorithms perform on the same level in terms of coherence on a given dataset.

The additional advantage of *Combo* is the processing speed. To receive the output for preprocessed 10,000 documents, *top2vec* needs 10–20 min, while *Combo* finds community partitions in seconds. However, to create a graph for our community detection approach, we needed additional information about the documents' topics, where several topics (in our case subjects) are provided for each document. On the contrary, *top2vec* works with the documents' text and does not require any prior information about topics.

Yet, this limitation of the community-detection method might be mitigated when it is possible to create a meaningful graph structure from available information (like a documents co-citation graph). Also, worth noting that by itself *Combo* does not find any keywords, as it gives us the community partitions. So, for additional analysis of received clusters, other tools are needed (as in our approach, *Sentence-BERT* was used to get embeddings for keywords), which is not the case for *top2vec*, as it was designed as a topic modeling algorithm, which does not depend on any predefined topics. On the other hand, *Combo* only requires some graph structure and does not depend on any kind of corpora preprocessing, while *top2vec* needs to get the embedding for the text to use it as an input for *UMAP* and *HDBSCAN*. So, in cases where the aim is to get a higher level representation of already given topics, *Combo* is preferable in the means of processing time and results reliability.

5 Conclusions and Future Work

Based on the findings of this study, it can be concluded that the community detection approach with *Combo* algorithm can be successfully applied to topic modeling through clustering some initial topics provided for the considered documents. The results of such purely network-based topic clustering in terms of processing time could outperform deep topic classification performed from scratch by one of the best algorithms for topic modeling *top2vec* while retaining comparable and often better topic coherence with well-interpretable describing keywords. This suggests that network methods can still be practical tools for topic modeling along with newer state-of-the-art transformer-based techniques, performing deep analysis of the documents' content. It also calls for further research on integrating network and transformed-based approaches for identifying more coherent topics in big text datasets whenever some initial topic classification or network links between documents are provided. For instance, future research could explore the use of *Combo* in topic modeling using word-based or citation-based graphs, where the graph is built based on words or citations. This could

provide insights into how the algorithm performs without predefined topics. One may also consider a question and methodology for finding communities of papers and subjects based on both – their network connections as well as content embeddings, which could inform the development of more effective topic-modeling and topic-clustering algorithms.

Acknowledgement. This research was supported by the MUNI Award in Science and Humanities (MASH Belarus) of the Grant Agency of Masaryk University under the Digital City project (MUNI/J/0008/2021). The work of Stanislav Sobolevsky was also partially supported by ERDF "CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence" (No. CZ.02.1.01/0.0/0.0/16_019/0000822).

References

1. Aletras, N., Stevenson, M.: Evaluating topic coherence using distributional semantics. In: Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)-Long Papers, pp. 13–22 (2013)
2. Amati, G., et al.: Topic modeling by community detection algorithms. In: Proceedings of the 2021 Workshop on Open Challenges in Online Social Networks, pp. 15–20 (2021)
3. Anello, E.: Topic modeling approaches: Top2Vec vs BERTopic, January 2023. www.kdnuggets.com/2023/01/topic-modeling-approaches-top2vec-bertopic.html
4. Angelov, D.: Top2Vec: distributed representations of topics. arXiv preprint arXiv:2008.09470 (2020)
5. Austin, E., Zaïane, O.R., Largeron, C.: Community topic: topic model inference by consecutive word community discovery. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 971–983. International Committee on Computational Linguistics, Gyeongju, Republic of Korea, October 2022. https://www.aclanthology.org/2022.coling-1.81
6. Belyi, A., et al.: Global multi-layer network of human mobility. Int. J. Geogr. Inf. Sci. **31**(7), 1381–1402 (2017)
7. Egger, R., Yu, J.: A topic modeling comparison between LDA, NMF, Top2Vec, and BERTopic to demystify twitter posts. Front. Sociol. **7**, 1–16 (2022)
8. George, L., Sumathy, P.: An integrated clustering and Bert framework for improved topic modeling (2022)
9. Grauwin, S., et al.: Identifying and modeling the structural discontinuities of human interactions. Sci. Rep. **7**(1), 46677 (2017)
10. Honnibal, M., Montani, I.: spaCy 2: natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing (2017)
11. Mavuduru, A.: How to perform topic modeling with Top2Vec, November 2021. https://towardsdatascience.com/how-to-perform-topic-modeling-with-top2vec-1ae9bb4e89dc
12. McInnes, L., Healy, J., Astels, S.: HDBSCAN: hierarchical density based clustering. J. Open Source Softw. **2**(11), 205 (2017)
13. McInnes, L., Healy, J., Melville, J.: UMAP: uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018)
14. Oesper, L., Merico, D., Isserlin, R., Bader, G.D.: WordCloud: a cytoscape plugin to create a visual semantic summary of networks. Source Code Biol. Med. **6**(1), 7 (2011)

15. Rehurek, R., Sojka, P.: Gensim-Python framework for vector space modelling. NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic 3(2) (2011)
16. Reihanian, A., Minaei-Bidgoli, B., Alizadeh, H.: Topic-oriented community detection of rating-based social networks. J. King Saud Univ. Comput. Inf. Sci. **28**(3), 303–310 (2016)
17. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. arXiv preprint arXiv:1908.10084 (2019)
18. Röder, M., Both, A., Hinneburg, A.: Exploring the space of topic coherence measures. In: Proceedings of the 8th ACM International Conference on Web Search and Data Mining, pp. 399–408 (2015)
19. Sobolevsky, S., Campari, R., Belyi, A., Ratti, C.: General optimization technique for high-quality community detection in complex networks. Phys. Rev. E **90**(1), 012811 (2014)
20. Sobolevsky, S., Kats, P., Malinchik, S., Hoffman, M., Kettler, B., Kontokosta, C.: Twitter connections shaping New York city (2018)
21. Tagarelli, A., Karypis, G.: A segment-based approach to clustering multi-topic documents. Knowl. Inf. Syst. **34**, 563–595 (2013)
22. Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., Zhou, M.: MiniLM: deep self-attention distillation for task-agnostic compression of pre-trained transformers (2020). arXiv: 10.48550/ARXIV.2002.10957
23. Xu, Y., Li, J., Belyi, A., Park, S.: Characterizing destination networks through mobility traces of international tourists-a case study using a nationwide mobile positioning dataset. Tour. Manage. **82**, 104195 (2021)
24. Zhang, Z., Fang, M., Chen, L., Namazi-Rad, M.R.: Is neural topic modelling better than clustering? An empirical study on clustering with contextual embeddings for topics. arXiv preprint arXiv:2204.09874 (2022)