The final version of this record is available at: https://doi.org/10.1109/PacificVis56936.2023.00031

Visually Guided Network Reconstruction Using Multiple Embeddings

Daniel Witschard*
Linnaeus University

Ilir Jusufi†
Blekinge Institute of Technology

Kostiantyn Kucher‡
Linköping University

Andreas Kerren§
Linköping University
Linnaeus University

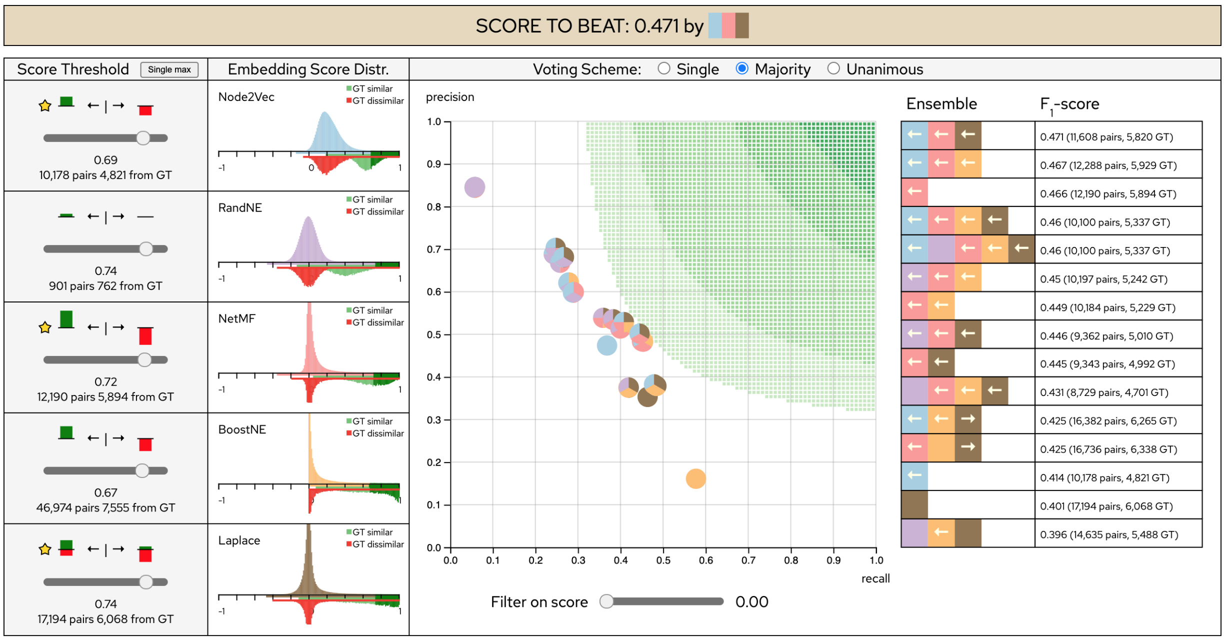

Figure 1: Using the EEVO tool for optimizing multiple-embedding network reconstruction. When searching for the best possible settings, the user can dynamically adjust similarity score thresholds and the choice of voting scheme. The tool displays the current performance of all possible ensembles as well as directive visual guidance to facilitate the search.

Abstract

Embeddings are powerful tools for transforming complex and unstructured data into numeric formats suitable for computational analysis tasks. In this paper, we extend our previous work on using multiple embeddings for text similarity calculations to the field of networks. The embedding ensemble approach improves network reconstruction performance compared to single-embedding strategies. Our visual analytics methodology is successful in handling both text and network data, which demonstrates its generalizability beyond its originally presented scope.

Keywords: Graph embedding, network embedding, similarity calculations, visual analytics, visualization.

Index Terms: Human-centered computing—Visualization—Visualization techniques; Computing methodologies—Machine learning;

1 Introduction

Embeddings are numeric vector representations of underlying data, and they are normally produced in such a way that items which are similar in the original data set (according to some domain-specific aspect) are embedded into vectors that lie close to each other in the embedding space, with regard to some chosen distance metric. The numeric vector format usually makes the embeddings more suitable than the original data as input for computational analysis tasks such as clustering, classification, and similarity calculations. For instance, it is usually more straightforward to calculate a distance measure with numeric vectors than by using the underlying (complex or unstructured) data [2, 11, 14, 15].

In this paper, we extend our previous work on optimizing multiple-embedding similarity calculations for text data [22, 23] to network topology data. Hence we sustain our previous claim that the methodology is generalizable. The main contributions of this paper are:

1. An extension of our previous work showing that the methodology can be generalized to any embeddable data type.
2. A detailed use case explaining how the methodology can be applied to network data.

The manuscript is organized as follows: Section 2 provides a brief overview of relevant related work, Section 3 outlines the computational approach, Section 4 presents the EEVO tool and network reconstruction use case, and Section 5 summarizes the main conclusions of the paper.

2 Related Work

This section contains a short overview of the fields of network embedding and ensemble methods. For a more detailed discussion on related work and the general methodology used in this paper, we refer to our previous work [23].

*e-mail: daniel.witschard@lnu.se
†e-mail: ilir.jusufi@bth.se
‡e-mail: kostiantyn.kucher@liu.se
§e-mail: andreas.kerren{@liu.se, @lnu.se}

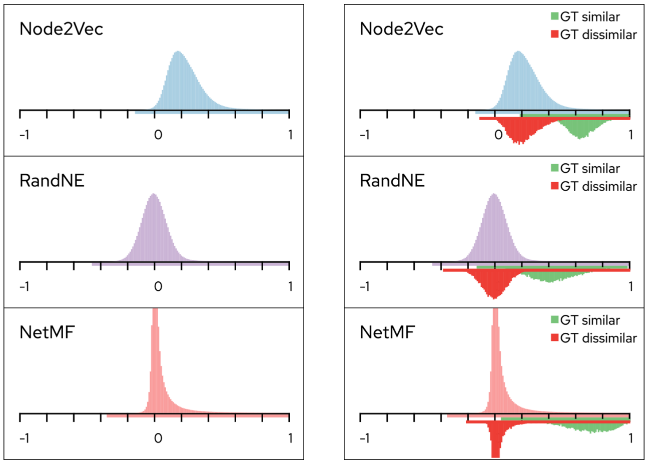

Figure 2: The score distribution plots from three of the node embeddings of the citation network. From the pure distributions (left), we can only observe that the embeddings distribute the scores differently over the pairs, but when we add the score distribution chart for the GT set (right; green/red for pairs with/without citation links, respectively) it is revealed that the different algorithms are not equally successful in separating the two subsets. Furthermore, we note that it is not possible to find a threshold (for any of the algorithms) which yields a perfect split.

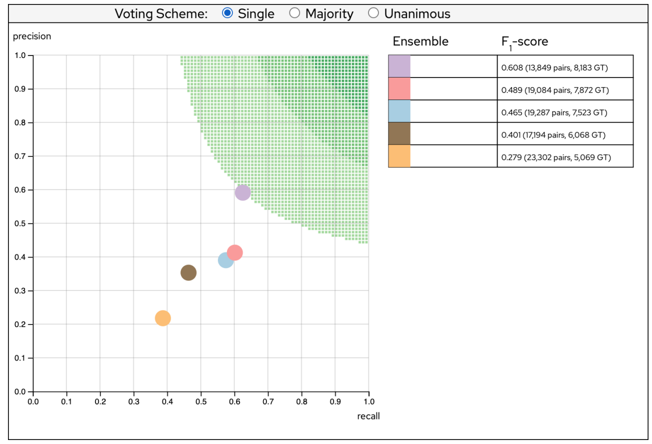

Figure 3: This figure shows the maximum F_1-scores for the individual embeddings. The green bands in the scatterplot indicate the distribution of the F_1-scores (the darker the green, the higher the score). Since the higher scores are situated in the upper right corner, it is easy to see that *RandNE* (represented as a violet dot) achieves the highest score and that *BoostNE* (yellow dot) achieves the lowest. The green coloring is reclined as higher scores are achieved to indicate the current highscore border.

2.1 Graph and Network Embeddings

Embedding calculations are not exclusive to textual data since they can also be applied to various important tasks and applications involving graph and network data. Technology for graph embedding, also known as Representation Learning on Graphs [12], targets the pure topological structure of the graph and ignores any attributed data. The goal is to preserve as much as possible of the structure information and important tasks are clustering, graph comparison, and graph reconstruction. Depending on the application, the item(s) to embed may be: (1) the whole graph, (2) subgraphs, (3) the nodes, or (4) the edges [9, 10]. Furthermore, even dynamic aspects can be taken into account for embedding purposes [18]. The field of network embedding [26] (sometimes also referred to as Attribute Enhanced Representation Learning [6]) is extending the field of graph embedding since not only the graph topology is considered, but also the attributed data.

2.2 Ensemble Methods

Ensemble methods are a well-studied and successful field of classification optimization. The main goal is to find a combination (called an *ensemble*) of several classifiers that provides better results than any of the individual classifiers taken on its own [8, 19]. There are a number of variants of this class of methods, such as bagging [3], boosting [4], and stacking [24].

The existing work on ensemble methods for embeddings mainly focuses on: (1) combining word embeddings from various models and/or text different corpora [17, 21], and (2) enhancing existing word embeddings with specific domain knowledge [25]. The goal is usually to create a new set of word embeddings that combines the strengths of all contributing sets.

3 Computational Approach

We use the same data set [13] as for our previous paper. It contains information of articles published at the IEEE VIS conferences during the period 1990–2018, but this time (instead of creating embeddings from the abstract text of the publications), we create embedding vectors from the *topology of the citation network* which is associated with the publications in our data set. This is done by embedding the nodes of the citation network with five different neighbourhood-based embedding algorithms: Node2Vec [10], RandNE [27], NetMF [20], BoostNE [16], and Laplacian Eigenmaps [1]. The assumption is therefore that the closer two pairs of nodes lie to each other in the citation network, the higher the similarity score will be for the corresponding embedding vectors (for each algorithm). Hence, we are able to use the similarity score as an indicator of how likely it is that a given node pair has a citation link between them. We follow the same computational approach as in our previous work:

1. We calculate the pairwise cosine similarity scores of all node pairs for each algorithm using node embedding vectors. This results in 5 separate similarity scores per node pair, reflecting the similarity between the nodes as seen by each individual algorithm.

2. The user can interactively set threshold scores, one for each algorithm, to classify node pairs into the categories *similar* (scores $\geq$ threshold) and *dissimilar* (scores $<$ threshold). This may result in some node pairs being classified as similar by some embedding algorithms and dissimilar by others, or any combination thereof.

3. We create ensembles by grouping embedding algorithms into all possible combinations. As the order of embeddings is not important, we have a total of 31 ensembles, each consisting of a unique combination of 1 to 5 embedding algorithms (e.g., one ensemble will be Node2Vec/RandNE/NetMF and another will be NetMF/BoostNE, etc.).

4. The user may interactively select a voting scheme to let the participating embeddings "vote" for the combined ensemble classification of each pair so that each algorithm provides its own classification, and the voting scheme resolves any disagreements. If a pair is classified as similar by an ensemble, it is assumed that there is a citation link between the nodes. Thus, for each unique combination of threshold scores and voting scheme, the ensembles will yield 31 unique variants of the network reconstruction.

5. To evaluate the performance of the ensembles, we compare their network reconstruction to a ground truth (GT) set of all the true citation links (approximately 13,000) in the network. The performance metric used is the F_1-score, which is the harmonic mean of *precision* (correctly suggested citation links divided by total suggested citation links) and *recall* (correctly suggested citation links divided by total true citation links).

6. The ensembles are ranked according to their performance, and the user can search for the combination of embeddings that yield the highest F_1-score, i.e., the best possible network reconstruction.

4 Use Case

In this section, we outline a use case of using EEVO for optimizing embedding-based network reconstruction calculations. The mental model of the use case task (i.e., network reconstruction) is the following: (1) since the nodes of the citation network have been embedded with neighborhood-aware embedding algorithms, (2) we will assume that node pairs classified as similar have a direct citation link, and that node pairs classified as dissimilar have no direct link, (3) changing the similarity score thresholds, or the voting scheme, will therefore yield a different reconstruction of the underlying citation network (since the set of pairs classified as similar will change), and (4) since we know the true network topology we can continuously compare to the GT set and eventually select the settings which yielded the most correct reconstruction.

The EEVO tool is a web-based tool implemented using D3 [7] and designed to help the user find the best-performing ensemble of embeddings. The tool provides continuous and interactive visual guidance [5] to simplify the search process. The key components of the EEVO interface are:

The Embedding View Where the embedding score thresholds can be set, and the corresponding classification statistics can be assessed (see Figure 1, left side).

The Ensemble Performance View Where the voting scheme can be selected, and the ensemble performance is displayed in a scatterplot and in a high-score table (see Figure 1, right side).

Each embedding type (i.e., each embedding algorithm) is color-coded with a unique color and the ensembles are represented by circular multi-colored glyphs in the scatterplot and by multi-colored rectangles in the high-score table. For instance, an ensemble made from the combination of the blue, purple and pink embeddings will have these three colors on its glyph and its rectangle. In the scatterplot, the glyphs are plotted with regards to the current precision (y-axis) and recall (x-axis), which means that the higher F1-scores are to be found in the upper right corner. As previously discussed, we load five different embedding types (see Section 3) into EEVO. We will therefore have a total of $2^5 - 1 = 31$ possible ensemble combinations to evaluate (if counting single embeddings as ensembles as well), and the tool continuously evaluates the performance of all of them. The user may choose between three different voting schemes: (1) *Single* – a pair is classified as similar if at least one of the embeddings in the ensemble has classified it as similar, (2) *Majority* – a pair is classified as similar if more than half of the embeddings in the ensemble have classified it as similar, and (3) *Unanimous* – all embeddings must classify the pair as similar.

Assess embedding interdependency. The analyst starts by assessing the *Embedding Score Distribution* column of the visualization (see Figure 1 left side and Figure 2). The pure score distributions only reveal general information (see Figure 2, left), but when the pairs of the GT set are highlighted more interesting conclusions can be drawn (see Figure 2, right). The analyst can conclude that there is a general, and encouraging, tendency for all the embeddings to assign higher scores to the GT pairs with citation links than to the ones without. Furthermore, he anticipates that variation between the different embeddings could be possible to exploit for a well-chosen ensemble configuration.

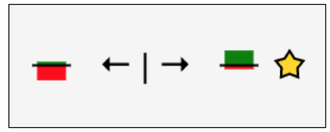

Figure 4: The graphical guidance support above the score thresholds sliders is intended to show the effect of moving the threshold setting one step to the left/right. The number of affected ensembles is encoded by the bar height, and the potential effect is encoded by direction and color: green and upwards for higher scores, red and downwards for lower scores. As can be seen from this example, there is currently no move that would be beneficial for all ensembles. The star indicates that a new high-score can be obtained if the slider is moved in this direction.

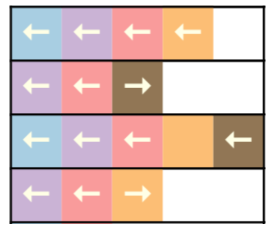

Figure 5: The guidance support in the high-score table is intended to show which way a slider should be moved to benefit a specific ensemble. Color encodes the embedding identity, and the arrow direction encodes direction (i.e., left means lowering the threshold and right means raising the threshold). As we can see from this example, the different ensembles have different "opinions" on in which way the sliders should be moved for both brown and orange embedding.

Assess one-by-one performance. To obtain a benchmark score for the ensemble calculations the analyst first searches for the best threshold setting for each individual embedding. By adjusting one slider at a time (leaving the others at their initial values), he/she can easily verify that the highest scoring single-embedding ensemble is RandNE and that the worst performing one is BoostNE (see Figure 3) By looking at the statistics cell of the high-score table, the analyst can conclude that the optimal threshold score for RandNE corresponds to a citation network reconstruction with a total of almost 14,000 edges and that just above 8,000 of those are correct. Furthermore, he/she can see that this reconstruction gives a F_1-score of 0.608, which in other words is the best result that we can achieve when using only one single embedding.

Assess ensemble performance. The analyst now focuses on trying to find settings which yield a higher F_1-score than the benchmark found in the previous step. As previously mentioned, to facilitate the search EEVO continuously calculates and displays directive visual guidance which allows the analyst to assess the consequences of decrementing or incrementing each score threshold by 0.01 Aggregated guidance information is displayed as "information scent" above each slider (see Figure 4), while ensemble-specific guidance is displayed in the high-score table (see Figure 5). The intention of the design is to augment the chances of finding optimal performing ensembles by giving the analyst the possibility to combine his/her own logical reasoning with the guidance.

The first choices to consider are which initial settings to use on the sliders and what voting scheme to use. For both, the main options

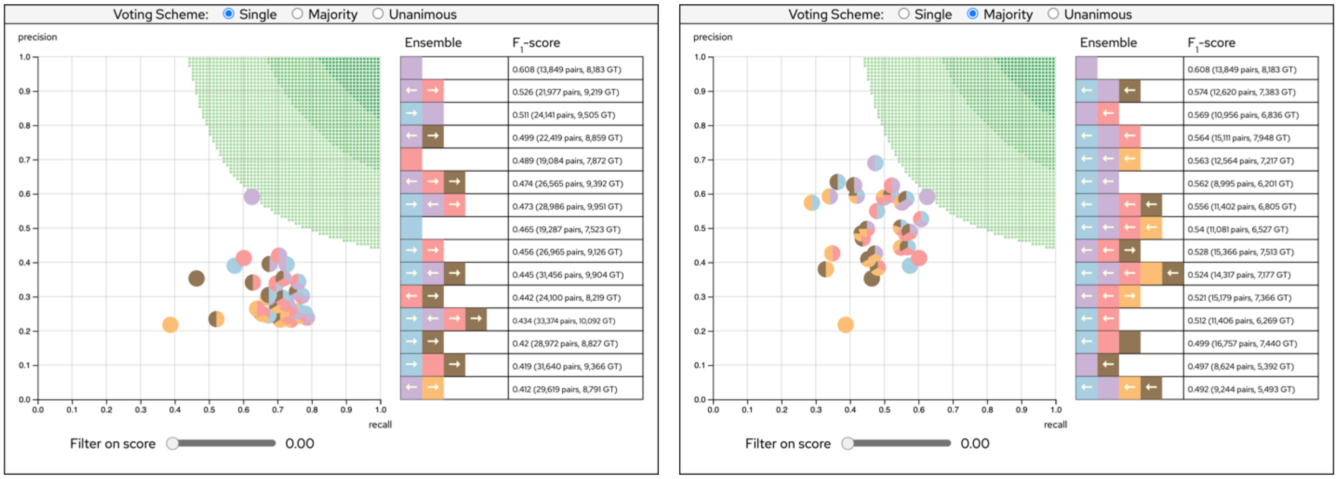

Figure 6: By comparing the visual impressions of using the voting schemes Single (left) and Majority (right), the analyst concludes that Majority seems to hold a greater potential for success for the current slider positions. This decision (which might be wrong) is based on the fact that more of the ensembles appear to be positioned within "striking distance" of the benchmark score, and that there seems to be consistent guidance for several of the ensembles.

are: (1) a random setting, or (2) an educated guess. The analyst chooses to set the slider positions to the values obtained from the previous step to see if this could make for a suitable starting point (i.e., each slider is positioned at the value which gives the highest F_1-score for the corresponding embedding). By switching between the different voting schemes and comparing the visual impression of the scatterplots and the high-score tables, the analyst then concludes that the voting scheme Majority seems to hold the most potential (see Figure 6). By applying logical reasoning when following the main directions of the guidance, the analyst can now easily find an ensemble with a score of 0.621 (namely, RandNE + Node2Vec), which corresponds to an improvement of +2% (which in turn could roughly be translated to about 250 more correctly reconstructed links in the network). To better understand the current results, the analyst can, at any point, click on a statistics cell in the highscore table to display the *Similarity Assessment View* (see Figure 7). In this view, the true distance between the nodes can be assessed, and a comparison of the abstract texts can be made. To facilitate the assessment, common words are highlighted and article pairs which have been correctly classified as having a citation link have green background color.

However, the settings found above do not necessarily yield the highest score that can be achieved. So, the analyst must now continue the search by trying other starting positions and/or different search strategies. By using the tool in this way, we have been able to find settings (for an ensemble consisting of three embeddings) which achieve an improvement of +3% (compared to the best single embedding), although this may be only a local optimum.

5 Discussion and Conclusions

In this paper, we have extended our previous work on using multiple embeddings for similarity calculations, which mainly focused on textual data. We provide a complementary use case for network reconstruction to demonstrate that the same methods (and the same VA tool) can be used for such fundamentally different data types as textual data and network data. By this, we show that our methodology is generalizable well beyond its previously presented scope. The main reason for this is the fact that our methods do not make any assumptions on the underlying data type, but rather treat the embeddings as generic vectors. This implies that they can be used for any embeddable data type, provided that it can be embedded in several different ways (which is usually possible by using different algorithms or by feeding the same algorithm different portions of the data). In turn, this means that the concept of combining several different embeddings—in order to obtain higher quality—could be used for many scenarios. We therefore claim that our methodology generalizes to numerous different data types, and our hope is that our contribution could be beneficial for many application areas.

Furthermore, we see the opportunity of combining the results of the previously published text embedding use case and the current network reconstruction use case in such a way that (1) the results from the text similarity calculations could be used to enhance the citation network reconstruction (e.g., by assuming that similar articles should have a citation link), or (2) the other way around (i.e., using the citation network proximity as a parameter for determining the text similarity of the publications). Here, we want to point out that combining text similarity calculations and citation analysis is not a novelty per se but, to the best of our knowledge, doing so by using multiple embeddings is a novel approach. More specifically, one of the main benefits of the embedding approach is that it allows for a homogeneous computational framework that do not depend on the underlying data types (as compared to using, and combining, data-type-dependent calculations). Furthermore, for textual data and network data, embedding methods already achieve state-of-the art results for similarity calculations, so methods for leveraging these technologies could be of broad interest.

Acknowledgements

This work was partially supported through the ELLIIT environment for strategic research in Sweden.

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, p. 585–591. MIT Press, Cambridge, MA, USA, 2001.

[2] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, Aug. 2013. doi: 10.1109/TPAMI.2013.50

[3] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

Figure 7: In the *Similarity Assessment View*, the analyst can assess the article pairs which are currently classified as having a direct citation link. Correctly classified pairs have green background color, and to facilitate the comparison of the abstract texts common words are highlighted in yellow.

[4] L. Breiman. Arcing classifier (with discussion and a rejoinder by the author). *Annals of Statistics*, 26:801–849, 1998.

[5] D. Ceneda, N. Andrienko, G. Andrienko, T. Gschwandtner, S. Miksch, N. Piccolotto, T. Schreck, M. Streit, J. Suschnigg, and C. Tominski. Guide me in analysis: A framework for guidance designers. *Computer Graphics Forum*, 39(6):269–288, Sept. 2020. doi: 10.1111/cgf.14017

[6] P. Cui, X. Wang, J. Pei, and W. Zhu. A survey on network embedding. *IEEE Transactions on Knowledge and Data Engineering*, 31(5):833–852, May 2019. doi: 10.1109/TKDE.2018.2849727

[7] D3 — Data-driven documents. https://d3js.org/, 2011. Accessed February 3, 2023.

[8] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma. A survey on ensemble learning. *Frontiers of Computer Science*, 14(2):241–258, 2020. doi: 10.1007/s11704-019-8208-z

[9] P. Goyal and E. Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, July 2018. doi: 10.1016/j.knosys.2018.03.022

[10] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 855–864. ACM, 2016. doi: 10.1145/2939672.2939754

[11] L. Gutiérrez-Gómez and J.-C. Delvenne. Unsupervised network embeddings with node identity awareness. *Applied Network Science*, 4(1):82, Oct. 2019. doi: 10.1007/s41109-019-0197-1

[12] W. L. Hamilton, R. Ying, and J. Leskovec. Representation learning on graphs: Methods and applications. *IEEE Data Engineering Bulletin*, 40(3):52–74, Sept. 2017.

[13] P. Isenberg, F. Heimerl, S. Koch, T. Isenberg, P. Xu, C. D. Stolper, M. Sedlmair, J. Chen, T. Möller, and J. Stasko. Vispubdata.org: A metadata collection about IEEE Visualization (VIS) publications. *IEEE Transactions on Visualization and Computer Graphics*, 23(9):2199–2206, Sept. 2017. doi: 10.1109/TVCG.2016.2615308

[14] A. Khan, Q. Shah, M. I. Uddin, F. Ullah, A. Alharbi, H. Alyami, and M. A. Gul. Sentence embedding based semantic clustering approach for discussion thread summarization. *Complexity*, 2020:4750871, Aug. 2020. doi: 10.1155/2020/4750871

[15] J. Kim, J. Yoon, E. Park, and S. Choi. Patent document clustering with deep embeddings. *Scientometrics*, 123(2):563–577, May 2020. doi: 10.1007/s11192-020-03396-7

[16] J. Li, L. Wu, and H. Liu. Multi-level network embedding with boosted low-rank matrix approximation. *CoRR*, Aug. 2018. doi: 10.48550/arXiv.1808.08627

[17] A. Muromägi, K. Sirts, and S. Laur. Linear ensembles of word embedding models. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, NoDaLiDa '17, pp. 96–104. ACL, May 2017.

[18] G. H. Nguyen, J. Boaz Lee, R. A. Rossi, N. K. Ahmed, E. Koh, and S. Kim. Dynamic network embeddings: From random walks to temporal random walks. In *Proceedings of the IEEE International Conference on Big Data*, BigData '18, pp. 1085–1092. IEEE, 2018. doi: 10.1109/BigData.2018.8622109

[19] D. Opitz and R. Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, Aug. 1999.

[20] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang. Network embedding as matrix factorization. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, feb 2018. doi: 10.1145/3159652.3159706

[21] R. Speer and J. Chin. An ensemble method to produce high-quality word embeddings. *CoRR*, Apr. 2016. doi: 10.48550/arXiv.1604.01692

[22] D. Witschard, I. Jusufi, R. M. Martins, and A. Kerren. A statement report on the use of multiple embeddings for visual analytics of multivariate networks. In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP '21) — Volume 3: IVAPP*, IVAPP '21, pp. 219–223. INSTICC, SciTePress, 2021. doi: 10.5220/0010314602190223

[23] D. Witschard, I. Jusufi, R. M. Martins, K. Kucher, and A. Kerren. Interactive optimization of embedding-based text similarity calculations. *Information Visualization*, 21(4):335–353, 2022. doi: 10.1177/14738716221114372

[24] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992. doi: 10.1016/S0893-6080(05)80023-1

[25] C. Xia, T. He, J. Wan, and H. Wang. Ensemble methods for word embedding model based on judicial text. In *Web Information Systems and Applications*, vol. 11817 of *LNCS*, pp. 309–318. Springer, 2019. doi: 10.1007/978-3-030-30952-7_31

[26] D. Zhang, J. Yin, X. Zhu, and C. Zhang. Network representation learning: A survey. *IEEE Transactions on Big Data*, 6(1):3–28, Mar. 2020. doi: 10.1109/TBDATA.2018.2850013

[27] Z. Zhang, P. Cui, H. Li, X. Wang, and W. Zhu. Billion-scale network embedding with iterative random projection, 2018. doi: 10.48550/arXiv.1805.02396