Lnu Licentiate No. 37, 2022

Towards Multiple Embeddings for Multivariate Network Analysis

Daniel Witschard

LINNAEUS UNIVERSITY PRESS

Daniel Witschard

Towards Multiple Embeddings for Multivariate Network Analysis

PhL Dissertation

Computer Science

2022

Linnæus University

This page will be replaced by the publisher (for the books in LUD Series)

Abstract

The study of multivariate networks (MVNs, i.e., large data sets where data points have relations to other data points and both these relations and the points themselves can have attributed data) is an important task in many different fields, such as social networks for the humanities, citation networks for bibliometrics and biochemical networks for life sciences. Furthermore, when dealing with visualization and analysis of MVNs, many open challenges still exist regarding both computational aspects (i.e., the challenge of computing different metrics of a large-scale MVN) and visual aspects (i.e. the challenge of displaying all the information of a large-scale MVN in a way that is comprehensible to the user). In the search for efficient and scalable visual analytics methods, especially for exploratory data analysis, this thesis explores a novel approach of aspect-driven MVN embedding and the use of ensembles of embeddings for multi-level similarity calculations. Starting from the observation that there already exist several different embedding techniques for datatypes that are common for real-world MVNs, the main question that we will try to answer is: *"Could the use of multiple embeddings provide for new and better solutions for visual analytics on multivariate networks?"* This main question then inspires the formulation of four more specific research goals regarding: (1) methods for combining embeddings, (2) the development of a general methodology framework, (3) new visualization methods, and (4) proof-of-concept applications for real-world scenarios.

The focus of our work lies on similarity-based analysis within the domains of bibliometrics and scientometrics, and our first major step is to develop a methodology for combining several different embeddings (for the same underlying data) to augment the quality of similarity calculations. This step includes an adaptation of some of the key ideas from ensemble methods to the field of embeddings, and also an interactive optimization process for finding the best performing ensembles. Upon this foundation, we develop an *aspect-driven* approach which seeks to divide an underlying MVN into separately embeddable aspects, which in turn allows for the resulting embedding vectors to be used in flexible analysis scenarios with high level of interaction. We then proceed to show how the concept of similarity-based analysis can be used to obtain valuable insights to, and a better understanding of, a large set of scientific publications. For this, we introduce the abstract concept of *similarity patterns* which we use to express how a specific set of similarity criteria are distributed over a data set. Furthermore, we present proof-of-concept applications which are designed to allow the user to exploit these similarity patterns at different levels of detail. We also show that our proposed methodology is generalizable beyond the scope of MVNs, and therefore could be applied to other fields as well.

Keywords: Multivariate networks, embeddings, ensemble methods, similarity calculations, visual analytics

Acknowledgments

Writing a licentiate thesis is not an easy task, and when starting you do not know where the process will take you. By a stroke of luck, I have been fortunate enough to share my journey with the talented members of the ISOVIS research group, and being part of such an ambitious and successful group has been a real privilege. First and foremost, I would like to thank my supervisors, Dr. Ilir Jusufi, Dr. Rafael Martins and Prof. Dr. Andreas Kerren for your continuous support, guidance and encouragement. Thank you for believing in me and setting high standards. I would also like to thank Dr. Kostiantyn Kucher and my PhD colleague Angelos Chatzimparmpas for helping me with countless practical questions, and thereby easing my administrative stress. Finally, I am very grateful towards my examiner, Prof. Dr. Welf Löwe, who saw an opportunity where so many others saw an impediment—otherwise my journey as a scientist would never even have started.

Table of Contents

List of Figures

List of Publications

This licentiate thesis report is based on the following refereed publications and submitted manuscripts (I have contributed to all stages of work as the lead author):

1. Witschard, D., Jusufi, I., Martins, R.M., Kerren, A. Paper title: 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 2021)*, Volume 3: IVAPP. 219-223.

2. Witschard, D., Jusufi, I., Martins, R.M., Kucher, K., Kerren, A. Paper title: Interactive Optimization of Embedding-based Text Similarity Calculations. Submitted to *Information Visualization Journal (SAGE Journals) 2022*.

3. Witschard, D., Jusufi, I., Kerren, A. Paper title: SimBaTex : Similarity-based Text Exploration. In *Posters of the 23rd EG/VGTC Conference on Visualization (EuroVis '21)*, 5-7.

4. Witschard, D., Jusufi, I., Kerren, A. Paper title: Simbanex : Similarity-based Network Exploration. Planned submission to *Graph Drawing Conference 2022*.

Other refereed publications and submitted manuscripts that are not directly part of this thesis:

1. Witschard, D., Jusufi, I., Kerren, A. Paper title: Dynamic Ranking of IEEE VIS Author Importance. Poster publication at *IEEE VIS Conference 2021*

2. Thygesen, S., Witschard, D., Bin Masood, T., Kerren, A., Hotz, I. Paper title: Comparing Chemical Feature Vector Representations Using Visual Analytics. Submitted to *EuroVA 2022*

Chapter 1

Introduction

Contents

In this chapter we will present the main scope, the main goals and the main limitations of our work, and the first section is based on the content of our position paper [100]. The study of multivariate networks (MVNs, i.e., large data sets where data points have relations to other data points and both these relations and the points themselves can have attributed data) is an important task in many different fields, such as social networks for the humanities, citation networks for bibliometrics and biochemical networks for life sciences [47,53]. Therefore, methods for computational analysis and visualization of MVNs have become important research fields and have attracted a lot of attention. One of the many challenges with visual analytics (VA) on MVNs is that the attributes and the network structure are equally important for obtaining a correct understanding of the underlying data. For instance, to better understand the interactions within a social network we need to consider both the topological structure of the network (i.e., how "close" different actors are to each other) and the specific values of attributes such as age, gender, interests etc. For example, if we would like to predict future friendship relations in a social network, we most probably need to consider both actor proximity/distance as well as actor similarity/dissimilarity to obtain a good result.

As we can see from the example in Figure 1.1, for MVN analysis it is often not enough to use only traditional statistical methods, or only pure graph analysis, since their scope is too limited. Instead we need more integrated VA methods that are able to exploit the attributed data in the direct context of the topological

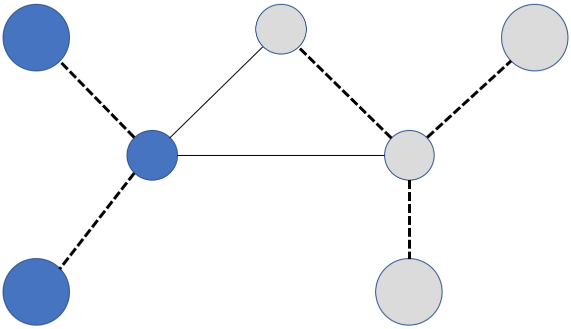


Figure 1.1: *When analyzing an MVN, we need to consider both the network structure and the attributed data for nodes and edges. In this example, there is one attribute for the edges (with two possible values) and two different attributes for the nodes (with two possible values each). We visualize the data set as a node-link diagram and encode the edge attribute to the edge line style and the two node attributes to node color and size. We can now see that a possible pattern emerges: the links between nodes of the same color differ from the links between nodes of different color, and nodes with links to nodes of different color are smaller. This pattern may not have been discovered if the attribute data and the network topology had been visually analyzed separately.*

aspects. One major step forward in tackling this challenge has been the recent development of extending graph-specific embedding technologies to the field of MVNs. 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 [8, 35, 70, 95, 104]. 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. A recent trend within the field of MVN embedding has been the development of methods for so-called attribute-enhanced representation learning which aims to jointly embed the topology of the network together with the attributed data. While this strategy has proven to be successful for some scenarios, it also entails some limitations in flexibility since it aims to join several different aspects of the underlying MVN into a single, inseparable, embedding (e.g., jointly embedding node text content and node position in the network topology). To achieve higher flexibility, it would instead be advantageous with a strategy that allows for using and combining several different embeddings of the underlying data. In this thesis we present such an approach and use it to build a framework for similarity-based analysis of MVNs. The strength of the proposed framework is showcased by proof-of-concept visualizations targeting several different real-world tasks. Although our examples are mainly from the fields of bibliometrics and scientometrics [29,62,69], we would like to point out that our proposed methodology operates on the vector level (i.e., it makes no assumptions on the nature of the embedded data) and therefore is generalizable also to other fields.

1.1 Motivation

As we have previously stated, MVNs are important data sets within many different fields and their analysis continues to be a challenge. Even though important steps forward have been taken, there are still many open research questions within the field of MVN embedding, and we are still far from any generic, comprehensive methodology. Therefore, new paths need to be explored and alternative ways forward need to be identified. Open challenges exist on the computational side as well as on the visualization side. For the computations, a great challenge lies in the fact that algorithms for network topology analysis often scale poorly to large data sets because of combinatorial explosion of the number of network paths between nodes. For the visualization, a great challenge lies in the fact that many MVNs are so big and complex that only a few separate characteristic, out of many possible, can be displayed in parallel. Thus, helping the analyst to maintain provenance and build a correct mental model requires solutions that go beyond standard visual representations.

When assessing the current research frontier, we see an opportunity to complement the resent efforts focused on developing MVN-specific embeddings. We therefore propose a more generic approach which seeks to leverage already existing embedding technology, and apply it in the context of MVNs. By doing so, we aim to fill a research gap which, to the best of our knowledge, has not yet been explored.

1.2 Research Goals

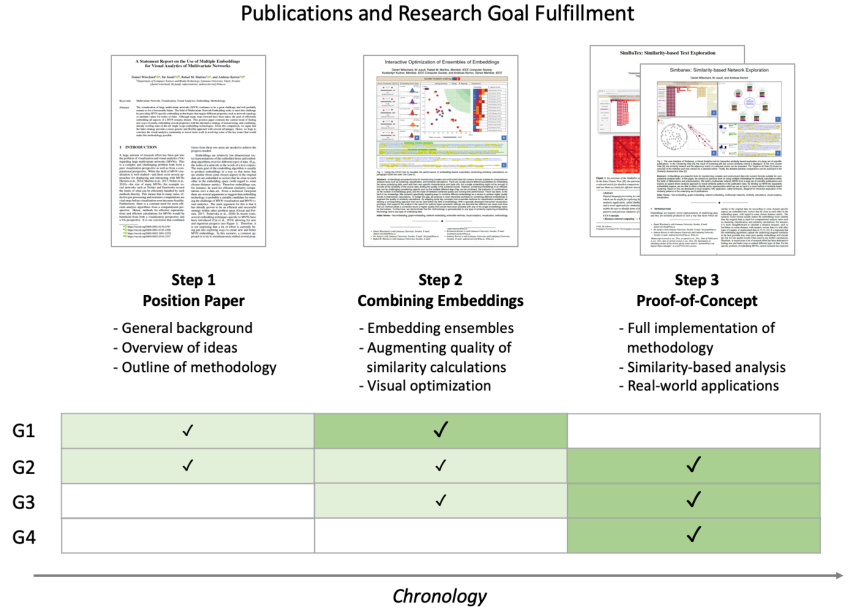
The aim of this thesis is to explore the use of multiple embeddings for analysis and visualization of MVNs, and our focus lies on ways to reuse and combine already existing embedding technologies. Furthermore, since embeddings can be used for several different computational tasks (each with its own special characteristics), we have chosen to specifically focus on similarity calculations and use a similarity-based approach for our proposed methodology. In other words, we first intend to use the embeddings vectors from the MVN to calculate pairwise similarities, and then to use the result from these calculations as the basis for the visualization and the exploration of the MVN. With this in mind, we formulate our main research goals as the following:

- *G1:* Find a method for how to combine several embeddings in order to augment the quality in similarity calculations.
- *G2:* Develop a general framework for MVN embedding which reuses already existing embedding technologies.
- *G3:* Develop new visualization methods which use the results of similarity calculations to reveal interesting characteristics of the underlying MVN.
- *G4:* Demonstrate how similarity calculations based on multiple embeddings can be of value for visual analytics on MVNs within the domains of bibliometrics and scientometrics.

1.3 Research Approach

Since this thesis deals with the elusive concept of similarity (which very much lies in the eyes of the beholder), we would like to start by pointing out that, as a general strategy, we will favor human-in-the-loop solutions and the use of interactive visualization. This bias is based on our firm belief that using visual analytics applications enables the user obtain important insights (leading to higher trust) as compared to using purely numerical methods. Furthermore, similarity-based aspects are challenging to capture with purely computational approaches, so we are convinced that they are better handled by applications which also make use of the tremendous pattern-recognition capabilities of the human brain.

In order to fulfill our research goals, there are several different steps needed. The first step is a literature review of potentially related work, and the result is mainly presented in the Related Work section; but it has, of course, also indirectly influenced other parts of our work (e.g., the design of our proposed visualizations or the choice of embedding algorithms that we use) and therefore supports all of our research goals. The second step, which mainly targets G1 and G2, is outlining and developing a methodology framework which is suitable to use for real-world scenarios (within the domains of bibliometrics and scientometrics) and which is anchored on a firm theoretical base. For this, we published/submitted two articles: (1) a statement report outlining our general ideas for using several embeddings and the potential this could hold for MVN analysis [100] (the first paper in the List of Publications), and (2) a paper covering our proposed methodology for combining several embeddings together with a proof-of-concept tool for visual optimization of the search for high-performing ensemble of embeddings (the second paper in the List of Publications). The third and final step, which mainly targets G3 and G4, is to provide functional proof-of-concept applications which showcase the use and the potential of our work in the context of MVN analysis. This step was covered by the publication of a poster paper [99] (the third paper


Publications and Research Goal Fulfillment
A Statement Report on the Use of Multiple Embeddings for Visual Analytics of Multivariate Networks
Interactive Optimization of Ensembles of Embeddings
SimBaTex: Similarity-based Text Exploration
Simbanex: Similarity-based Network Exploration
Chronology

	Step 1 Position Paper	Step 2 Combining Embeddings	Step 3 Proof-of-Concept
	- General background - Overview of ideas - Outline of methodology	- Embedding ensembles - Augmenting quality of similarity calculations - Visual optimization	- Full implementation of methodology - Similarity-based analysis - Real-world applications
G1	✓	✓	
G2	✓	✓	✓
G3		✓	✓
G4			✓

Figure 1.2: *A schematic overview of the different steps of the research process and how the corresponding publications are connected to the fulfilment of the research goals. The body of this thesis has been developed through a series of papers which build upon each other and have the proposed methodology as a common narrative thread.*

in the List of Publications) and by submitting a full paper (the fourth paper in the List of Publications), both containing the descriptions of the implementations of prototype VA tools which showcase our methodology on real-world scenarios from the fields of bibliometrics and scientometrics. An outline of the different steps, the corresponding publications and their connections to the research goals is given in Figure 1.2.

As can be seen, the steps and the research goals mentioned in this section form a narrative thread together, both logically and chronologically. To facilitate the presentation and the understanding of our work contribution we will use this thread as a base for the outline of this thesis.

1.4 Main Contributions

The main contributions of this work are:

1. A novel way to apply ensemble methods to embeddings which can be used as a means to improve the quality of embedding-based similarity calculations.

2. A VA tool, called EEVO, that guides the search for better hyperparameter settings and allows the analyst to build a mental model of the inner workings of the ensemble calculations.
3. A general methodology for dividing a MVN into separately embeddable aspects and strategies for combining the resulting embeddings.
4. A VA tool, called Simbanex, which allows the user to explore a large set of scientific documents by interactive specification of similarity criteria and assessment of the corresponding similarity patterns and similarity details.
5. The presentation of different use cases which illustrate the strengths of our proposed approach and showcase the usage of our tools.

Furthermore, as we make no assumptions on the data type of the embedded items, the generalizability of our method is promising since it can be used for any similarity-capturing embedding technology. This is demonstrated in Section 3.4 by two example use cases with fundamentally different tasks and data types. We therefore hope that our contribution could prove to be useful and valuable for a wide range of applications.

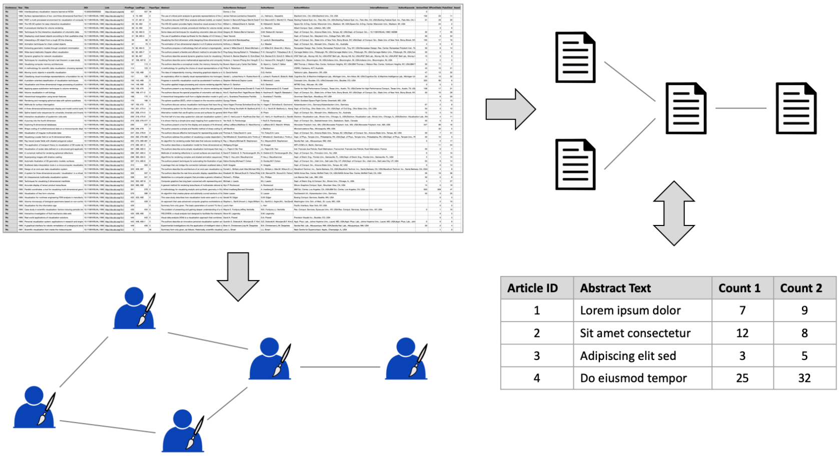
1.5 Limitations

As stated in Section 1.2, we limit our focus to similarity calculations since we need a manageable scope for the thesis. The consequence of this choice is that, depending on the circumstances, our proposed methodology is not necessarily directly transferrable to other computational tasks such as clustering and classification. The reason for this is that our methodology for combining embeddings can be viewed as a method for obtaining several yes-or-no-votes regarding a specific question, which in our case is *"Are these two entities similar?"*. However, in the context of clustering, our methodology could be seen as providing several different measurements of the distance between two items and most clustering algorithms cannot make use of this extra information.

Furthermore, the chosen similarity-based analysis approach implies that similarity relations will be our viewport to the underlying MVN and act as a filter for what is shown or not shown. Since a relation requires at least two items, this means that this approach is not intended for tasks dealing with single-item characteristics, such as for instance *"Show all items with a value of attribute X that is higher than 0.5"*. Instead, a typical example of similarity-based analysis would be a request such as *"Display all items which are similar to a selected target"* (and at the same time filter out all items which are dissimilar). Therefore, our proposed proof-of-concept visualizations are designed to highlight, and exploit, the parts of an MVN where interesting similarity patterns are found, and they should not be regarded as visualizations for conveying the full details of the MVN. With this in mind, we need to clearly state that our aim with this thesis is to showcase the possibilities and the strengths of the similarity-based approach—but at the same time we do not intend to give the impression that this approach is relevant for all MVN analysis scenarios and tasks.

1.6 Data Set

All our proof-of-concept applications are based on the IEEE VIS data set [43] which contains 18 different features for articles published at the IEEE VIS conferences. The choice of the data set is motivated by its quality, visibility in our research community, and our own familiarity with its topic. From this set, we have extracted roughly 3,000 articles published during the period 1990–2018, and in our visualizations we exploit the corresponding citation network, the corresponding co-author network, the abstract texts and two different numerical citation counts (see Figure 1.3).



Article ID	Abstract Text	Count 1	Count 2
1	Lorem ipsum dolor	7	9
2	Sit amet consectetur	12	8
3	Adipiscing elit sed	3	5
4	Do eiusmod tempor	25	32

Figure 1.3: *From the original data set, we extract the co-author network and the article citation network. For the article nodes of the citation network we also extract the abstract text and two different citation counts.*

The content of the chosen data set naturally makes our proposed proof-of-concept visualizations lean towards tasks from the fields of bibliometrics and scientometrics . However, we would like to point out that a great deal of our suggested approach is generic and applicable to any embeddable MVN. The reason for this is that the methodology operates on the level of the embedding vectors, and not on the level of the underlying data.

1.7 Thesis Outline

So far, the motivation behind our work and the aims and goals of the thesis were described together the research approach and its limitations. The rest of this thesis is organized as follows. Chapter 2 presents an overview of previous research from different fields that are relevant for our specific problem area. Chapter 3 contains a general methodology for combining embeddings to augment the quality of similarity calculations. This is a key concept of our contribution and together with it, a prototype VA tool for helping the analyst to find optimal ensembles is presented. In Chapter 4, we extend the general methodology to a second level of combining embeddings by showing how a MVN can be divided into multiple separately embeddable aspects. We also present a prototype VA application which implements the full range of our proposed methodology and allows the analyst to perform similarity-based exploration of the citation network of our data set. Finally, in Chapter 5, we discuss the overall results and implications of our work in view of the research goals that were set out for this thesis.

Chapter 2

Related Work

Contents

In this chapter we present previous research that is related to our work. We start with the key concepts of embedding technology and ensemble methods, which will both lay the foundation for our novel VA methodology. To set the context for our proposed prototype applications, we then proceed with an outline of the field of visual analytics and MVN visualization. Finally, to get an overview of the use case domain, we conclude with the fields of text similarity calculations, bibliometrics, and scientometrics.

2.1 Embeddings

Embeddings are (often low-dimensional) numeric vector representations of complex and/or unstructured data, created in order to be suitable for computational analysis tasks such as clustering, classification, and similarity calculations [8]. The main goal of embedding algorithms is usually to produce embeddings where items that 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. This makes embeddings highly suitable as input for computational analysis tasks, such as clustering, classification, and similarity calculations. The reason for this is that it is more straightforward to calculate a distance measure, such as Euclidean or cosine distance, with numeric vectors than it is with other types of complex and/or unstructured data [38,56,57]. How well the embedding captures the underlying targeted similarity is crucial since poor quality embeddings will elevate the risk for poor quality results when used in any further calculations.

2.1.1 Word and Text Embeddings

In general, word embeddings are distributed representations obtained from unsupervised training of a deep learning model on some large corpus of natural language text [2,8,9,22,96]. By using a large amount of training data to predict words given a specific context (or vice versa), the model will be able to learn semantic similarities of word pairs, e.g., *good* is similar to *super*, and *bad* is similar to *awful*. The algorithm then projects such similar word pairs to embedding vectors that lie close to each other in the embedding space [3,95]. Arguably, the single most influential word embedding technology is Word2Vec, which was introduced in 2013 [68], and arguably, the current state-of-the-art is the BERT model [25]. There are different approaches on how to use word embeddings to obtain embeddings for sentences or paragraph-sized text [61], starting from the intuitive (but limited) approach to take the average of the embeddings of each word in the text. However, sophisticated approaches are needed in order to exploit the syntactical structure of sentences. This is crucial to do since the same set of words may be arranged to form sentences with very different meanings, and the same word may have different meaning depending on the context [70]. To do so, the use of deep learning models is a popular choice, and approaches have, for example, been developed for recursive neural networks [92], convolutional neural networks [48], and recurrent neural networks [58]. Arguably, the current state-of-the-art technology for text embedding is the Universal Sentence Encoder (USE) [18].

2.1.2 Graph and Network Embeddings

Embedding calculations are not exclusive to textual data, for instance, they can be applied to various important tasks and applications involving graph and network data [72,89]. Technology for graph embedding, also known as Representation Learning on Graphs [40], 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 [35,36]. Furthermore, even dynamic aspects can be taken into account for embedding purposes [73]. The field of network embedding [104] is closely related to the field of graph embedding. The main difference is that in addition to the graph topology some (or all) of the attributed data is also considered, which allows for a more elaborated embedding process. Consequently, this type of technology is sometimes referred to as Attribute Enhanced Representation Learning [23].

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 of the classifiers on its own [26,76]. The *bagging* approach [15] involves classifiers of the same type of algorithm that are trained in parallel, and the final combined result is obtained by applying a deterministic algorithm (e.g., average or majority vote) to the set of individual predictions. In contrast, the *boosting* approach [16] organizes the training of classifiers of the same type of algorithm are trained in sequence, while each misclassification is given a higher weight of importance in the training of classifiers. Hence, classifiers added late to the ensemble will have put more focus on correctly classifying items that were misclassified by early added classifiers. In this way, the total ensemble will be actively steered towards having the potential for correctly classifying a major part of all the items. The final combined result is obtained by taking linear combination (inversely weighted by the error of each classifier) of the individual results. Another alternative approach is *stacking* [101]: classifiers of different types of algorithms are trained in parallel, and the final combined result is obtained by applying a deterministic algorithm to the set of individual predictions or by using this set to train a meta-model for making the final decision.

2.3 Visual Analytics

Previous information visualization (InfoVis) and visual analytics contributions have provided guidelines for designing, implementing, and evaluating interactive solutions that allow users to gain and externalize knowledge [5,84] about data and complex computational analyses. Such approaches often rely on individual or multiple interactive views [41,81] designed to facilitate certain user tasks [14], including comparison [32], provenance [80], and guidance [17], among others. The existing work in InfoVis and VA covers multiple techniques supporting various data types, including texts [45,60] and graphs/networks [53,75], and various applications, including the analyses of social media [21] and scientific publications [29], for instance.

One core idea of VA is the involvement of human analysts in complex computational analyses via interactive user interfaces. The necessity for such *human-in-the-loop* approaches was recognized decades ago, for instance, by the operations research community [30], and applied for the tasks associated with combinatorial complexity [79], human-guided search [59], and multiple criteria decision making [28]; the survey by Meignan et al. [67] covers this field of interactive optimization methods in operations research. Some of the relevant contributions for this problem originated from the VA community, including the visual optimization techniques for RFID benchmarking by Wu et al. [102], visual multiobjective optimization approaches by Berger et al. [11,12], and hybrid visual steering technique for simulation ensembles by Matković et al. [66]. Further review of visual analytic methods for interactive optimization is provided in the recent work by Hakanen et al. [39]. Our work shares the idea of involving the human analyst in the interactive search for optimized configurations via VA. But in contrast to the approaches discussed above, our work has a focus on different models and tasks, as discussed next. Recently, the attention of the VA research community has been drawn to the various problems in ML [19, 27, 82, 83, 88, 93]. Multiple approaches have been proposed for facilitating the ML training process, including, for instance, ManiMatrix by Kapoor et al. [50], which supports interactive optimization for multiclass classification problems. More specifically, VA approaches have been applied for the results of embedding calculations and also for the purposes of understanding such embeddings better. For instance, Embedding Projector [91] applies dimensionality reduction (DR) methods to display a projection plot for embedding vectors while allowing the users to search and inspect the underlying data items in the original space. cite2vec by Berger et al. [10] focuses on the particular task of interactive citation-driven document collection exploration that is based on joint word-document embeddings. ConceptVector by Park et al. [77] allows the users to construct lexicon-based concepts for text analysis purposes, which involves interaction with the output of one of the supported word embedding algorithms. Another relevant application for document collection analysis is discussed by Ji et al. [46], who make use of a paragraph embedding approach in their VA system. Word Embedding Visual Explorer by Liu et al. [63] focuses on the investigation of semantic relationships in word embeddings; this approach is supplemented by a case study with a comparison of embeddings produced by two algorithms, Word2Vec [68] and GloVe [78]. Liu et al. [64] discuss Latent Space Cartography, a more general approach for interactive analysis and interpretation of latent spaces and distributed representations, which includes the task of comparing latent space variants (i.e., embeddings), among others. embComp by Heimerl et al. [42] allows the user to explore word similarity between two different corpora, or for the same corpus embedded by two different methods, by analysis of nearest neighbors within the two embedding spaces. Finally, Parallel Embeddings by Arendt et al. [6] support exploration and comparison of clusters and cohorts of embedded data over time. While the contributions discussed above provide an important foundation for visual analysis of embeddings, with our proposed workflow, the focus is on investigation and comparison of not only *individual* embeddings for the given data sets, but rather *ensembles* of multiple embedding types used for joint decision making. Here, we should acknowledge the existing works discussing VA support for ensemble learning, including EnsembleMatrix by Talbot et al. [94], the workflow discussed by Schneider et al. [87], StackGenVis by Chatzimparmpas et al. [20], and ExMatrix by Neto and Paulovich [71], for instance. However, these approaches address construction of ML model ensembles for tasks such as classification, while the focus of our proposed approach is on investigating ensembles of *embeddings* for similarity analyses, affecting our methodology correspondingly.

2.3.1 MVN Visualization

The problem of MVN visualization has attracted considerable attention and comprehensible overviews of the main ideas, techniques and challenges are given by Jusufi [47] and Kerren et al. [54]. Furthermore, Nobre et al. provide a state-of-the-art survey [74] of this field in which they introduce a classification scheme according to the four different axes: (1) choice of layout, (2) view operations, (3) layout operations, and (4) data operations. Using the terminology of this scheme we can conclude that the methodology proposed in this thesis has a large focus on the so-called *Data Operations*. In other words, we first use a computational approach (which in our case includes embedding technology and similarity calculations) to reveal interesting aspects of the MVN. Then, we design our visualizations to exploit these specific aspects rather than trying to capture the whole MVN. Relating this back to the classification taxonomy for visualization approaches introduced by Jusufi, and also to the choice of layout from Nobre, we can see that our contribution could, for instance, be displayed as one view in a set of multiple coordinated views (or any other layout strategy proposed in these two works) showing different aspects of the underlying MVN.

2.4 Text Similarity Calculations

As shown in the survey by Wang and Dong, calculating text similarity is a generic task with many important applications within several different fields [98]. There are two major subgroups of methods for calculating the similarity between two text documents: word based and embedding based. The main advantage of the word based group is that it is conceptually simple and easy to implement, while the main disadvantage is that pre-processing of the text is usually needed and that semantic similarity is not supported. The main advantage of the embedding-based group is that it can handle semantic similarity and exploit syntactical structure, while the main disadvantage is that they are complex to implement and require substantial training. The disadvantage of training can however be alleviated by using pre-trained models. One of the earliest word-based methods is the Jaccard index/similarity [65] which is calculated by dividing the number of unique common words by the total number of unique words. A more sophisticated, and very popular word-based method, is the TF-IDF-method [85] in which a vector representation of the text is created with a dimension for each unique word in the corpus and with the values calculated as the number of occurrences of the word in the document divided by the number of occurrences of the word in the corpus. The similarity score can then be calculated by using the document vectors for computing, for instance, the cosine similarity value.

As already mentioned in Subsection 2.1.1, most embedding based text similarity methods make use of some form of deep neural networks to compute a vector representation of the text. A similarity metric of choice can then be calculated by using the embedding vectors.

2.5 Bibliometrics and Scientometrics

The concept of bibliometrics can be described as *"the application of mathematical and statistical methods to books and other media"*, and within the subfield of scientometrics the focus lies on analyzing the quantitative aspects of scientific publications and their use. Ranking of publications and authors as well as generation of various aggregated statistical representations are common tasks, often in combination with visualization techniques to facilitate a better understanding of the underlying data [69]. So called *distant reading* (i.e., using representations which convey information from the underlying text without the need for actually reading it) is an important concept that has been introduced to alleviate the inherent limitations of normal reading, which in turn is often referred to as *close reading* [45]. Since close reading is time consuming, and time typically is a limiting factor, there is a high demand for distant reading applications which support the navigation of large document sets and convey relevant aggregated information, but still also allow on-demand access to the underlying text for detailed examination. Natural language processing (NLP) in combination with visualization has proved to be a successful combination for tackling such challenges. Belinkov and Glass survey the impressive computational progress that has taken place in the field of NLP since the introduction of neural network models [7]. Kucher and Kerren [60] provide a taxonomy for, and an overview of, existing methods for text visualization. The survey of Federico et al. [29] focuses on visual approaches for analyzing scientific literature and patents while Liu et al. [62] target visualization and visual analysis of scholarly data. Finally, the BioVis Explorer by Kerren et al. [52] provides a way to navigate BioVis publications, and their connections, based on their respective visualization techniques. As can be noted from several of these publications, the scholarly domain in general, and the research domain in particular, are in themselves good examples of the bibliometric and scientometric challenges since the publication rate, in many research fields, makes it hard for any practitioner to maintain an overview and identify the most relevant information. A final observation that is relevant to our work is that it is not uncommon for corpus exploration to be in part driven by questions like *"Are there any groupings of similar documents within the set?"* or *"Are there documents which are similar to this specific document?"*. Therefore, the ability to exploit similarity relations [33] can be highly relevant for providing useful insights.

Chapter 3

Multiple Embedding Similarity Calculations

Contents

3.1 Introduction

In this chapter, we lay out the foundation of our proposed methodology, starting with the observation that the search for new and better ways to embed different types of data has attracted a lot of interest in recent years. For some data types, such as graphs/networks and words/text, there exist several different algorithms, each with its specific characteristics and trade-offs [3,23,35]. As a consequence, choosing the best embedding technology for a given application is an important and often non-trivial task. A straightforward way to handle this type of choice would be to: (1) choose a quality metric of importance for the current application, (2) evaluate all algorithms on this metric on a representative data set, and then (3) choose the one with the highest score. This intuitively appealing strategy provides a deterministic way for an optimal single-component choice and is in line with the existing works [13,49,95]. However, one might also consider an alternative approach inspired by the question: *"Would it be possible to combine several different embedding types as a means to achieve higher quality?"*, and this is the path that we will explore in this thesis.

Our starting point for this new approach is the observation that ensemble methods (i.e., different strategies for combining the results from several classification algorithms) are commonly used for augmenting the quality of the results for supervised classification problems [26, 76]. Hence, much in the same way, if a similar methodology could be applied to embeddings, this would open for the possibility to leverage already existing embedding technologies. Furthermore, for certain situations, this approach could hold the potential to outperform any of the single embeddings (used on their own) and achieve state-of-the-art results. In this thesis, we are therefore proposing a novel way to apply ensemble methods to embedding-based similarity calculations. To the best of our knowledge, there has been no previous research specifically targeting this possibility.

The task of constructing effective ensembles of embeddings while maintaining a clear picture of the respective process and results is related to the general challenges of interpretability, explainability, and trustworthiness in machine learning (ML) and artificial intelligence (AI) [1, 4, 31, 37]. One strategy proposed for these challenges is to make use of perceptual and cognitive abilities of human analysts, allowing them to construct and interact with ML models through the means of interactive visual analytic (VA) solutions [19, 27, 82, 83]. In particular, several VA approaches focusing on exploration or comparison of *individual* embedding algorithms have been discussed in the literature, including the works by Smilkov et al. [91], Park et al. [77], or Ji et al. [46], for instance. With this in mind, and also following the methodology of VA [51, 55], we have attempted to bring the human analyst into the process by developing a prototype VA tool to help the analyst with the task to construct effective ensembles (see Section 3.4). While providing a visual representation of the optimization process [39, 67], the use of the tool also gives direct insights to the inner workings of the ensemble calculations, and hence it supports the construction of a mental model of this complex process [5, 84]. The high-level workflow of our proposed methodology is depicted in Figure 3.1, and our hope is that our work will contribute to the field of human-centered AI in the sense that our application helps to open up the "black box" of ML, leading to better understanding and higher trust.

In the following sections of this chapter, we first develop a general methodology for applying ensemble methods to embedding-based similarity calculations. Then, we showcase this methodology on two different use cases with the help of a prototype VA application.

3.2 General Methodology

Although similarity calculations using embeddings are not equivalent to item classification problems (since the former is a way to score a relation, and the latter is a way to classify an item), there are some resemblances that we will exploit to make our adaptation. First, we note that similarity calculations over

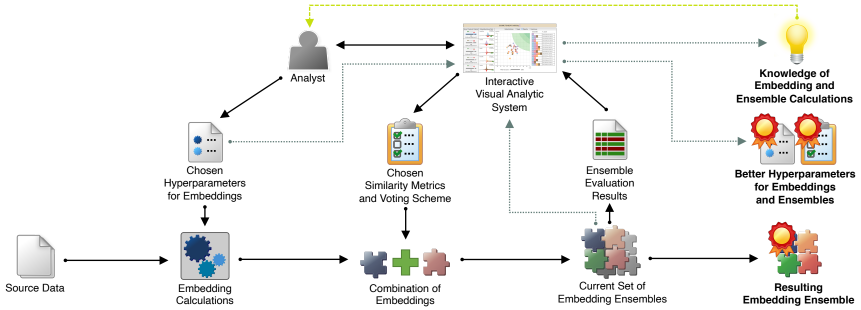


Figure 3.1: *Using a visual analytic approach, the analyst can combine multiple embeddings computed for the given data and investigate the performance of the ensembles with regards to the specified metrics and the voting scheme. This leads to better-performing embedding ensembles, better hyperparameter settings, and improved knowledge of—and trust for—embedding calculations and ensembles of embeddings. This process is inherently iterative (as expressed by the presence of cycles within the graph) and dashed and dotted edges represent indirect interactions.*

a set of embeddings typically will assign similarity scores to all item pairs in the set. Second, we observe that a common way for applications to exploit the similarity scores is to introduce the concept of similarity score threshold(s) that divides the set of all item pairs into (at least) two subsets depending on how similar/dissimilar they are. We may therefore regard a similarity calculation with a single similarity score threshold as a classification of a pair of items into one of the two classes, *similar* and *dissimilar*. Loosely speaking, we may view this as creating a new set (where the items to classify are all possible item pairs from the original set) and then performing binary classification on these new items. Based on this reasoning, we therefore conclude that: (1) if it is possible to obtain different embeddings for the same underlying data item (e.g., by using different algorithms or by using the same algorithm with different hyperparameter settings), then (2) it should be possible to combine these different embeddings by ensemble methods to yield a combined result for the similarity calculations. In other words, our proposed methodology is to use several different embeddings to calculate several similarity scores for a given item pair and then combine these scores to obtain a final classification. For instance, a straightforward way to combine the results of several embeddings would be to use the concept of bagging (outlined in Section 2.2) and to apply a majority voting scheme. This combined result would then, hopefully, have the potential to outperform similarity calculations using any of the contributing embeddings by themselves (see Figure 3.2 for a generic example).

In other words, the combiner function handles the pairwise scores (or classifications), but ignores how they were calculated. Therefore, the different embedding types might very well differ in aspects such as dimensionality and

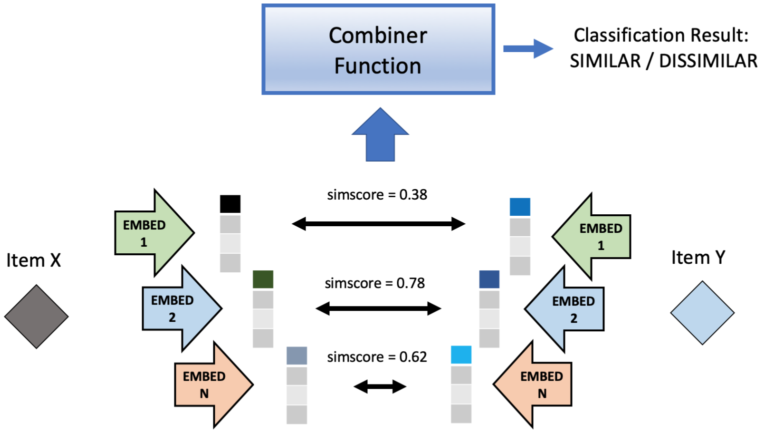


Figure 3.2: *A generic example. The underlying data items are embedded in several different ways (i.e., by using different algorithms or by using the same algorithm with different hyperparameter settings) and the pairwise similarity scores are calculated for each embedding type. The scores are then combined to yield a final combined classification of* similar *or* dissimilar*. The combiner function can range in complexity from a simple voting scheme up to a separately trained machine learning model.*

embedding space since they will not be mixed in the calculations. Furthermore, we observe that when calculating similarity scores, the embeddings are treated as pure numerical vectors and that no assumptions are made on the type of the underlying data. Thus, our proposed methodology generalizes to any data type that can be embedded with a similarity-preserving embedding technology. Finally, we note that the choice of combiner function and the score threshold values act both as hyperparameter settings for the ensemble calculations. Hence, trying to achieve the highest possible quality would be equivalent to searching for the best-performing hyperparameter settings for the ensemble calculations.

The last vital piece is an equivalent to the training step, during which the ensemble performance is evaluated against sets of already correctly labeled (i.e., correctly classified) training and verification data. The goal for this step is both to obtain a high performance score on the training data and, equally important, to obtain a good generalizability to previously unseen data. As an equivalent to the labeled training set, we introduce the concept of *"ground truth" (GT) sets* that are used for the performance evaluation, see Section 3.3. Hence, finding the best possible hyperparameter settings with regards to the GT set is, in essence, equivalent to the training process. In our example use cases (see Section 3.4), we give one example of working with a GT set that is fully known (Use Case 1, graph reconstruction) and one example of working with a partially known GT set (Use Case 2, text similarity). By demonstrating that our proposed methodology can be used also with partial and small GT sets, we want to highlight the fact that it is applicable to real-world scenarios and does not require ideal conditions.

To summarize: in complement to the search for high-quality single embeddings, we are proposing an alternative strategy that seeks to exploit the variance and the different advantages of several different embedding algorithms. This provides an alternative method for situations when training a single algorithm to achieve the required quality threshold is deemed to be unfeasible. This is also in analogy with supervised classification, where combining several "weak" classifiers can be a better choice than opting to train a single "strong" classifier [86].

3.2.1 Task Analysis

We end the presentation of the general methodology by listing, and briefly outlining, some of the tasks that we find to be the most important in the process of combining embeddings. Since this is a new methodology, this selection is mainly based on our experience and best knowledge and it is also inspired by the review of related literature. As can be noted, the proposed tasks have high resemblance to those performed when using ensemble methods for unsupervised classification.

(T1) Assess component interdependency. Assessing the interdependency of the participating components can give important insights into how a fruitful ensemble combination can be constructed. Since a high dependency will probably lead to similar results of each component, a combination consisting only of dependent components may not handle problematic cases well since all components risk to be wrong. On the other hand, an ensemble combination of components with low interdependency may handle such situations better since the diversity of the individual results can be higher. However, opting for a combination with lowest possible interdependency among the components does not automatically make for a successful ensemble.

(T2) Assess one-by-one performance. Assessing the one-by-one performance is an important step to gain understanding in the strengths and weaknesses of each component in terms of which cases it handles correctly and which cases that are problematic. Furthermore, it is also a vital step for finding the best possible single-component performance score which will then serve as a benchmark for the ensemble calculations. A suitable performance metric must be chosen before this task can be executed.

(T3) Assess ensemble performance. Assessing the ensemble performance is the last step before choosing the final ensemble configuration. In essence, this step can be seen as a search for optimal hyperparameter settings where the settings usually consist of a combination of (but not limited to): (1) the specific components to include in the ensemble, (2) the hyperparameter settings (if any) for each participating component, and (3) the combiner function for obtaining the final ensemble result.

(T4) Choose final configuration. The final ensemble configuration is chosen with regard to its performance on the test data and with regard to its assumed generalizability to previously unseen data.

3.3 Process

In this section, we outline the step-by-step process that we apply to our data. This process is in turn partly supported by our prototype VA tool, called EEVO, which will be presented in detail Section 3.4. We discuss our process based on two specific use cases dealing with fundamentally different underlying data types, see Section 3.4. For *Use Case 1* (graph reconstruction), we use the associated citation network of the articles; and for *Use Case 2* (text similarity), we use the article abstracts. While our use cases could potentially be part of realistic applications in scientometrics [29,90,103], we should emphasize that—for this chapter—these specific examples are not of the main interest, but rather the generic approach of combining different embeddings. A proof-of-concept application for MVN analysis based on the methodology described in this chapter will be presented in Chapter 4. Below, we describe the process together with specific details for each use case if needed.

Step 1 – Embed the Data

To obtain the different embeddings, we embed each data item by either using different algorithms, as in Use Case 1, or by using the same algorithm with different hyperparameter settings, as in Use Case 2. Of course, it would also have been possible to use a combination of these two approaches. For practical reasons, we have limited the number of embedding types to five.

Specific for Use Case 1: We embed the nodes of the citation network with five different neighbourhood-based embedding algorithms chosen more or less arbitrary: Node2Vec, RandNE, NetMF, BoostNE, and Laplacian Eigenmaps. The assumption is that the closer two pairs of nodes lie to each other in the citation network, the higher the similarity score that is yielded by their corresponding embedding vectors.

Specific for Use Case 2: We embed the text of each abstract in five different ways by using USE [18] on different parts of the text, as described below. Hence, the variation between the embeddings is due to which part of the underlying text that has been fed into the embedding algorithm. The assumption is that the more semantically similar a pair of abstracts are to each other, the higher the similarity score that is yielded by their corresponding embedding vectors.

Type 1 – Embed the first 400 characters of the text and thus capture similar beginnings, but ignore everything else.

Type 2 – Embed the last 400 characters of the text and thus capture similar endings, but ignore everything else.

Type 3 – Concatenation of Type 1 and Type 2. Capture abstracts with similar beginnings and similar endings.

Type 4 – Embed the full text and thus capture overall similarity, but with the risk of being "diluted" in the sense that it becomes more and more challenging to capture "a single meaning" as the text grows longer.

Type 5 – Embed keyword sentences extracted from the text and thus capture overall similarity, but with the risk of the keywords not being representative.

The rationale for the partitioning of the text is that there is often an implicit structure regarding what is written in the beginning and what is written at the end of an article abstract, and therefore this structure could (at least in theory) be exploited. Furthermore, the limit of 400 characters (which is a somewhat arbitrary choice) has been set in relation to the average length of the abstracts, which is just below 1,000 characters (or roughly 150 words in about 8 to 10 sentences). Setting it as a fixed limit instead of a relative limit in percentages alleviates the problem of ambiguity if a much shorter text is being compared to a much longer one.

Step 2 – Calculate the Pairwise Similarity Scores

We use cosine similarity to calculate the pairwise similarity scores for all item pairs and for all embedding types. Thus, for each item pair we obtain five different similarity scores (one for each embedding type).

Step 3 – Create a Ground Truth Set

To be able to assess and compare the quality of the similarity calculations we need to have some a priori knowledge of article pairs that are similar/dissimilar. We therefore create GT sets consisting of pairs which have been manually verified to be similar and pairs which have been verified to be dissimilar.

Specific for Use Case 1: As GT set of similar pairs we use the (about 13,000) existing citation links; and as GT set of dissimilar pairs, we use a set of the same size sampled from the set of node pairs without a direct citation link.

Specific for Use Case 2: As GT set of similar pairs, we use a small sampled set of 58 pairs that have been manually verified as similar, and 58 pairs that have been verified as dissimilar (coding was carried out by the author of this thesis and one colleague from the research group). It is important to note that these GT sets are not complete (as was the case in Use Case 1); and in Step 4 we will elaborate on what effects this brings to the calculations and the results. The rationale for using a small sampled GT set is twofold in that: (1) it is a daunting task to find all similar pairs within the set, and (2) we want to specifically demonstrate how the methodology can be used for real-world scenarios where no *a priori* GT set exists. With Use Case 2, we show that a small GT set of 58 pairs (out of a total of almost 5 million possible pairs) is still enough to perform a reasonably accurate performance ranking.

Step 4 – Calculate Single-embedding Performancet

In this step, the single-embedding ensemble performances are evaluated to get a benchmark for the coming multiple-embedding ensemble calculations. As performance metric, we use the F_1 score, which is calculated according to the following formula:

$$F_1 = \frac{2}{\frac{1}{p} + \frac{1}{r}}$$

Where p denotes the precision (which is calculated as the number of items from the GT set classified as similar divided by the total number of items classified as similar), and r denotes the recall (which is calculated as the number of items from GT set classified as similar divided by the total number of items in the GT similar set). For each single-embedding ensemble, we use the following incremental algorithm:

1. Set the starting similarity score threshold for the embedding to 1 and the granularity of the steps to 0.01.

2. Pairs with scores above or equal to the current threshold are classed as similar, and the others are classed as dissimilar. Check the current classification result against the GT set of similar pairs and calculate the current F_1-score.

3. Decrement the similarity score threshold by 0.01 and return to the previous step.

4. When finished (i.e., arrived at threshold -1) the maximal performance of the current embedding is the maximum of all the yielded F_1-scores.

5. *When max performance has been calculated for all embedding types*: the maximal F_1-score of all embedding types is chosen as our benchmark.

This benchmark represents the best possible result that we can achieve with a single-embedding strategy and calculating it answers the question *"If we could use only one type of embedding for this task, which one should we choose?"*. Thus, if we

Algorithm 1 Pseudocode for finding single embedding benchmark

```
$benchmark \leftarrow -1$
for all embeddings do
    $threshold \leftarrow 1$
    $max_{emb} \leftarrow -1$
    while $threshold \geq -1$ do
        $f_1 \leftarrow$ calculate current $F_1$ score
        if $f_1 > max_{emb}$ then
            $max_{emb} \leftarrow f_1$
        end if
        $threshold \leftarrow threshold - 0.01$
    end while
    if $max_{emb} > benchmark$ then
        $benchmark \leftarrow max_{emb}$
    end if
end for
```

can find a multiple-embedding ensemble that performs better, we would have sustained the claim that ensemble methods can work also within the field of embedding-based similarity calculations. As a final note, we would like to point out the fact that the process remains essentially the same even if another quality metric than the F_1 score is chosen.

Specific for Use Case 2: We use the same scheme as above for Use Case 2, but since we use a sampled GT set, the scores will only be approximative (since there will most probably exist similar pairs that have not been sampled). It is important to point out that this means that there are no strict guarantees that a higher scoring ensemble is actually performing better than a lower scoring one. To alleviate this problem, our suggested solution is to give the analyst a possibility to inspect the yields from the ensembles, so that a separate assessment of the quality of the similarity can be made (see Section 3.4).

Step 5 – Search for Optimal Ensemble Configuration

This process step is the main focus of our proposed visual analytics tool, and it will be covered in greater detail in the next section. The aim of this step is to determine: (1) which embeddings to combine, (2) which score thresholds to set, and (3) which combiner function to use to achieve the highest possible quality for the similarity calculations.

3.4 Visualization and Use Cases

In this section, we give an overview of our proposed interactive visual analysis tool, called EEVO, by showing how it can be used for our data set. EEVO is implemented as a web-based tool using D3 [24], and makes use of embeddings computed via the Google Colab platform [34]. As mentioned before, the rationale for providing a visualization for the optimization process is that the analyst can get important insights to the inner workings of the similarity calculations, and therefore in turn tune the process better to meet the current needs. Furthermore, similarity is an elusive concept (which very much lies in the eye of the beholder) so it is hard to capture by purely computational methods.

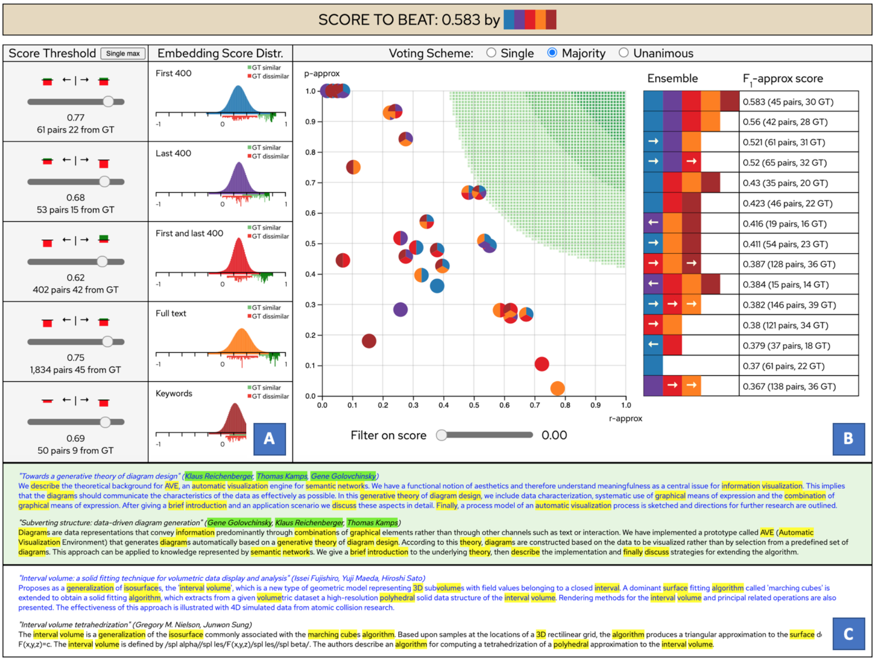
The visualization loads five different embedding types which means that we will have a total of $2^5 - 1 = 31$ possible ensemble combinations to evaluate (if counting single embeddings as ensembles as well). The performance of all these are continuously evaluated by the tool so the user does not need to make an active choice on which ensembles to track. Furthermore, EEVO allows for three different combiner functions in form of voting schemes, as specified below. The mental model of this is that: (1) each embedding type first provides a “vote" on whether a specific pair is similar or not, and (2) the votes are then combined according to the voting scheme to arrive at a final, unified classification of similar/dissimilar.

Single – A pair is classified as similar if at least one of the embeddings in the ensemble has classified it as similar.

Majority – A pair is classified as similar if more than half of the embeddings in the ensemble have classified it as similar.

Unanimous – All embeddings must classify the pair as similar.

As can be seen in Figure 3.3, the visualization interface of EEVO consists of three main views: (A) the Embedding View, (B) the Ensemble Performance View, and (C) the Similarity Assessment View (displayed on demand). In the Embedding View, the embedding score thresholds can be set, and the corresponding classification statistics can be assessed directly under the sliders. In the Ensemble Performance View, the voting scheme can be selected, and the ensemble performance is displayed on a scatterplot (which can be filtered using the "Filter on score" slider) and in a high-score table. The ensembles are represented by circular multi-colored glyphs in the scatterplot and by multi-colored rectangles in the high-score table. The color encoding (i.e., categorical attributes) corresponds to the colors of the participating embeddings. For instance, an ensemble that consists of the blue, red, and purple embeddings will have these three colors on its glyph and on its high-score rectangle. In the Similarity Assessment View, details of the text pairs classified as similar can be assessed and compared. To facilitate the analysis, common words are highlighted with yellow



0.77
61 pairs 22 from GT

0.68
53 pairs 15 from GT

0.62
402 pairs 42 from GT

0.75
1,834 pairs 45 from GT

0.69
50 pairs 9 from GT

Voting Scheme: ○ Single ● Majority ○ Unanimous

Filter on score 0.00

Ensemble	F$_1$-approx score
🟦🟪🟥🟧🟫	0.583 (45 pairs, 30 GT)
🟦🟪🟥🟧	0.56 (42 pairs, 28 GT)
🟦→ 🟪 🟧	0.521 (61 pairs, 31 GT)
🟦→ 🟪 🟥→	0.52 (65 pairs, 32 GT)
🟦🟥🟧🟫	0.43 (35 pairs, 20 GT)
🟦🟥🟫	0.423 (46 pairs, 22 GT)
🟪← 🟧 🟫	0.416 (19 pairs, 16 GT)
🟦→ 🟧 🟫	0.411 (54 pairs, 23 GT)
🟥→ 🟧 🟫→	0.387 (128 pairs, 36 GT)
🟪← 🟥 🟧 🟫	0.384 (15 pairs, 14 GT)
🟦→ 🟥→ 🟧→	0.382 (146 pairs, 39 GT)
🟥→ 🟧	0.38 (121 pairs, 34 GT)
🟦← 🟥	0.379 (37 pairs, 18 GT)
🟦	0.37 (61 pairs, 22 GT)
🟪 🟥→ 🟧→	0.367 (138 pairs, 36 GT)

"Towards a generative theory of diagram design" (*Klaus Reichenberger, Thomas Kamps, Gene Golovchinsky*)
We describe the theoretical background for AVE, an automatic visualization engine for semantic networks. We have a functional notion of aesthetics and therefore understand meaningfulness as a central issue for information visualization. This implies that the diagrams should communicate the characteristics of the data as effectively as possible. In this generative theory of diagram design, we include data characterization, systematic use of graphical means of expression and the combination of graphical means of expression. After giving a brief introduction and an application scenario we discuss these aspects in detail. Finally, a process model of an automatic visualization process is sketched and directions for further research are outlined.

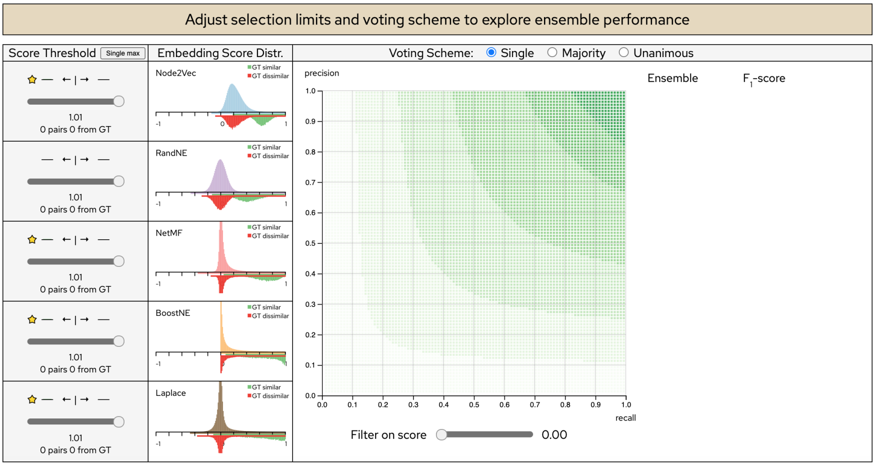
"Subverting structure: data-driven diagram generation" (*Gene Golovchinsky, Klaus Reichenberger, Thomas Kamps*)
Diagrams are data representations that convey information predominantly through combinations of graphical elements rather than through other channels such as text or interaction. We have implemented a prototype called AVE (Automatic Visualization Environment) that generates diagrams automatically based on a generative theory of diagram design. According to this theory, diagrams are constructed based on the data to be visualized rather than by selection from a predefined set of diagrams. This approach can be applied to knowledge represented by semantic networks. We give a brief introduction to the underlying theory, then describe the implementation and finally discuss strategies for extending the algorithm.

"Interval volume: a solid fitting technique for volumetric data display and analysis" (*Issei Fujishiro, Yuji Maeda, Hiroshi Sato*)
Proposes as a generalization of isosurfaces, the 'interval volume', which is a new type of geometric model representing 3D subvolumes with field values belonging to a closed interval. A dominant surface fitting algorithm called 'marching cubes' is extended to obtain a solid fitting algorithm, which extracts from a given volumetric dataset a high-resolution polyhedral solid data structure of the interval volume. Rendering methods for the interval volume and principal related operations are also presented. The effectiveness of this approach is illustrated with 4D simulated data from atomic collision research.

"Interval volume tetrahedrization" (*Gregory M. Nielson, Junwon Sung*)
The interval volume is a generalization of the isosurface commonly associated with the marching cubes algorithm. Based upon samples at the locations of a 3D rectilinear grid, the algorithm produces a triangular approximation to the surface d
F(x,y,z)=c. The interval volume is defined by /spl alpha//spl les/F(x,y,z)/spl les//spl beta/. The authors describe an algorithm for computing a tetrahedrization of a polyhedral approximation to the interval volume.

Figure 3.3: *Using the EEVO tool to visualize the performance of embedding-based ensembles conducting similarity calculations on paragraph-sized text (see further Section 3.4.2). In the left view (A), the similarity score thresholds can be adjusted and the corresponding results of the single-embedding similarity calculations are displayed. In the right view (B), a voting scheme can be selected and the performance scores of the ensembles are displayed, both in a scatterplot and in a highscore table. In the bottom view (C), the texts that have been classified as similar can be assessed and compared.*

spans, common authors are highlighted with green spans, and the enclosing rectangles of text pairs that belong to the GT set are color-coded with light green background color. When EEVO is loaded, all similarity score thresholds are put just above the score of the highest scoring pair for each embedding, and therefore all pairs start out by being classified as dissimilar (see Figure 3.4). When updating the hyperparameter settings (i.e., the similarity score thresholds or the type of voting scheme), the performance scores of all possible 31 ensembles are continuously updated, so that the analyst does not need to make an active selection on which ensembles to track. The design of EEVO is intended to facilitate the construction of a mental model of the transformation of single-embedding classifications (on the left-hand side) to the resulting ensemble classifications (on the right-hand side). An example of this can be seen in Figure 3.3, where the number of pairs classified as similar is much higher for some of the individual embeddings (see the numbers below the sliders) than for the combined ensemble



Adjust selection limits and voting scheme to explore ensemble performance

Score Threshold | Single max | Embedding Score Distr. | Voting Scheme: ● Single ○ Majority ○ Unanimous

Node2Vec: 1.01, 0 pairs 0 from GT

RandNE: 1.01, 0 pairs 0 from GT

NetMF: 1.01, 0 pairs 0 from GT

BoostNE: 1.01, 0 pairs 0 from GT

Laplace: 1.01, 0 pairs 0 from GT

Filter on score 0.00

Figure 3.4: *The initial view of EEVO. The default slider settings lead to all pairs being classified as dissimilar and therefore all ensembles have F_1 scores equal to 0 and are hidden. Setting new score thresholds will lead to pairs being classified as similar and ensemble glyphs appearing on the scatterplot and highscore table.*

results (see the numbers in the high-score table). This is due to the fact that we are using voting scheme Majority as combiner function, and it has a filtering effect. Using a different voting scheme would, of course, have yielded a different result.

3.4.1 Use Case 1: Graph Reconstruction

The mental model of the task (i.e., network reconstruction) is that 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. Thus, changing the score thresholds will yield a different reconstruction of the underlying citation network.

Assess embedding interdepencency.

Before making any adjustments to the hyperparameter settings, the analyst focuses on the *Embedding Score Distribution* column of the visualization (see Figure 3.4). Visualizing the pure score distributions (see Figure 3.5, left) does not allow for any deeper insights, but by keeping track of the scores of the pairs of the GT set, EEVO is able to provide more interesting details (see Figure 3.5, right). The analyst notes that there is a general, and encouraging, tendency for all the embeddings to assign higher scores to the similar pairs than to the dissimilar. He also observes the differences in how well the embeddings succeed in separating the two sets and the variation in distribution shapes (both regarding the upwards-oriented

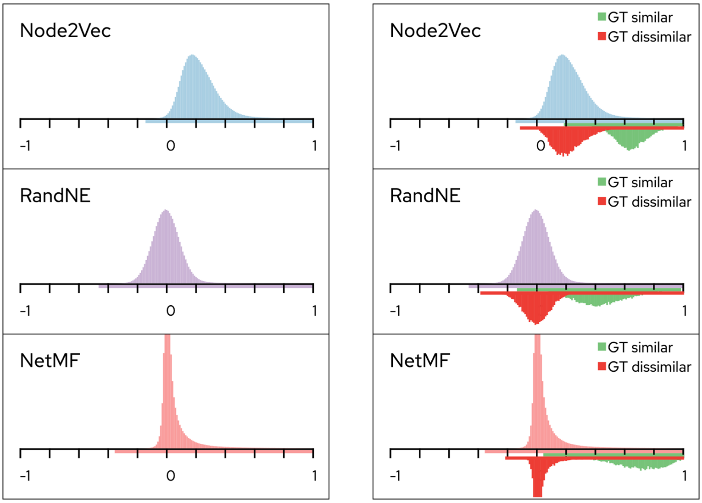
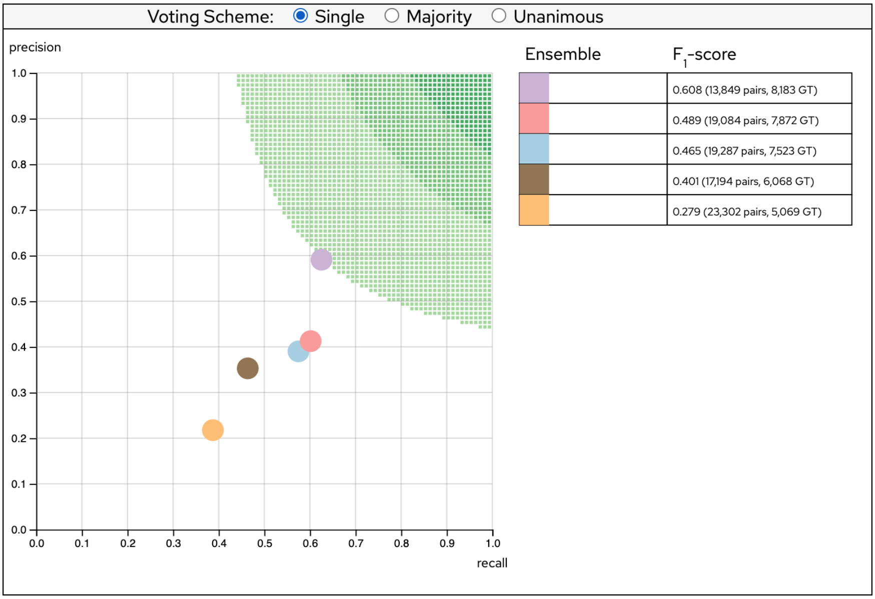


Figure 3.5: *The score distribution plots from three of the node embeddings of the citation network. From the pure distributions (left), we can observe that the embeddings distribute the scores differently over the pairs, but no further detailed conclusions can be drawn. However, adding the score distribution chart for the GT set (right; green and red for the similar and dissimilar pairs, respectively) clearly reveals that the different embedding types are not equally successful in separating the two subsets, and that there is usually no score threshold that yields a perfect split.*

total distributions and the downwards-oriented GT distributions) and anticipates that this could be exploited by an ensemble combination.

Assess one-by-one performance.
By adjusting the similarity score threshold on one slider at a time (leaving the others at their initial values), the analyst can assess the maximum performance score for each embedding one at a time. By doing so (and observing the scatterplot and the high-score table), it is straightforward to verify that the highest scoring single-embedding ensemble is *RandNE* and that the worst performing one is *BoostNE* (see Figure 3.6). As can be seen in the statistics cell of the high-score table, the optimal threshold score for RandNE corresponds to a network reconstruction with a total of almost 14,000 edges and just above 8,000 of those being correct, which gives a benchmark score of 0.608 (see Figure 3.6). Hence, this 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 an ensemble combination that performs better than the benchmark score found in the task above. As mentioned before, this is in essence a search for optimal hyperparameter settings within the parameter space of the possible voting schemes and the possible similarity score threshold values. To facilitate the search, the tool continuously calculates and displays



Voting Scheme: ● Single ○ Majority ○ Unanimous

Ensemble	F_1-score
🟪	0.608 (13,849 pairs, 8,183 GT)
🟥	0.489 (19,084 pairs, 7,872 GT)
🟦	0.465 (19,287 pairs, 7,523 GT)
🟫	0.401 (17,194 pairs, 6,068 GT)
🟧	0.279 (23,302 pairs, 5,069 GT)

Figure 3.6: *The maximum F_1-scores for individual embeddings in Use Case 1. The higher the score is, the more to the upper right corner the dot is situated. It is easy to see that RandNE (violet dot) achieves the highest score and that BoostNE (yellow dot) achieves the lowest. The green bands in the scatterplot indicate the distribution of the F_1-scores (i.e., any two points on the current border have the same score) and the border is reclined as higher scores are achieved. This gives the analyst a visual reference of the target area for new high scores.*

directive visual guidance [17], which conveys information on what would be 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 3.7), while the ensemble-specific guidance is displayed in the high-score table (see Figure 3.8). The aim of the design is that the analyst would combine the guidance with logical reasoning to augment the chances of finding optimal performing ensembles. Unfortunately, for most cases, this is not as easy as just following the guidance since some important aspects need to be considered:

Inconsistent guidance: A move that is beneficial for one ensemble may not be so for another, so the sum of all guidance may very well appear inconsistent.

Many vs. few: When in conflict, always choosing the move that is beneficial for the highest number of ensembles may not be the best strategy, especially when close to an optimum. Typically, the final moves will only be beneficial to the very top scoring ensemble(s).

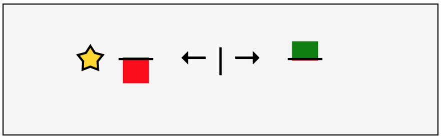


Figure 3.7: *Guidance above the score thresholds sliders indicates what would be the effect of moving the threshold one step up or down. The bar size encodes the number of affected ensembles, and the color and direction encode the potential effect: green and upwards for higher scores, red and downwards for lower scores. A star indicates that a new session high-score can be obtained. The guidance does not necessarily have to be consistent over all ensembles, since some may benefit from a move while others might not: here, a move to the left would benefit only one ensemble and be disadvantageous to several others. Nevertheless, it would still result in a new session high-score.*

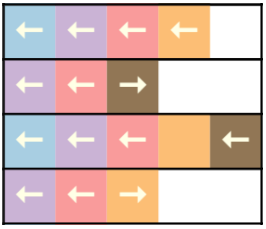
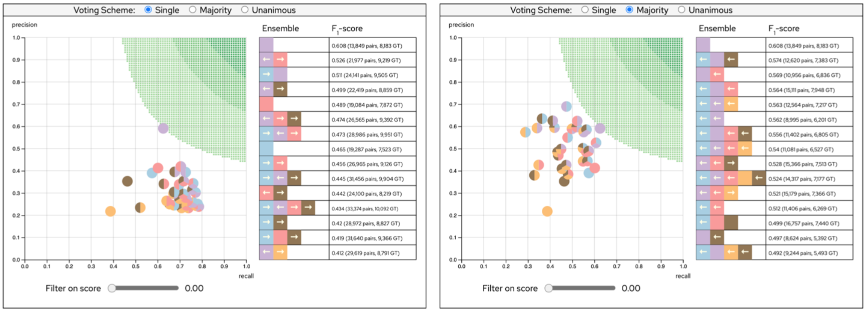


Figure 3.8: *Guidance in the high-score table indicates the moves that would be beneficial for a specific ensemble. Color encodes the embedding identity, and the arrow direction encodes in which direction the corresponding slider should be moved to augment the score of this ensemble: left means lowering the threshold and right means raising the threshold. As can be seen regarding the different arrow directions for brown and orange embeddings, the guidance does not necessarily have to be consistent over all ensembles, since some may benefit from a move while others may benefit from the exact opposite.*

Preserving potential: Existing guidance can be viewed as "potential for improvement" and is calculated under the premises that all other sliders are kept fixed. Therefore, if there is guidance on several sliders, moving one of them might very well "destroy" the potential on the others since the conditions now have changed. Thus, when given a scenario with potential on several sliders, it is prudent to proceed in smaller steps on alternating sliders and with readiness to backtrack. Otherwise, there is a risk of "over-shooting" a branching point where new (and possibly important) guidance would have been discovered.

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 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 task 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



Voting Scheme: ● Single ○ Majority ○ Unanimous

Ensemble	F_1-score
🟪	0.608 (13,849 pairs, 8,183 GT)
🟪← 🟥→	0.526 (21,977 pairs, 9,219 GT)
🟦→ 🟪	0.511 (24,141 pairs, 9,505 GT)
🟪← 🟫→	0.499 (22,419 pairs, 8,859 GT)
🟥	0.489 (19,084 pairs, 7,872 GT)
🟪← 🟥→ 🟫→	0.474 (26,565 pairs, 9,392 GT)
🟦→ 🟪← 🟥→	0.473 (28,986 pairs, 9,951 GT)
🟦	0.465 (19,287 pairs, 7,523 GT)
🟦→ 🟥→	0.456 (26,965 pairs, 9,126 GT)
🟦→ 🟪← 🟫→	0.445 (31,456 pairs, 9,904 GT)
🟥← 🟫→	0.442 (24,100 pairs, 8,219 GT)
🟦→ 🟪← 🟥→ 🟫→	0.434 (33,374 pairs, 10,092 GT)
🟦→ 🟫→	0.42 (28,972 pairs, 8,827 GT)
🟦→ 🟥 🟫→	0.419 (31,640 pairs, 9,366 GT)
🟪← 🟧→	0.412 (29,619 pairs, 8,791 GT)

Filter on score 0.00

Voting Scheme: ○ Single ● Majority ○ Unanimous

Ensemble	F_1-score
🟪	0.608 (13,849 pairs, 8,183 GT)
🟦← 🟪 🟫←	0.574 (12,620 pairs, 7,383 GT)
🟪 🟥←	0.569 (10,956 pairs, 6,836 GT)
🟦← 🟪 🟥←	0.564 (15,111 pairs, 7,948 GT)
🟦← 🟪← 🟧←	0.563 (12,564 pairs, 7,217 GT)
🟦← 🟪←	0.562 (8,995 pairs, 6,201 GT)
🟦← 🟪 🟥← 🟫←	0.556 (11,402 pairs, 6,805 GT)
🟦← 🟪← 🟥← 🟧←	0.54 (11,081 pairs, 6,527 GT)
🟪← 🟥← 🟫→	0.528 (15,366 pairs, 7,513 GT)
🟦← 🟪← 🟥← 🟧 🟫←	0.524 (14,317 pairs, 7,177 GT)
🟪← 🟥← 🟧→	0.521 (15,179 pairs, 7,366 GT)
🟦← 🟥←	0.512 (11,406 pairs, 6,269 GT)
🟦← 🟥 🟫	0.499 (16,757 pairs, 7,440 GT)
🟪 🟫←	0.497 (8,624 pairs, 5,392 GT)
🟦← 🟪 🟧← 🟫←	0.492 (9,244 pairs, 5,493 GT)

Filter on score 0.00

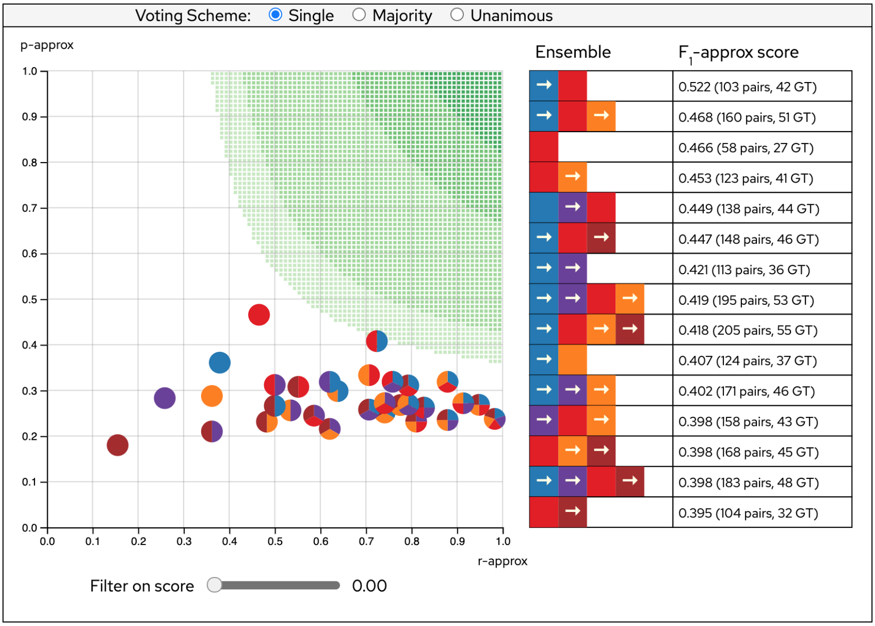
Figure 3.9: *By comparing the visual impressions of using the voting schemes Single (left) and Majority (right), the analyst concludes that Majority seems to hold a greater potential for success for the current slider positions. This decision (which might be wrong) is based on the fact that more of the ensembles appear to be positioned within "striking distance" of the benchmark score, and that there seems to be consistent guidance/potential on several sliders for several of the ensembles.*

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 3.9). By applying logical reasoning when following the main directions of the guidance and applying the strategy of alternating which sliders to move, the analyst can now fairly easy find an ensemble with a score of 0.621 (namely, RandNE + Node2Vec), which corresponds to an improvement of +2%. This could roughly be considered as the potential of about 250 links more being correctly reconstructed in the network. However, this is not the highest score that can be achieved, so the analyst must now continue the search by trying different starting points and different settings. By using the tool in this way, we have been able to find an ensemble with 5 contributing embeddings that achieves an improvement of +3% for this use case, although it might be just a local optimum.

3.4.2 Use Case 2: Text Similarity

The mental model of the task (i.e., text similarity comparison) is that we will assume that node pairs classified as similar have a high semantic text similarity, and that node pairs classified as dissimilar have low semantic text similarity. Thus, each possible combination of score thresholds (one for each embedding type) will yield a different number of pairs in each set.

Assess component interdependency.
For this use case, the distribution shapes are more similar to each other than in Use Case 1 (see Figure 3.3), and this is expected since we have used the same algorithm on different parts of the same text. However, the GT distributions reveal that there seems to be some variance in how the different embedding types



Voting Scheme: ● Single ○ Majority ○ Unanimous

Ensemble	F_1-approx score
🟦→ 🟥	0.522 (103 pairs, 42 GT)
🟦→ 🟥 🟧→	0.468 (160 pairs, 51 GT)
🟥	0.466 (58 pairs, 27 GT)
🟥 🟧→	0.453 (123 pairs, 41 GT)
🟦 🟪→ 🟥	0.449 (138 pairs, 44 GT)
🟦→ 🟥 🟫→	0.447 (148 pairs, 46 GT)
🟦→ 🟪→	0.421 (113 pairs, 36 GT)
🟦→ 🟪→ 🟥 🟧→	0.419 (195 pairs, 53 GT)
🟦→ 🟥 🟧→ 🟫→	0.418 (205 pairs, 55 GT)
🟦→ 🟧	0.407 (124 pairs, 37 GT)
🟦→ 🟪→ 🟧→	0.402 (171 pairs, 46 GT)
🟪→ 🟥 🟧→	0.398 (158 pairs, 43 GT)
🟥 🟧→ 🟫→	0.398 (168 pairs, 45 GT)
🟦→ 🟪→ 🟥 🟫→	0.398 (183 pairs, 48 GT)
🟥 🟫→	0.395 (104 pairs, 32 GT)

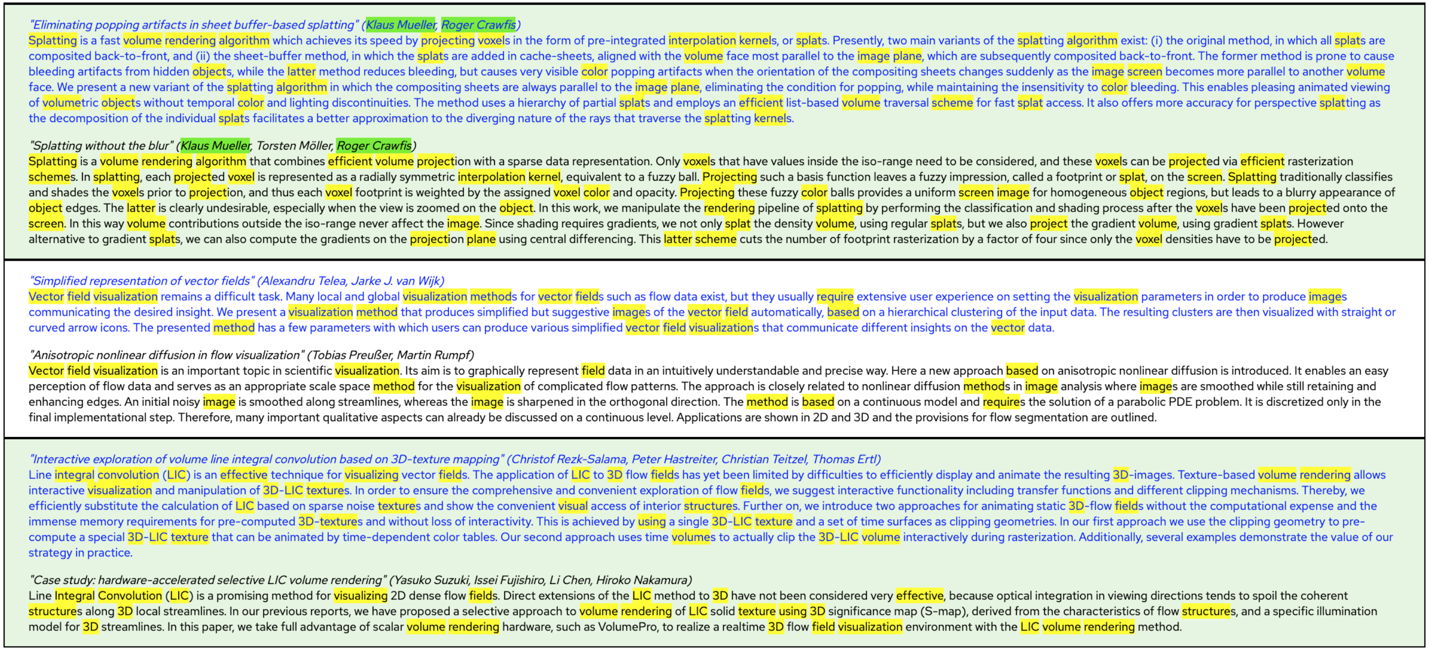
Filter on score 0.00

Figure 3.10: *When setting the sliders to the thresholds optimal for each single embedding, the analyst observes that using the voting scheme Single immediately yields a score that is higher than the benchmark for two of the ensembles. Following the guidance leads to even higher scores.*

distribute scores over the GT pairs, so the analyst hopes that this will be enough to be exploited in an ensemble combination.

Assess one-by-one performance.
By using the same methodology as for Use Case 1, the analyst concludes that the highest scoring single-embedding ensemble is *First and last 400* with a benchmark score of 0.466.

Assess ensemble performance.
Using the same approach as for Use Case 1 (i.e., setting the initial slider values to the position optimal for the corresponding single embedding), the analyst observes that a score higher than the benchmark is immediately yielded for the voting scheme Single (see Figure 3.10). Encouraged by this, the analyst follows the guidance on alternating sliders and is hereby able to achieve a score of 0.553, which is an improvement of 19%. However, since this use case makes use of a sampled GT set, the analyst cannot fully rely on the scores (especially if the difference is small), since they are approximative. Therefore, the analyst needs to assess the yielded classifications by using the Similarity Assessment View (see Figure 3.11) and determine which of the top scoring ensembles is truly the best (if needed, this view could also be used for Use Case 1). Furthermore, the achieved



"Eliminating popping artifacts in sheet buffer-based splatting" (Klaus Mueller, Roger Crawfis)
Splatting is a fast volume rendering algorithm which achieves its speed by projecting voxels in the form of pre-integrated interpolation kernels, or splats. Presently, two main variants of the splatting algorithm exist: (i) the original method, in which all splats are composited back-to-front, and (ii) the sheet-buffer method, in which the splats are added in cache-sheets, aligned with the volume face most parallel to the image plane, which are subsequently composited back-to-front. The former method is prone to cause bleeding artifacts from hidden objects, while the latter method reduces bleeding, but causes very visible color popping artifacts when the orientation of the compositing sheets changes suddenly as the image screen becomes more parallel to another volume face. We present a new variant of the splatting algorithm in which the compositing sheets are always parallel to the image plane, eliminating the condition for popping, while maintaining the insensitivity to color bleeding. This enables pleasing animated viewing of volumetric objects without temporal color and lighting discontinuities. The method uses a hierarchy of partial splats and employs an efficient list-based volume traversal scheme for fast splat access. It also offers more accuracy for perspective splatting as the decomposition of the individual splats facilitates a better approximation to the diverging nature of the rays that traverse the splatting kernels.

"Splatting without the blur" (Klaus Mueller, Torsten Möller, Roger Crawfis)
Splatting is a volume rendering algorithm that combines efficient volume projection with a sparse data representation. Only voxels that have values inside the iso-range need to be considered, and these voxels can be projected via efficient rasterization schemes. In splatting, each projected voxel is represented as a radially symmetric interpolation kernel, equivalent to a fuzzy ball. Projecting such a basis function leaves a fuzzy impression, called a footprint or splat, on the screen. Splatting traditionally classifies and shades the voxels prior to projection, and thus each voxel footprint is weighted by the assigned voxel color and opacity. Projecting these fuzzy color balls provides a uniform screen image for homogeneous object regions, but leads to a blurry appearance of object edges. The latter is clearly undesirable, especially when the view is zoomed on the object. In this work, we manipulate the rendering pipeline of splatting by performing the classification and shading process after the voxels have been projected onto the screen. In this way volume contributions outside the iso-range never affect the image. Since shading requires gradients, we not only splat the density volume, using regular splats, but we also project the gradient volume, using gradient splats. However alternative to gradient splats, we can also compute the gradients on the projection plane using central differencing. This latter scheme cuts the number of footprint rasterization by a factor of four since only the voxel densities have to be projected.

"Simplified representation of vector fields" (Alexandru Telea, Jarke J. van Wijk)
Vector field visualization remains a difficult task. Many local and global visualization methods for vector fields such as flow data exist, but they usually require extensive user experience on setting the visualization parameters in order to produce images communicating the desired insight. We present a visualization method that produces simplified but suggestive images of the vector field automatically, based on a hierarchical clustering of the input data. The resulting clusters are then visualized with straight or curved arrow icons. The presented method has a few parameters with which users can produce various simplified vector field visualizations that communicate different insights on the vector data.

"Anisotropic nonlinear diffusion in flow visualization" (Tobias Preußer, Martin Rumpf)
Vector field visualization is an important topic in scientific visualization. Its aim is to graphically represent field data in an intuitively understandable and precise way. Here a new approach based on anisotropic nonlinear diffusion is introduced. It enables an easy perception of flow data and serves as an appropriate scale space method for the visualization of complicated flow patterns. The approach is closely related to nonlinear diffusion methods in image analysis where images are smoothed while still retaining and enhancing edges. An initial noisy image is smoothed along streamlines, whereas the image is sharpened in the orthogonal direction. The method is based on a continuous model and requires the solution of a parabolic PDE problem. It is discretized only in the final implementational step. Therefore, many important qualitative aspects can already be discussed on a continuous level. Applications are shown in 2D and 3D and the provisions for flow segmentation are outlined.

"Interactive exploration of volume line integral convolution based on 3D-texture mapping" (Christof Rezk-Salama, Peter Hastreiter, Christian Teitzel, Thomas Ertl)
Line integral convolution (LIC) is an effective technique for visualizing vector fields. The application of LIC to 3D flow fields has yet been limited by difficulties to efficiently display and animate the resulting 3D-images. Texture-based volume rendering allows interactive visualization and manipulation of 3D-LIC textures. In order to ensure the comprehensive and convenient exploration of flow fields, we suggest interactive functionality including transfer functions and different clipping mechanisms. Thereby, we efficiently substitute the calculation of LIC based on sparse noise textures and show the convenient visual access of interior structures. Further on, we introduce two approaches for animating static 3D-flow fields without the computational expense and the immense memory requirements for pre-computed 3D-textures and without loss of interactivity. This is achieved by using a single 3D-LIC texture and a set of time surfaces as clipping geometries. In our first approach we use the clipping geometry to pre-compute a special 3D-LIC texture that can be animated by time-dependent color tables. Our second approach uses time volumes to actually clip the 3D-LIC volume interactively during rasterization. Additionally, several examples demonstrate the value of our strategy in practice.

"Case study: hardware-accelerated selective LIC volume rendering" (Yasuko Suzuki, Issei Fujishiro, Li Chen, Hiroko Nakamura)
Line Integral Convolution (LIC) is a promising method for visualizing 2D dense flow fields. Direct extensions of the LIC method to 3D have not been considered very effective, because optical integration in viewing directions tends to spoil the coherent structures along 3D local streamlines. In our previous reports, we have proposed a selective approach to volume rendering of LIC solid texture using 3D significance map (S-map), derived from the characteristics of flow structures, and a specific illumination model for 3D streamlines. In this paper, we take full advantage of scalar volume rendering hardware, such as VolumePro, to realize a realtime 3D flow field visualization environment with the LIC volume rendering method.

Figure 3.11: *By clicking a statistics cell in the high-score table, the Similarity Assessment View is displayed. The view is populated with the yield of the corresponding ensemble, and the analyst can evaluate the quality of the classifications. To facilitate the analysis, common words are highlighted with yellow spans, common authors are highlighted with green spans, and the enclosing rectangles of text pairs that belong to the GT set are color-coded with light green background color.*

score is not the highest possible one, so the analyst must now continue the search by trying different starting points and different settings. By using the tool in this way, we have been able to find an ensemble with 5 contributing embeddings that achieves an improvement of +25% for this use case, although this might be a local optimum as well.

As a final remark, we would like point out the fact that the current implementation of EEVO does not scale exceptionally well when loading many different embedding types. This is mainly due to the fact that it is very expensive to calculate the guidance, and this is also the main reason for why the guidance has been limited to only one step. With an optimized implementation it would be possible (and very valuable) to extend the guidance to also look "further ahead", as well as calculating the results of combinations of moves.

3.5 User Study

Evaluation is an important step for determining if a new interactive visualization approach is successful or not with regard to certain criteria, for instance, usability [44]. Since we are proposing a novel methodology for an area which has not, to the best of our knowledge, been studied before, it has not been possible for us to directly compare the proposed approach with any previous baseline approaches via controlled experiments or long-term case studies involving experts. Instead, we have opted to perform an initial user study which focused on two specific questions: (1) how well our proposed tool supports the user in finding high-performing ensembles, and (2) if the design of the tool is straightforward enough to allow even users without expert knowledge of embeddings and ensembles to succeed with the search. The study had a total of 6 participants from the field of computer science with the following profiles:

Participant 1 – Graduate at Master's level, general knowledge of ML and visualization.

Participant 2 – Senior lecturer, expert knowledge of ML and intermediate knowledge of visualization.

Participant 3 – Master's student, general knowledge of ML and visualization.

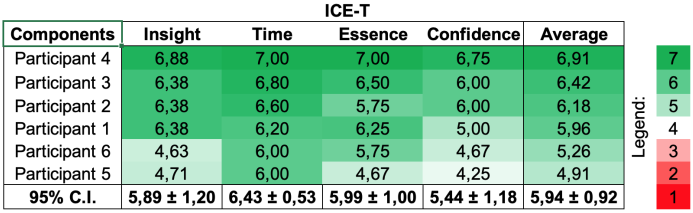
Participant 4 – Post doc, general knowledge of ML and intermediate knowledge of visualization.

Participant 5 – PhD student, expert knowledge of ML and intermediate knowledge of visualization.

Participant 6 – Post doc, intermediate knowledge of ML and general knowledge of visualization.

All sessions were individual with a maximal duration of one hour. Each participant was given an introduction to the EEVO tool and then spent approximately 20–30 minutes on the task of trying to find an ensemble which could outperform the best single embedding for Use Case 2 discussed in Section 3.4. At the end of the sessions, the participants were asked to give their overall impression of the tool and to fill out an ICE-T evaluation form [97] (this heuristic evaluation approach focuses on self-reported estimates of visualization value aspects such as its ability to decrease the time necessary for answering questions about the data, to facilitate discovery of insights, etc.) All of the participants were able to find an ensemble which outperformed the best single embedding. With regard to our focus questions, the results hence suggest that our tool can be used for its intended task, and also that it can be used without prior expert knowledge within the fields of embeddings and ensembles. Furthermore, the oral feedback given at the end of the sessions was consistent and could be condensed to the following:

- The tool was perceived as being user-friendly and having an appropriate design for the intended use.
- The continuous guidance provided good support for solving the task and the chosen visual metaphors were straightforward and easy to interpret.
- For better analysis of situations when the guidance is ambiguous or non-existent, an extended "guidance horizon" (beyond the current limit of one step left/right) would be beneficial.



ICE-T

Components	Insight	Time	Essence	Confidence	Average
Participant 4	6,88	7,00	7,00	6,75	6,91
Participant 3	6,38	6,80	6,50	6,00	6,42
Participant 2	6,38	6,60	5,75	6,00	6,18
Participant 1	6,38	6,20	6,25	5,00	5,96
Participant 6	4,63	6,00	5,75	4,67	5,26
Participant 5	4,71	6,00	4,67	4,25	4,91
95% C.I.	**5,89 ± 1,20**	**6,43 ± 0,53**	**5,99 ± 1,00**	**5,44 ± 1,18**	**5,94 ± 0,92**

Legend: 7 6 5 4 3 2 1

Figure 3.12: *The ICE-T scores with the participants sorted on average score. The respective ICE-T categories focus on the ability of the visualization approach to discover* insights*, decrease* time *for task solving, convey* essense *of the data, and generate* confidence *about the data [97]. Green is indicating good results, as opposed to red.*

For aggregating the results from the ICE-T questionnaire responses, we performed a numerical translation of the answer options to a scale from 1 to 7, with higher scores indicating better results. Figure 3.12 provides an overview of the scores, indicating that a majority of the participants have graded EEVO at the higher end of the scale. On the whole, our general assessment of the study setting and the obtained feedback is that the consistent and positive results provide support for the claims that the methodology is working, and that our proposed application can be used for the intended task.

Chapter 4

Similarity-based Network Exploration

Contents

4.1 Introduction

In this chapter we will add one more level of embedding combinations to the foundation that we have already laid. Our starting point is the observation that, for the specific problem of embedding MVNs, current research has explored several methods to embed both the network structure and the attributed data together [23]. However, separate embedding technologies for data types that are common building blocks for MVNs already exist (e.g., separate embeddings for network structure, word/text, categorical attributes etc.) [3,35]. This opens for an alternative approach where different *aspects* of the underlying MVN are first separately embedded and then combined to form the full embedding representation. Here, we explore such an *aspect-driven* approach on an attributed article citation network built from a large set of scientific publications and our *all-embedding* approach covers the aspects: (1) citation network topology, (2) the abstract text, (3) co-author information, and (4) numerical attributes, see Figure 4.1. On this base we build an interactive application, called Simbanex (short for similarity-based network exploration), which is intended to be used within the fields of bibliometrics and scientometrics and allows the user to perform interactive similarity-based exploration of the underlying set of scientific documents. To demonstrate the usefulness of this type of similarity-driven exploration, we present two different use cases, where the first focuses on citation link analysis and the second on topic similarity. Furthermore, we show that the proposed aspect-driven all-embedding strategy may be applied to any complex data that can be broken down into separately embeddable aspects, so the general methodology is generalizable and not only limited to MVNs. Nevertheless, the strategy also has some limitations and the main trade-off for the application design is between having a heterogeneous framework of several different technologies, or a homogeneous framework based on the same concept (in our case embeddings). We opt for the latter, since we want to explore how far we can come by mainly using, and combining, already existing and well-proven embedding technologies. The early versions of the Simbanex tool had a more narrow focus on text similarity only, and it was named Simbatex (short for similarity-based text exploration) [99].

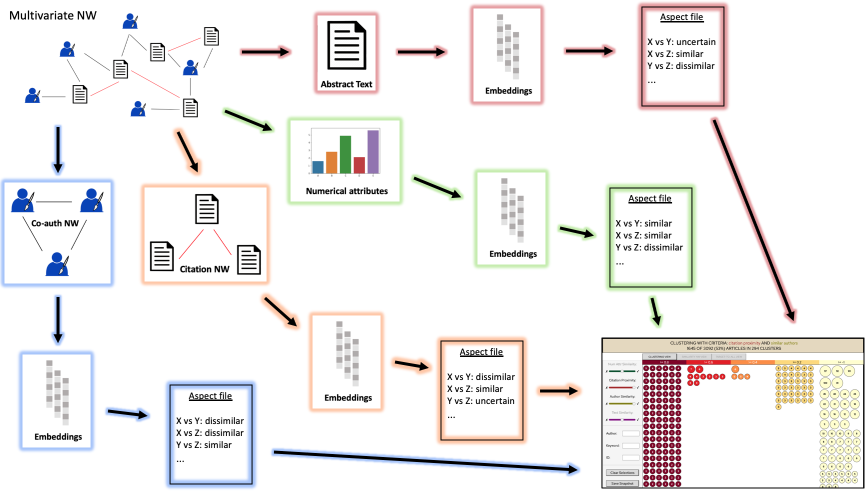


__Figure 4.1:__ A schematic view of how the aspect-driven approach has been applied to the data set. The underlying MVN is partitioned in to several different node-based aspects: co-author information (blue), position in the citation network (orange), numerical data (green), and the abstract text (red). Each aspect is embedded separately and then the pairwise similarity classifications (i.e., similar, dissimilar, or uncertain) are calculated using the methodology described in Section 3.2. The possibility to combine these aspect classifications allows for flexible construction of dynamic queries in the Simbanex application (see Sections 4.2 and 4.3). This methodology can be generalized to any complex entity which may be divided into separately embeddable aspects.

4.2 Process

In this section, we outline the step-by-step process that we apply to our data and the major computational concepts that are used within Simbanex. The main idea of our approach is is a two-level process which first uses the methodology developed in chapter 3 to obtain the best possible aspect classifications, and then combines these partial classifications to arrive at a final classification using the three classes *similar*, *dissimilar*, and *uncertain* (see further Figure 4.1). In other words: when comparing two entities we first divide them into several smaller parts that we can compare one by one, and we then deduce the similarity of the two entities based on these comparisons. A coarse grained list of the process is:

1. Divide the MVN into several different aspects
2. Embed each aspect separately
3. Calculate the pairwise similarity scores
4. Determine the aspect classifications
5. Save the classifications to file

To better illustrate the above, we will now go through how each step was applied to our data set.

Step 1 – Divide Into Aspects
To demonstrate the methodology, we have chosen the following four separate node-based aspects for the publications in our data set, see Figure 4.1.

1. The position in the citation network topology
2. The abstract text
3. The co-author information
4. The numerical citation counts

The rationale for this choice is that is a mix of data types which each provide a different example on how the embeddings can be combined for the similarity calculations. Furthermore, although limited, this set of aspects still illustrates the methodology well enough to provide an understanding of how more aspects could be added if needed. The advantage of using these four aspects separately, as compared to trying to capture them all at once with only one embedding, is that we will have a greater flexibility when specifying our search queries. Instead of just having the possibility of determining whether two articles are similar or dissimilar (with regards to all aspects at once) the aspect-driven strategy will give us the possibility to specify criteria that use separate combinations of them.

For example: *Articles with similar abstract text AND dissimilar authors AND lie far from each other in the citation network AND have similar citation counts*. Furthermore, we may also choose to exclude some of the aspects totally from a query and, for instance, search for publications with similar authors and similar citation counts regardless of the similarity of the abstract texts or the positions in the citation network.

Step 2 – Embed Each Aspect
For each publication, we now create several different embeddings for each aspect in the following way.

Position in the citation network: Exactly as in Section 3.3, we use neighbourhood-aware technology to embed the nodes of the citation network, and the assumption is therefore that the embedding vectors of article nodes that lie close to each other will yield a high cosine similarity value; and that the vectors of article nodes that lie far apart will yield a low cosine similarity value. We use three different algorithms (RandNE, Node2Vec, and Laplacian Eigenmaps) and therefore obtain three different embedding vectors for each publication regarding this specific aspect. The benefit of having several different embeddings for a specific aspect is that we can apply the methodology outlined in Section 3.2 when determining the final classification. Thus, we have the potential to obtain a better result than by using just one single embedding type.

Abstract text: Exactly as described in Section 3.3 we use paragraph text embedding technology to embed the abstract text of the publications and the assumption is therefore that the embedding vectors of abstracts that are semantically similar to each other will yield a high cosine similarity value, and that the vectors of abstracts that are semantically dissimilar will yield a low cosine similarity value. We use the Universal Sentence Encoder (USE) [18] to embed the text, and we obtain five different embedding vectors for each publication by feeding it five different "portions" of the text as follows:

Type 1 – Embed the first 400 characters of the text and thus capture similar beginnings, but ignore everything else.

Type 2 – Embed the last 400 characters of the text and thus capture similar endings, but ignore everything else.

Type 3 – Concatenation of Type 1 and Type 2. Capture abstracts with similar beginnings and similar endings.

Type 4 – Embed the full text and thus capture overall similarity, but with the risk of being "diluted" in the sense that it becomes more and more challenging to capture "a single meaning" as the text grows longer.

Type 5 – Embed keyword sentences extracted from the text and thus capture overall similarity, but with the risk of the keywords not being representative.

The rationale for the partitioning of the text is that there is often an implicit structure regarding what is written in the beginning and what is written at the end of an article abstract, and therefore this structure could (at least in theory) be exploited. Furthermore, the limit of 400 characters (which is a somewhat arbitrary choice) has been set in relation to the average length of the abstracts, which is just below 1,000 characters (or roughly 150 words in about 8 to 10 sentences).

Co-author information We use neighbourhood-aware technology to embed the nodes of the co-author network that is associated with our citation network. The assumption is therefore (in analogy with the embedding of the citation network) that the embedding vectors of author nodes that lie close to each other will yield a high cosine similarity value and that the vectors of author nodes that lie far apart will yield a low cosine similarity value. However, this time we need to handle the fact that there are usually several co-authors of a publication. So, to obtain a single embedding vector for each publication, we take the average of all the corresponding author node embedding vectors. Hence, the resulting embedding captures information of all co-authors, and it yields high cosine similarity values with vectors from articles which have co-occurring authors. We use the same three algorithms as for the citation network and therefore obtain three different embedding vectors for each publication also regarding this specific aspect.

Citation counts We embed each of the two numerical citation counts with a custom embedding which captures the following similarity rules:

1. For counts below 100, a maximum difference of 10 is allowed (e.g., 2 and 8 will be regarded as similar, but 5 and 17 will be regarded as dissimilar).

2. For counts between 100 and 500, a maximum difference of 50 is allowed (e.g., 122 and 161 will be regarded as similar, but 328 and 381 will be regarded as dissimilar).

3. For counts above 500 a maximum difference of 500 is allowed (e.g., 537 and 968 will be regarded as similar, but 1,044 and 1,613 will be regarded as dissimilar).

The rationale for the above set of rules is that it is not uncommon to have different "similarity binning granularity" depending on where on the scale we are measuring. For example, many people would probably regard a citation count of 2 to be quite different from 53, but at the same time regard a citation count of 438 to be quite similar to 502, although the distance between the two former is less than between the two latter. The specific bins have been determined by assessing the histograms of the citation counts of the data set, see Figure 4.2. Alternatively, the similarity rules could have been expressed as a maximum allowed percentage

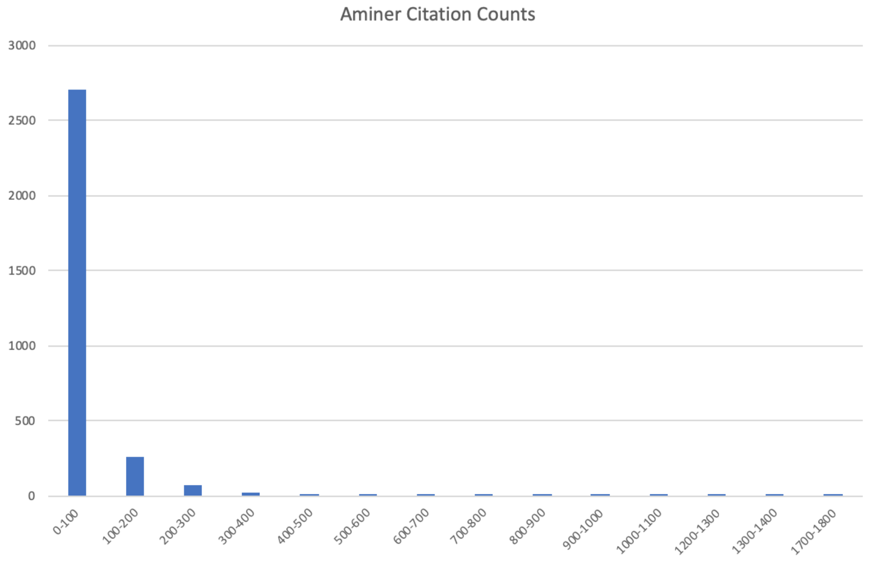


Figure 4.2: *The Aminer citation counts represented as a histogram with bin size 100. As can be seen, the bulk of the data is heavily skewed to the left with a long tail to the right. From this distribution, it is reasonable to argue for the use of different "similarity binning granularity" depending on location on the x-axis. Using only a large bin would make all of the observations to the left similar, and using only a small bin would make no observations to the right similar.*

difference between the numbers (i.e., X and Y are regarded as similar if they differ less than Z% from each other) but this is not straightforward to capture with an embedding.

Step 3 – Calculate Pairwise Scores

The next step, after embedding each aspect separately, is to calculate the pairwise similarity scores. As we have seen, a total of 13 different embedding vectors have been created for each publication in our data set, and we now calculate the pairwise cosine similarity scores for each embedding type. This yields 13 different similarity scores (for each article pair) which can be grouped by aspect, so that every pair now has 3 similarity scores for the position in the citation network, 5 similarity scores for the abstract text, 3 similarity scores for the author information, and 2 similarity scores for the citation counts.

Step 4 – Determine Aspect Classifications

First, we find the best possible score thresholds for each embedding (as described in Section 3.3, Step 4). We then introduce an *uncertainty interval* around these thresholds which is used in the following way: (1) pairs with scores above the interval are classified as similar, (2) pairs with scores below the interval are classified as dissimilar, and (3) pairs with scores within the interval are classified as uncertain. We then apply the following voting scheme: *" (1) If a majority of the embeddings vote for similar, the pair is classified as similar with regard to this aspect. (2) If a majority vote for dissimilar, the pair is classified as dissimilar with regard to this aspect. (3) If none of the previous holds true, the pair is classified as uncertain with regard to this aspect."* Thus, for each article pair this results in 4 separate classifications which describe how similar (or not) the articles are with regard to each of the 4 chosen aspects. The rationale for using an uncertainty interval is that it is usually hard to find a perfect single-threshold split for similar/dissimilar, and that it is often of interest to be able to identify, and further investigate, so-called *near misses*.

Step 5 – Save to File
Finally, we store the pre-calculated classifications into files that will be loaded into the visualization on start. This allows for queries with high responsiveness since the classifications can be directly mapped to the GUI slider positions (see further Section 4.3.1). Only minor further calculations are needed, and they can be performed within the browser which eliminates the need for a synchronous backend.

As can be noted from the content of this section, one major advantage of the proposed all-embedding strategy is that it gives a straighforward and homogeneous framework for calculating the similarity classifications, even for complex data types such as network topology and paragraph-sized text. Furthermore, we note that the methodology is generalizable beyond the scope of MVNs since the approach may be used on any complex entity that can be broken down into separately embeddable aspects. On the other hand: (1) it may be challenging to use embeddings to capture similarity in a way that obtains a good split of similar/dissimilar pairs, and (2) alternative methods still need to be considered for data types for which no suitable embedding technology exist. Therefore, depending on the circumstances, the proposed strategy may not always be the best choice.

4.3 Simbanex

In this section, we give an overview of the visual design of Simbanex, which is implemented as a web-based tool using D3 [24]. One of the main design goals has been to provide a user interface that is conceptually simple regarding the possible interactions and has high responsiveness. The visualization interface consists of four main views: (1) the *Clustering View* (see Figure 4.3 [A]), (2) the *Similarity Network View* (see Figure 4.3 [B]), (3) the *Target-to-All View* (see Figure 4.3 [C]), and (4) the *Similarity Assessment View* (see Figure 4.3 [D]). The first three views are accessible (when populated) by the three tab buttons at the top of

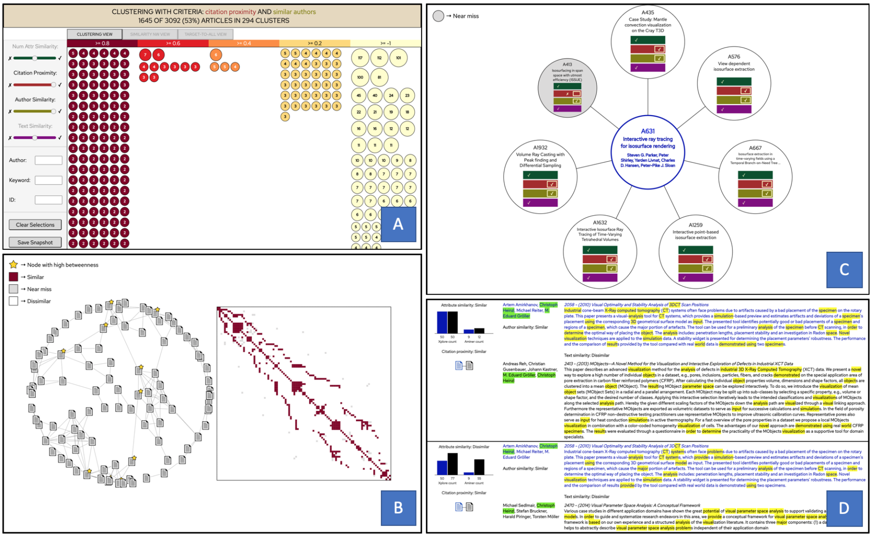

CLUSTERING WITH CRITERIA: citation proximity AND similar authors
1645 OF 3092 (53%) ARTICLES IN 294 CLUSTERS
CLUSTERING VIEW
SIMILARITY NW VIEW
TARGET-TO-ALL VIEW
>= 0.8
>= 0.6
>= 0.4
>= 0.2
>= -1
Num Attr Similarity:
Citation Proximity:
Author Similarity:
Text Similarity:
Author:
Keyword:
ID:
Clear Selections
Save Snapshot
A
→ Node with high betweenness
→ Similar
→ Near miss
→ Dissimilar
B
→ Near miss
A435 Case Study: Mantle convection visualization on the Cray T3D
A413 Isosurfacing in span space with utmost efficiency (ISSUE)
A576 View dependent isosurface extraction
A631 Interactive ray tracing for isosurface rendering
Steven G. Parker, Peter Shirley, Yarden Livnat, Charles D. Hansen, Peter-Pike J. Sloan
A1932 Volume Ray Casting with Peak finding and Differential Sampling
A667 Isosurface extraction in time-varying fields using a Temporal Branch-on-Need Tree ...
A1632 Interactive Isosurface Ray Tracing of Time-Varying Tetrahedral Volumes
A1259 Interactive point-based isosurface extraction
C
Attribute similarity: Similar
Xplore count
Aminer count
Author similarity: Similar
Citation proximity: Similar
Text similarity: Dissimilar
Attribute similarity: Dissimilar
D

Figure 4.3: *The user interface of Simbanex, a Visual Analytics tool for interactive similarity-based exploration of a large set of scientific publications. In the Clustering View [A], the result of clustering with the current similarity criteria is displayed. In the Intra Cluster View [B], the similarity network and the adjacency matrix of a selected cluster can be assessed. The Target-to-all View [C] shows an overview of the matches and near misses for a selected article. Finally, the detailed pairwise comparisons can be assessed in the Similarity Assessment View [D].*

the application, and the *Similarity Assessment View* is displayed in combination with the *Target-to-All View*. The design seeks to reuse already well-proven visual metaphors (such as circles for clusters, word-highlighting for text similarity, and node-link diagrams and matrix representations for networks), and it also provides a custom design for the target-to-all comparisons. Furthermore, to facilitate for the user, the visualization continuously provides a textual explanation of the current settings and the current results in the header banner (see further Figures 4.3 [A], 4.6, and 4.10).

On the conceptual side, we introduce the abstract metaphors of *similarity distribution* and *similarity patterns* as mental models for thinking about what happens when a combination of similarity criteria is executed over a set of items. For instance, if we search for pairs with text similarity and citation proximity we get a different clustering result than if we search for pairs with text similarity and author dissimilarity (see Section 4.3.1). This could be thought of as the two different criteria specifications having different distributions over the set, in the sense that they each reveal the set of item pairs for which its criteria hold true. Furthermore, we can think of the individual items of the set as being chained together by similarity links that form different patterns depending on the set of active criteria (see Section 4.3.2). We argue that exploring these types of patterns can give important insights to the underlying data, and Simbanex is therefore designed to allow the user to assess them at different levels of detail. We will discuss this in more detail in the following subsections.

4.3.1 The Clustering View

When the visualization is loaded, the articles are represented as unclustered article icons in the *Clustering View* (see Figure 4.3 [A]). There are four different similarity criteria to use (*Numeric Attribute Similarity*, *Citation Proximity*, *Author Similarity*, and *Text Similarity*), and the user may select yes/no/unactive for each individual criterion. In accordance with the pre-calculations described in Section 4.2, setting a slider to YES means *"Find all pairs that have been classified as similar for this aspect"*, setting a slider to NO means *"Find all pairs that have been classified as dissimilar for this aspect"*, and setting the slider to the middle, inactive position means *"Do not use this aspect for filtering purposes"*. The user may dynamically select any desired combination to be executed over the data set, and the system will cluster and display all article pairs (if any) that meet the specification. In the terms of our previously introduced terminology, the clustering result is the top-level similarity pattern, and it reveals the similarity distribution of the selected criteria combination over the data set. Clusters are represented as circles where size encodes the number of articles in each cluster, and where spatial position and color both encode the average pairwise similarity score within the cluster. Any changes to the settings results in an animated sequence where each article is clustered together with the articles that it is similar to (if any) given the current settings. It is important to note that this clustering method does not necessarily give a cluster where all items are similar to all other items within the cluster. The reason for this is that item X may be similar to item Y (which puts them both into the same cluster) and item Y may be similar to item Z (which puts Z into the same cluster as Y and X) even though item X and item Z are dissimilar. Therefore, since a similarity relation is not necessarily always transitive, all members of a cluster will not always be similar to all others, but all members will always be connected to each other by at least one path. Finally, since the concept of clustering on only dissimilarity is somewhat counter-intuitive the system will handle such cases by filtering and not by clustering. Thus, if all activated sliders are in the NO position, the system will display the icons of the publications that are dissimilar to all others with regards to the activated aspects, but it will not put them into one common cluster.

4.3.2 The Similarity Network View

Clicking a cluster displays the *Similarity Network View* (see Figures 4.3 [B] and 4.4) where the similarity links between the articles are displayed both in a node-link diagram and in an adjacency matrix. This view allows for analysis of the similarity pattern of the current cluster which, with regard to our introduced terminology, is an intermediate level pattern. As previously discussed, depending on the selected criteria combination different intra-cluster pattern may occur for the same set of items. For example, when using one set of criteria we might get the pattern *"X and Y are similar AND Y and Z are similar BUT X and Z are dissimilar"*, and when using another set of criteria we might get the pattern *"X and Y and Z are all similar"*. Thus, the pattern reveals information on the transitive property of the selected criteria as well as on the overall pairwise homogeneity/heterogeneity within the cluster. This in turn allows for even more nuanced similarity analysis since two items that are not similar when directly compared may still be connected by an indirect "similarity-path" and may therefore still be similar in some sense. Furthermore, the network pattern/topology can also be used to find items which act as bridges between groups of items with higher inter-connectivity. Since such items can be important to locate and analyze further, Simbanex highlights nodes with high betweenness centrality with a golden star. When the user hovers an article node, the application highlights its similarity matches and any near misses as well as the corresponding row/column of the adjacency matrix.

4.3.3 The Target-to-all View

Clicking an article icon in the *Similarity Network View* displays the *Target-to-All View* (see Figures 4.3 [C] and 4.5) which supports the understanding of how similar the selected target article is to any of the other articles in the data set, given the current criteria settings. The similar items and any near misses are displayed in a radial layout which aims to provide an efficient at-a-glance overview of the pairwise comparison of each aspect. By hovering a comparison node, the user can display a word cloud containing any co-occurring authors and any co-occurring words.

4.3.4 The Similarity Assessment View

The *Similarity Assessment View* (see Figures 4.3 [D] and 4.8) is displayed just below the *Target-to-All View* and shows the full details of all of the pairwise comparisons. To facilitate the assessment, the data for the selected target article is color-coded in blue, and co-occurrences of words and authors are highlighted with colored spans. Furthermore, all four system-generated aspect classifications are displayed so that the user can assess them in the direct context of the actual data.

4.3.5 Tracking

To support more specific search and analyses, the user may track articles on author name and/or keyword in order to filter the results to show matches

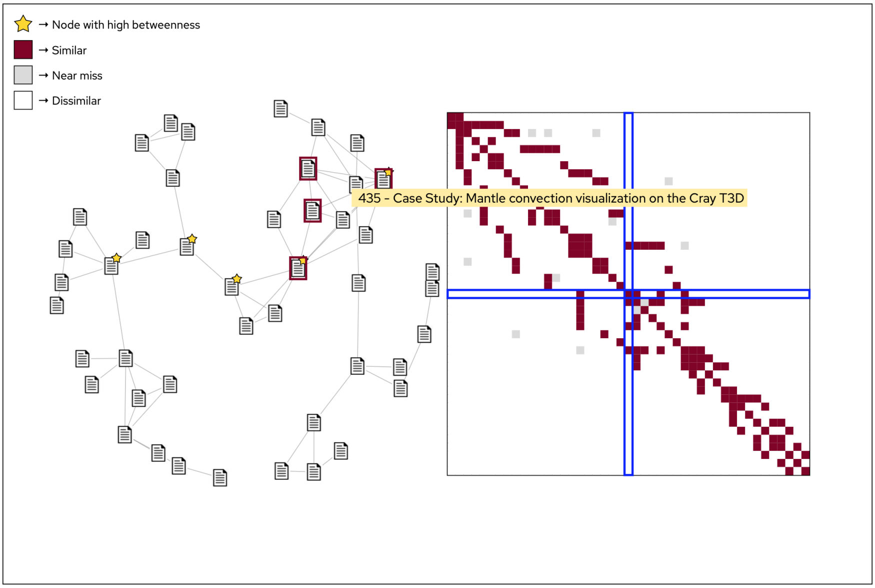

→ Node with high betweenness
→ Similar
→ Near miss
→ Dissimilar
435 - Case Study: Mantle convection visualization on the Cray T3D

Figure 4.4: *The Similarity Network View. Clicking a cluster circle displays the similarity network of the cluster. As can be seen, similarity is not necessarily a transitive relation, so while the items are all connected by at least one path they are still not necessarily all similar to each other. The network pattern/topology reveals the transitive properties of the selected similarity criteria over this specific subset of the data and can be used to establish an indirect similarity-path between two objects that are not similar when directly compared. In this example, the user is hovering the mouse cursor over an article icon to highlight similarity matches and near misses as well as the nodes position in the adjacency matrix.*

only relevant to the current selection. Since matches may still include articles outside of the selection, the tracked articles will be highlighted in green-colored frames throughout the views. In the *Clustering View*, the color coding of the cluster will now indicate the fraction of tracked articles within the cluster (the darker the green the higher the fraction). Spatial position will still encode the average pairwise similarity score within the cluster (see Figure 4.10). The tracking feature is helpful for answering questions such as if articles that mention certain keywords also show high overall similarity.

Finally, we want to point out the fact that the current implementation of Simbanex takes advantage of the fact that similarity is relatively scarce within the data set with regards to the current aspects (i.e., most of the article pairs are dissimilar for most of the aspects). In a scenario where similarity would be more common, we would have less of a filtering effect since the comparisons would yield more matches and larger clusters, which in turn would lead to a less responsive user interface.

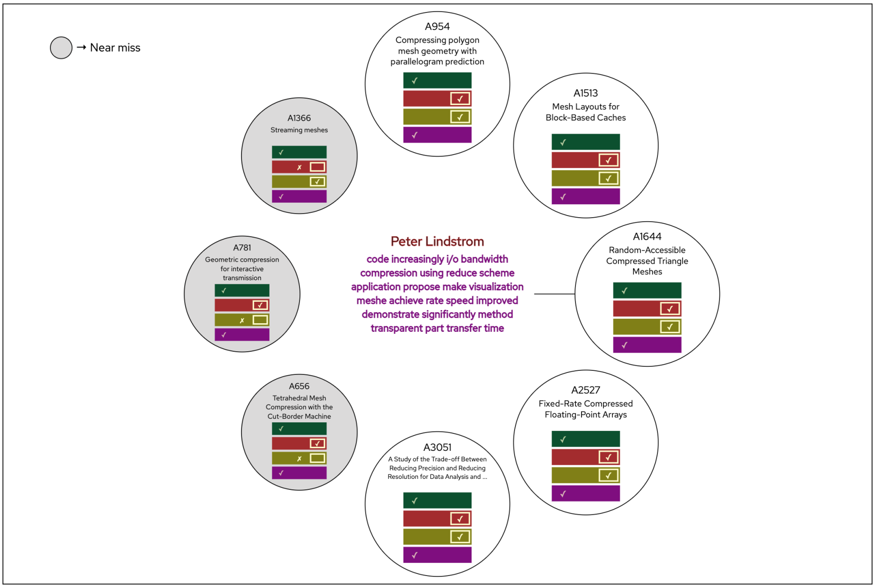

→ Near miss
A954
Compressing polygon mesh geometry with parallelogram prediction
A1513
Mesh Layouts for Block-Based Caches
A1366
Streaming meshes
A781
Geometric compression for interactive transmission
Peter Lindstrom
code increasingly i/o bandwidth compression using reduce scheme application propose make visualization meshe achieve rate speed improved demonstrate significantly method transparent part transfer time
A1644
Random-Accessible Compressed Triangle Meshes
A656
Tetrahedral Mesh Compression with the Cut-Border Machine
A2527
Fixed-Rate Compressed Floating-Point Arrays
A3051
A Study of the Trade-off Between Reducing Precision and Reducing Resolution for Data Analysis and ...

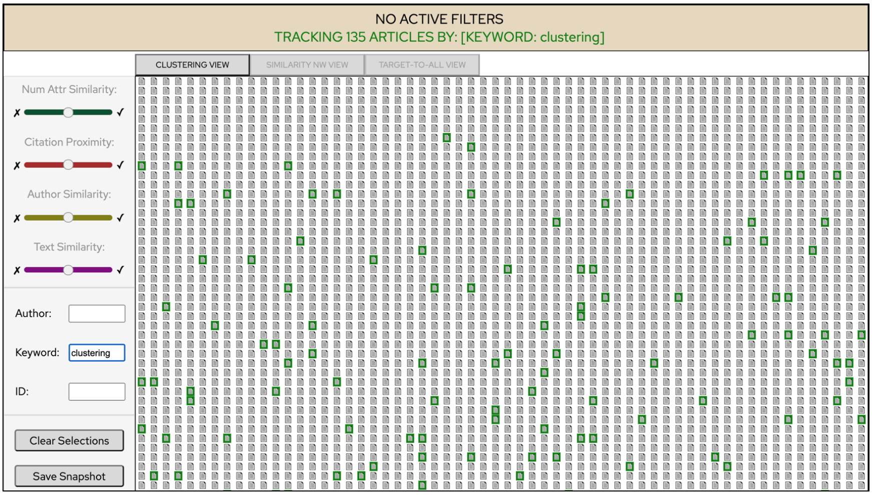
Figure 4.5: *Clicking an article icon in a similarity network view displays the radial Target-to-All View. The design of this view is intended to facilitate an at-a-glance assessment of all matches in the data set for the selected target. The colored charts indicate the current setting of activated sliders (the small white frames within the colored areas) and marks indicate whether a full match, or a near miss, was achieved. For non-activated sliders (no white frame present in the colored area) an indication is given for the setting that would result in a match for the corresponding aspect. In this case, the user has put the Citation Proximity slider (brown) and Author Similarity slider (olive) to YES and has initiated a target-to-all comparison which has resulted in 5 matches and 3 near misses. The user is hovering the node of the comparison with article A1644 to display a word cloud of co-occurring authors and co-occurring words.*

4.4 Use Cases

In this section, we outline two different use cases that highlight some of the strengths of the similarity-based approach. These two use cases have been selected to showcase how the methodology could be used as a part of realistic applications in bibliometrics and scientometrics [29, 90, 103].

4.4.1 Use Case 1 – Citation Link Analysis

Simbanex makes it easy to interactively explore and get a better understanding of some of the citation patterns within the set of publications. The similarity-based approach makes it possible to distinguish between citations between publications with similar abstracts and citations between publications with dissimilar abstracts, and this can be exploited for different tasks.



NO ACTIVE FILTERS

TRACKING 135 ARTICLES BY: [KEYWORD: clustering]

Author:

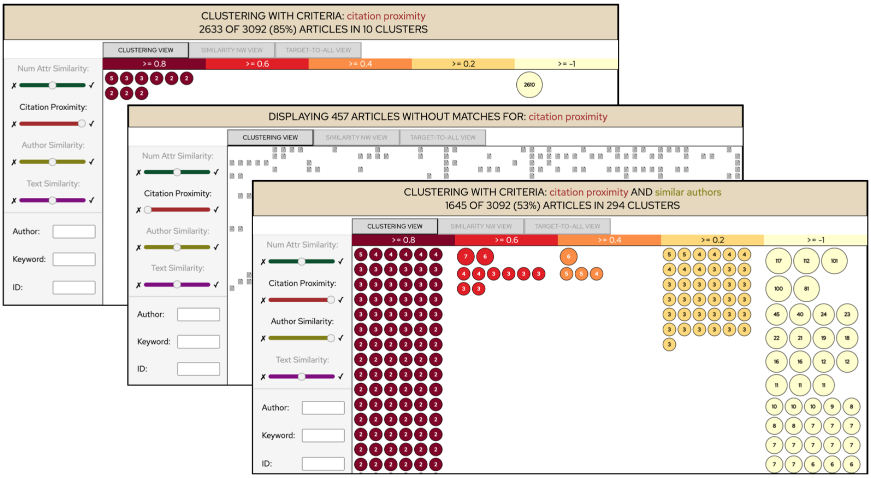
Keyword: clustering

ID:

Figure 4.6: *A search for the keyword* clustering *results in a total of 135 publications. When in tracking mode, Simbanex will highlight tracked articles with green frames throughout the views (see Section 4.4.2 and Figure 4.10).*

1. Starting with the simple case of determining the level of intra-set citations the user puts the Citation Proximity slider to YES and can quickly assess that there are intra-set citation links concerning roughly 85% of all publications, see Figure 4.7 [background].

2. Switching the Citation Proximity slider to NO makes it possible to assess the other 15% of the publications that do not cite publications within the data set (a so-called *outgoing* citation link) and are not cited by any other publication within the data set (a so-called *incoming* citation link), see Figure 4.7 [middle]. By browsing the abstracts, the user can quickly discover that an unproportionally large amount of these cases are from early years with regard to the time span of the data set. The user concludes that this is to be expected since this means that these publications would have less previous articles to cite within the data set to cite which will substantially lower their probability for having an outgoing citation link. Interestingly enough, very few publications from later years of the time span are found within the subset although the reverse effect (i.e., a lower probability for incoming citation links), would be expected for these articles. The user therefore concludes that citing within the data set is a trend that has grown stronger over the years and that it is very common for recent publications to do so.



CLUSTERING WITH CRITERIA: citation proximity
2633 OF 3092 (85%) ARTICLES IN 10 CLUSTERS

DISPLAYING 457 ARTICLES WITHOUT MATCHES FOR: citation proximity

CLUSTERING WITH CRITERIA: citation proximity AND similar authors
1645 OF 3092 (53%) ARTICLES IN 294 CLUSTERS

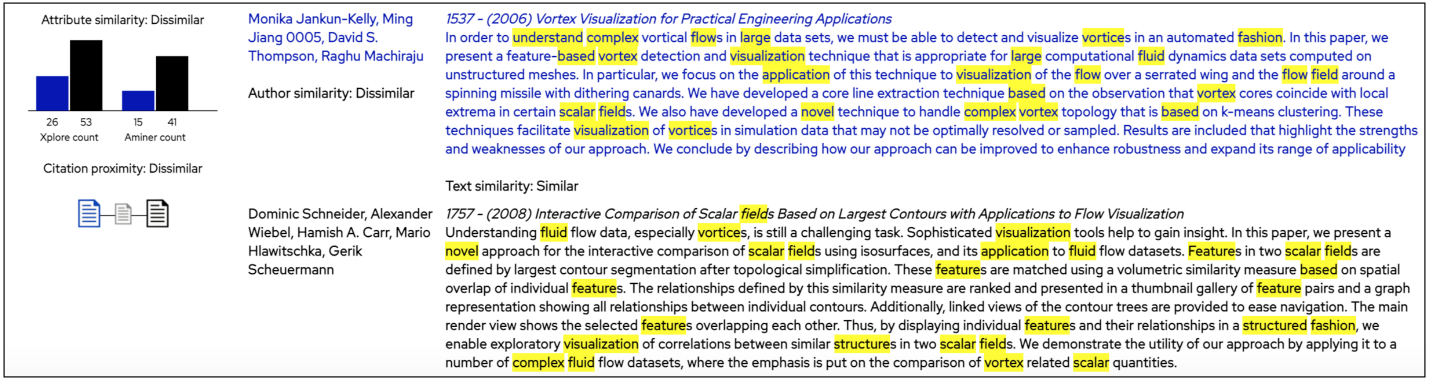
Figure 4.7: In the first three steps of Use Case 1 (in order from the background to the fore), the user explores the level of intra-set citation and the level of self-citation.

3. To assess the level of self-citation, the user now sets both the Citation Proximity and the Author Similarity sliders to YES and concludes that the self-citation amounts to about 53%, see Figure 4.7 [foreground].

4. The user then decides to explore whether any potentially missing citation links between similar publications can be found, and therefore sets the Citation Proximity slider to NO and the Text Similarity slider to YES. This reveals that there are 11 article pairs with high text similarity and no citation link. Eight of these pairs have high pairwise author similarity and three pairs have low pairwise author similarity.

5. The user clicks on a cluster of the three with no author similarity to display the similarity network, which in this case is trivial. Clicking an article node displays the the similarity details so that an assessment can be made of whether the match qualifies as a possible citation that should have been made or not, see Figure 4.8.

6. The user plans to make a submission to an upcoming conference and does not want to miss to cite any previous articles with similar content. He therefore embeds his proposed abstract with USE (as specified in Section 4.2), saves the results, and puts the files into a specified directory of the application. Then, he selects the *Upload Abstract* button to display articles with high semantic text similarity (if any) to make an assessment of whether they are relevant candidates for outgoing citations or not, see Figure 4.9.



Attribute similarity: Dissimilar

Citation proximity: Dissimilar

Monika Jankun-Kelly, Ming Jiang 0005, David S. Thompson, Raghu Machiraju

Author similarity: Dissimilar

1537 - (2006) Vortex Visualization for Practical Engineering Applications

In order to understand complex vortical flows in large data sets, we must be able to detect and visualize vortices in an automated fashion. In this paper, we present a feature-based vortex detection and visualization technique that is appropriate for large computational fluid dynamics data sets computed on unstructured meshes. In particular, we focus on the application of this technique to visualization of the flow over a serrated wing and the flow field around a spinning missile with dithering canards. We have developed a core line extraction technique based on the observation that vortex cores coincide with local extrema in certain scalar fields. We also have developed a novel technique to handle complex vortex topology that is based on k-means clustering. These techniques facilitate visualization of vortices in simulation data that may not be optimally resolved or sampled. Results are included that highlight the strengths and weaknesses of our approach. We conclude by describing how our approach can be improved to enhance robustness and expand its range of applicability

Text similarity: Similar

Dominic Schneider, Alexander Wiebel, Hamish A. Carr, Mario Hlawitschka, Gerik Scheuermann

1757 - (2008) Interactive Comparison of Scalar fields Based on Largest Contours with Applications to Flow Visualization

Understanding fluid flow data, especially vortices, is still a challenging task. Sophisticated visualization tools help to gain insight. In this paper, we present a novel approach for the interactive comparison of scalar fields using isosurfaces, and its application to fluid flow datasets. Features in two scalar fields are defined by largest contour segmentation after topological simplification. These features are matched using a volumetric similarity measure based on spatial overlap of individual features. The relationships defined by this similarity measure are ranked and presented in a thumbnail gallery of feature pairs and a graph representation showing all relationships between individual contours. Additionally, linked views of the contour trees are provided to ease navigation. The main render view shows the selected features overlapping each other. Thus, by displaying individual features and their relationships in a structured fashion, we enable exploratory visualization of correlations between similar structures in two scalar fields. We demonstrate the utility of our approach by applying it to a number of complex fluid flow datasets, where the emphasis is put on the comparison of vortex related scalar quantities.

Figure 4.8: *One of the suggestions for "missing citations" from Use Case 1. As can be seen there already is an indirect (1–hop) link between the two publications, and maybe it could have been relevant to have a direct citation link instead.*

4.4.2 Use Case 2 – Topic Similarity

For the second use case, we will use Simbanex to locate topic clusters from selected keywords:

1. In this use case, another user has a special interest in finding out if there are any specific sub groupings within the set of publications for which clustering is an important topic. He therefore enters the keyword "clustering" into the keyword search field and gets a result of a total of 135 articles (which are now highlighted with green frames).

2. Since it is still not an easy task to assess whether these 135 articles form smaller topic clusters or not (within the larger scope of clustering), the user sets the Text Similarity slider to YES. The system clusters the publications and this time also filters the result so that only clusters containing at least one tracked item remains visible.

3. From the intensity of the green color of the clusters, the user concludes that there is one cluster containing a high fraction of tracked articles, although the average pairwise similarity score indicates lower intra-cluster similarity, which lowers the probability for a sub-topic cluster, see Figure 4.10 [background].

4. To further investigate, the user clicks the cluster to display the similarity network and notes that one article acts like a bridge between 3 of the others, which is a pattern that still makes a topic cluster possible, see Figure 4.10 [foreground].

5. By clicking the article nodes of the similarity network and assessing the abstract texts, the user can conclude that the articles indeed form a cluster on the sub topic of "visual cluster analysis". Furthermore, for some of the articles there are out-of-cluster near misses which are relevant, so the article set can be expanded further.

6. By noting that the discovered subset of articles would not have been easily selected only by a combination of keywords, the user concludes that there are cases when a similarity-based approach can be used for topic detection.

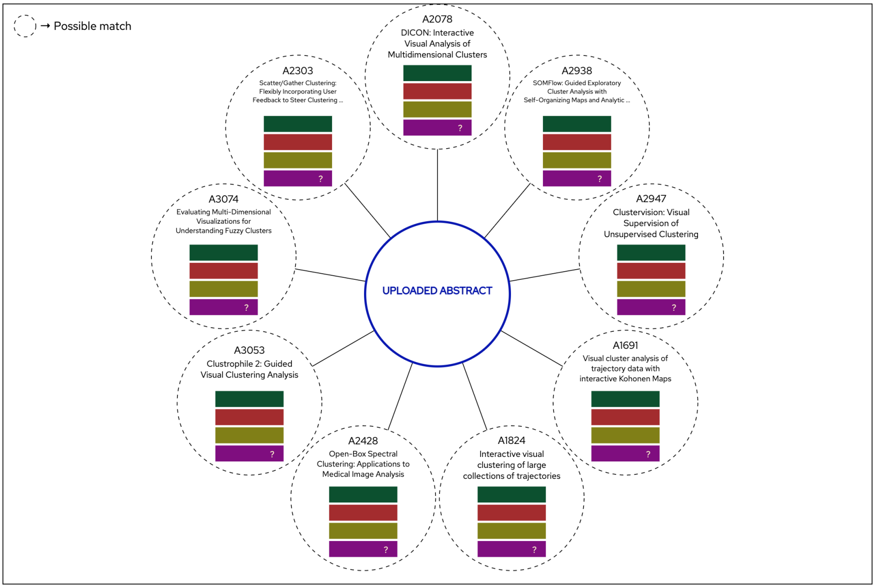
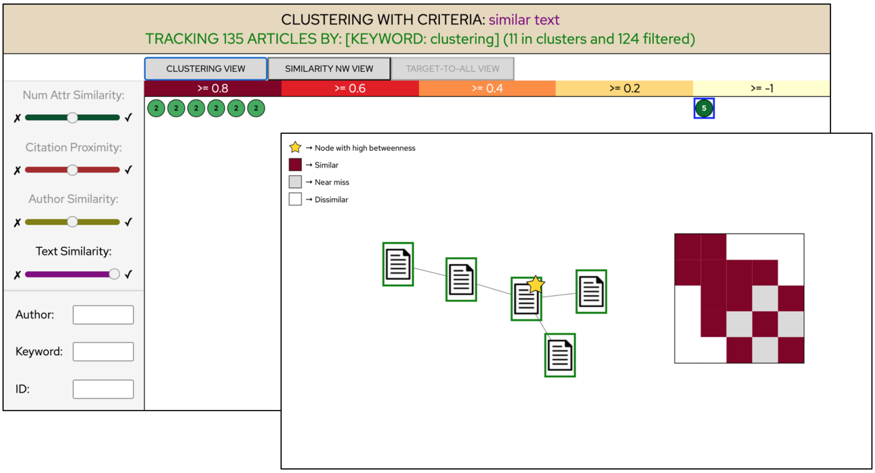

→ Possible match
A2078
DICON: Interactive Visual Analysis of Multidimensional Clusters
A2303
Scatter/Gather Clustering: Flexibly Incorporating User Feedback to Steer Clustering ...
A2938
SOMFlow: Guided Exploratory Cluster Analysis with Self-Organizing Maps and Analytic ...
A3074
Evaluating Multi-Dimensional Visualizations for Understanding Fuzzy Clusters
A2947
Clustervision: Visual Supervision of Unsupervised Clustering
UPLOADED ABSTRACT
A3053
Clustrophile 2: Guided Visual Clustering Analysis
A1691
Visual cluster analysis of trajectory data with interactive Kohonen Maps
A2428
Open-Box Spectral Clustering: Applications to Medical Image Analysis
A1824
Interactive visual clustering of large collections of trajectories
?

Figure 4.9: *When an abstract is uploaded, the system displays possible matches (if any) based on semantic similarity. The user can then assess each suggestion individually to verify if it makes for relevant citations or not.*


CLUSTERING WITH CRITERIA: similar text

TRACKING 135 ARTICLES BY: [KEYWORD: clustering] (11 in clusters and 124 filtered)

Figure 4.10: *When tracking articles on keywords and/or author names the color intensity of the clusters indicate the fraction of tracked items within the cluster (the darker the green the higher the fraction). Clusters that do not contain any tracked articles are filtered. The* Similarity Network View *for the selected cluster reveals that all contained articles are not similar to each other. However, further assessment of the abstract texts shows that they still form a topic cluster on visual cluster analysis (see Section 4.4.2).*

Arriving at the end of this chapter we can note that we have now presented the two key concepts of our proposed framework: (1) multiple embedding similarity calculations (Chapter 3), and (2) the aspect-driven approach (Chapter 4). We have discussed them both separately and also showed how they can be used in combination to achieve both high quality and high flexibility in similarity-based MVN analysis. In the next chapter, we will conclude this thesis by discussing the most important results and relating them back to the research goals set out in Section 1.2.

Chapter 5

Discussions and Conclusions

Contents

In this thesis we have explored a novel approach to MVN embedding and presented a VA methodology which introduces the concept of using, and combining, several different embeddings for the same underlying data. We have shown that deploying several different embeddings can allow for higher quality and flexibility in similarity calculations than when using only a single embedding. We have also shown the strengths and possibilities of a similarity-based approach for MVN analysis. Our proposed strategy is generalizable beyond the scope of MVNs, since it can be applied to any complex data type that may be broken down into separately embeddable aspects. Furthermore, we have presented different visualizations which showcase our proposed methodology for real-world scenarios within the field of bibliometrics and scientometrics. We would also like to remind the reader, once again, that it has not been our intention to give the impression that our approach is relevant for all problems within these fields. Nevertheless, we believe that our proposed methodology and applications give a relevant contribution, and that they provide a novel approach which could hopefully be relevant to problems also beyond the scope of the ones that we have discussed. In this final chapter, we will first discuss our results in the context of the three research goals that we identified in Chapter 1 followed by some general remarks and "pearls of wisdom". Finally, we will discuss an alternative approach for combining embeddings, and why we chose not to use it, as well as outline possible directions for future work.

5.1 Fulfillment of Research Goal 1

As stated by research goal 1, *G1*, our framework needs a method for combining several different embeddings (of the same underlying data) in order to augment the quality of the similarity calculations. The method which we have introduced in Chapter 3 fulfils this goal since it provides a methodology for obtaining several different embeddings for the same underlying data and for finding ensemble combinations which perform better similarity calculations than any of the single embeddings taken on their own, see Figure 5.1 and 5.2. As we have shown, it is normally possible to obtain different embedding vectors for any embeddable data item by either using different embedding algorithms or by varying the hyperparameter settings for the same algorithm. The process also seamlessly handles embedding types of different dimensionality since it operates on the level of the calculated similarity scores (which are always calculated within a single embedding type). Furthermore, the proposed EEVO tool for the optimization of ensemble configurations is conceptually simple and lets the user stay in control of the demarcation line between similar and dissimilar. As previously stated, we believe this to be important since similarity is an elusive concept which is not easy to capture by purely computational means. Even though there is no guarantee that the search for an optimal ensemble will be successful for all cases, our results clearly indicate that an ensemble is often a better choice than only a single embedding. These results are in line with the results from ensemble methods for supervised classification, where it has been shown that a well-chosen ensemble combination is often a better choice than any of the classifiers taken by its own. In other words: combining the efforts of several classifiers could hold the potential for achieving higher quality, but it is not always possible to unlock that potential. Interestingly enough, and much in line with ensemble learning for supervised classification, it seems that even an embedding which performs poorly on its own can give a valuable contribution when combined with others. Therefore, it is in general not possible to predict ensemble performance based on the individual performances only.

We want to conclude this section by pointing out that our methodology does not rule out the possibility that there may exist an embedding technology that can achieve high enough quality on its own—we are instead suggesting an alternative strategy to use for cases when no such candidate can be identified.

5.2 Fulfillment of Research Goal 2

Research goal 2, *G2*, states that our proposed methodology should provide ways to reuse, and leverage, already existing embedding technologies. Using the fact that several embedding technologies already exist for data types that are common building blocks for MVNs (e.g., network topology, word/text and categorical data)

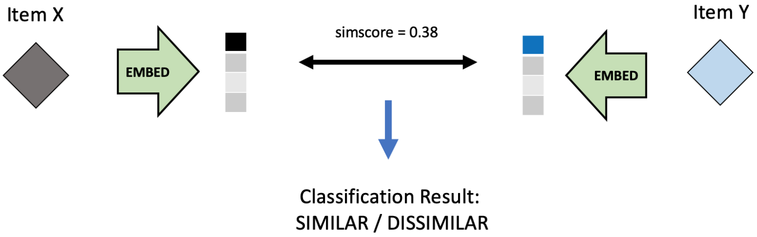


Figure 5.1: *With a single embedding of each item, one similarity score can be calculated and compared to a threshold value to arrive at a decision if the pair is similar or dissimilar.*

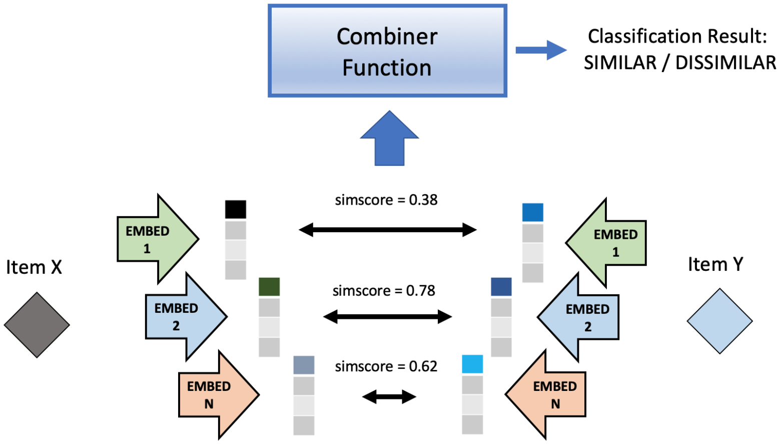


Figure 5.2: *With several embeddings of each item, several similarity scores can be calculated and fed into a combiner function to arrive at a decision if the pair is similar or dissimilar. This figure is identical to Figure 3.2 and is replicated here to facilitate the comparison to Figure 5.1.*

and by introducing the aspect-driven approach, we have shown examples of how a complex entity, such as an MVN, can be divided into separately embeddable aspects. The different aspects may then be separately embedded by any already existing embedding algorithm for that specific data type. The benefit of this approach is that we obtain a flexible vector representation of the underlying data which can then be used for a multifaceted analysis where single aspects may be included or excluded depending on the needs. This in turn opens for more subtle similarity analysis than just binary similar/dissimilar since scenarios like *"similar with regards to N out of M aspects"* may be identified and further handled. Furthermore, as we have demonstrated, the methodology developed to fulfil research goal 1 contributes to the fulfillment of this research goal as well since it can be used within the context of each aspect (i.e., for each aspect several different embedding algorithms can be used). An example of this is the case of network node topology embedding in Section 3.4.1, which illustrates a situation where three already existing node embedding algorithms are used and the individual yields of their similarity calculations are combined. Hence, we see that our proposed methodology is well suited for reusing and leveraging already existing embedding technologies. The main advantage of this approach is that we obtain a flexible framework that can be applied to several different scenarios, without the need of developing new specific embedding algorithms. As we have shown, the strategy of dividing a complex entity, like a MVN, into more limited aspects drastically increases the possible choices of embedding algorithms.

5.3 Fulfillment of Research Goals 3 and 4

Research goals 3 and 4, *G3* and *G4*, are tightly connected since they state that we should develop new solutions for visualizing similarity-based aspects of MVNs and also show how this approach can be of value for the analysis. The fulfillment of these two goals is mainly achieved by the implementation our prototype VA tools, Simbanex and Simbatex, by which we showcase the potential of similarity-based analysis within the scope of MVNs generated from a large set of scientific publications.

Although our selected use cases mainly fall within fields such as Scientometric, Bibliometric and Science Mapping, the task of searching for *"something similar to what I have already found"* generalizes to many fields as a common starting point for analysis, and hence, it can be relevant for very different scenarios in many different fields. In our proposed applications, we show that the similarity-based approach can be used for: (1) obtaining a better general understanding of the given data set (for instance, the number of items pairs that fulfill a specific set of similarity criteria), and (2) for searching for items that are similar to a selected target (for instance, when recommending possible citation suggestions for an uploaded article abstract). These two tasks are the principle building blocks of a similarity-based analysis and to develop a richer terminology for the mental model we introduced the abstract concept of *similarity patterns* as a generic term for the yielded result. As we have seen, such similarity patterns may occur at different level of abstraction and come in the form of clustering results (showing how the set is partitioned by the selected criteria), similarity networks (showing how specific items are connected by similarity links) or as the details of the pairwise comparisons (showing exactly how similar/dissimilar two selected items are). In our tools, we have shown several different scenarios for how to visualize and exploit these patterns in order to obtain important insights to the underlying MVN, see Figure 5.3. A key step for this is to to unlock the pattern-recognition capabilities of the human analyst by the specially developed visualization techniques which are implemented in all our proposed

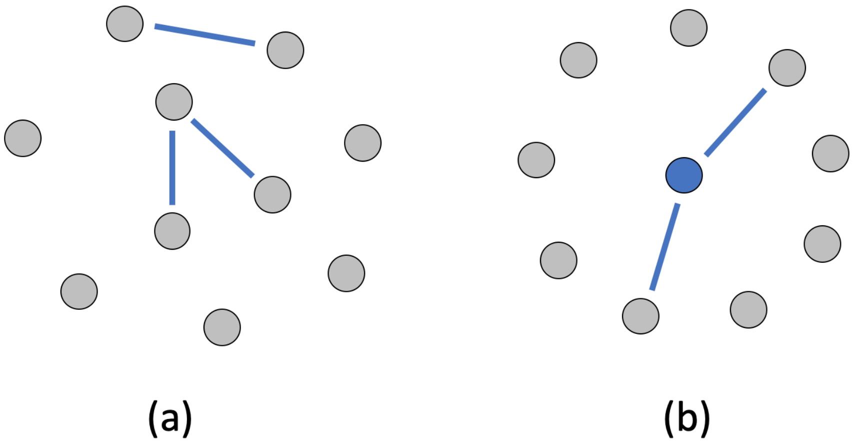


Figure 5.3: *The two main scenarios of similarity-based analysis: (a) assessing the number of pairs that are similar given a set of criteria, and (b) a target-to-all search to find items which are similar to a selected target. In both these examples, similarity is indicated by a link between the items (i.e., items connected by a link are similar to each other).*

visualizations. In this way the purely computational steps may be used to quickly direct the analyst to the parts of the MVN where interesting similarity patterns occur, and the analyst may then be in control of the subtler parts of the analysis on the greyscale between similar and dissimilar. As we have shown with the use cases for our tools, this human-in-the-loop approach allows for advanced exploration scenarios of the underlying MVN.

5.4 General Remarks and Insights

As we have seen in Chapters 3 and 4, our proposed strategy is based on two different conceptual levels of embedding combination. Ideally, the targeted MVN would first be partitioned into several different aspects, and then each aspect would be embedded by several different embedding techniques. This two-level process yields a highly flexible vector representation of the MVN which can be used for a variation of analysis scenarios. As we have shown, the multiple embeddings for each aspect can be used for augmenting the quality of the similarity calculations, and then a similarity criterion can be mapped to each individual aspect, see Figure 5.4. This in turn allows for high flexibility when searching since each aspect criterion can be activated, deactivated or even negated depending on the search interest. This fine-tuning ability is important since, for many real-world scenarios, we can find item pairs along a floating scale of "somewhat similar" (i.e., the items are similar to some extent for some of the

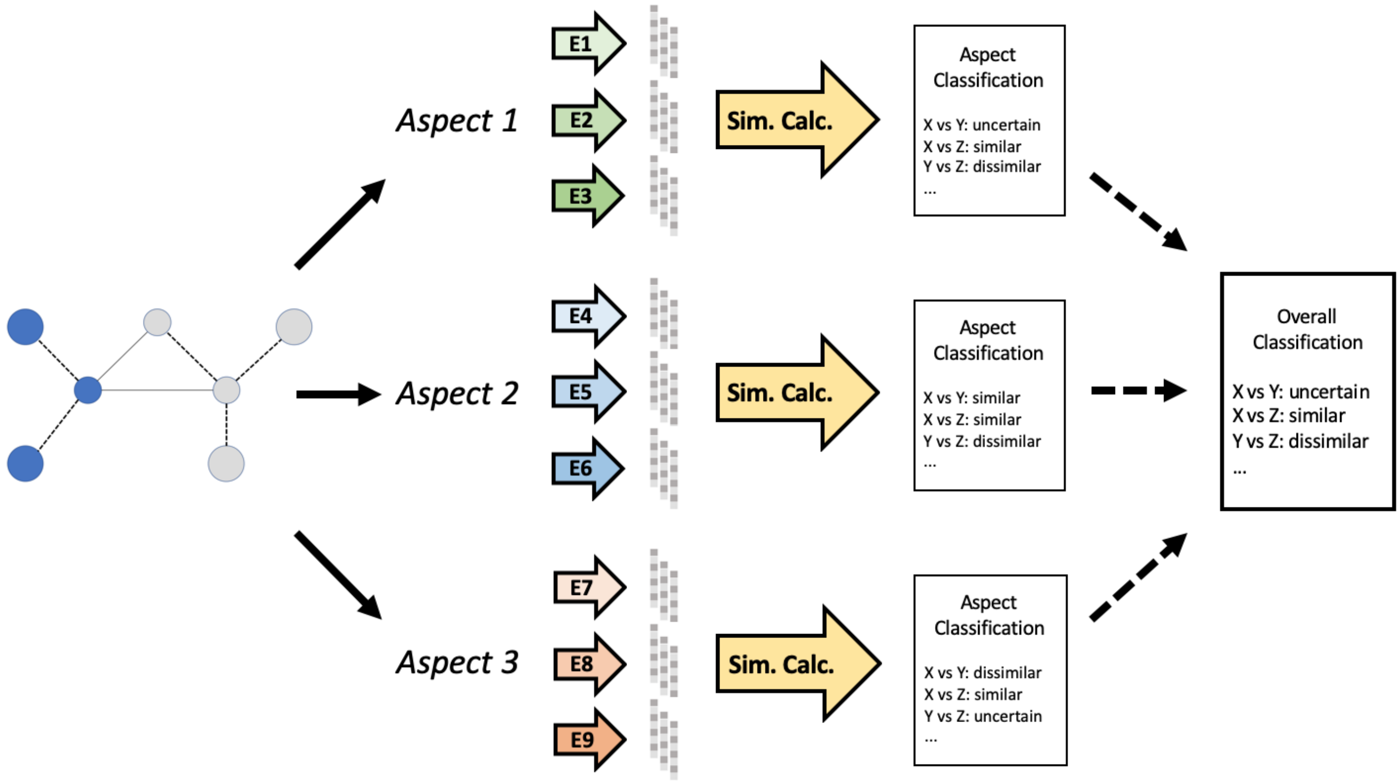


Figure 5.4: *A schematic view of the process. An MVN is split into different aspects which are then separately embedded using multiple embedding algorithms. Similarity calculations are performed within the scope of each aspect and from this an aspect classification can be obtained. In accordance with the current criteria settings the different aspect classifications are then combined to obtain an overall classification.*

aspects) between the pairs that are “indeed similar” and “indeed dissimilar”. Therefore, being able to analyze this greyscale range can provide better insight to the similarity relation between any two items. An example of this would be comparing individuals using the five different criteria height, weight, age, gender, and eye color. Intuitively, if person A and person B are similar on 4 out of these 5 criteria many people would probably regard them as “more similar” than C and D who are similar only on 2 of the criteria. Since there is usually no unambiguous demarcation line for where similarity begins or where it ends, it is a challenge to find the right trade-off for the computational process so that the classification of similar, dissimilar and “somewhere in between” corresponds to the user expectation. Classifying too many item pairs as similar distorts the analysis and leads to lower user trust, while classifying too few pairs limits the possibilities for a correct analysis and leads to less correct insights regarding the data set. To handle this challenge, we have introduced the concept of “near misses” in our proposed applications as a way to alert the user of items which almost fulfill the selected set of similarity criteria. In this way the analyst can choose to assess these items further to see if they are relevant for the current task or not.

5.4.1 Alternative Method

Handling the different aspect embeddings as separate vectors is not the only possible method that could have been used. One alternative way for combining the embeddings, which we have also explored, is to concatenate all the separate embeddings and thus obtain one joint embedding for each item in the data set. We could then use these joint vectors in the similarity calculation to yield a single overall similarity value, see Figure 5.5). With this method, item pairs with similar aspects embeddings will yield a high cosine similarity value for their joint embeddings. The reason for this is that if V1 and V2 yield high cosine similarity and V3 and V4 yield high cosine similarity, then the concatenation of V1 and V3 will yield high cosine similarity with the concatenation of V2 and V4. However, using a joint embedding means that the only hyperparameter for controlling the classification would be the unique threshold value for the single similarity calculation. This would in turn mean that the concept of ensembles and voting schemes would no longer be applicable and that there would be no corresponding way of combining the different aspect embeddings to augment the quality of the similarity calculations. Furthermore, to the best of our knowledge, there is no theory for deducing the cosine similarity value for a pair of joint vectors based on the cosine similarity values of their constituent vector parts. In other words, knowing the cosine similarity value for V1 and V2 and the cosine similarity value for V3 and V4 does not allow us to easily deduce the cosine similarity value for V5 = concatenate(V1, V3) and V6 = concatenate(V2, V4). Therefore, with a similarity calculation using two joint embeddings, it would be much harder to

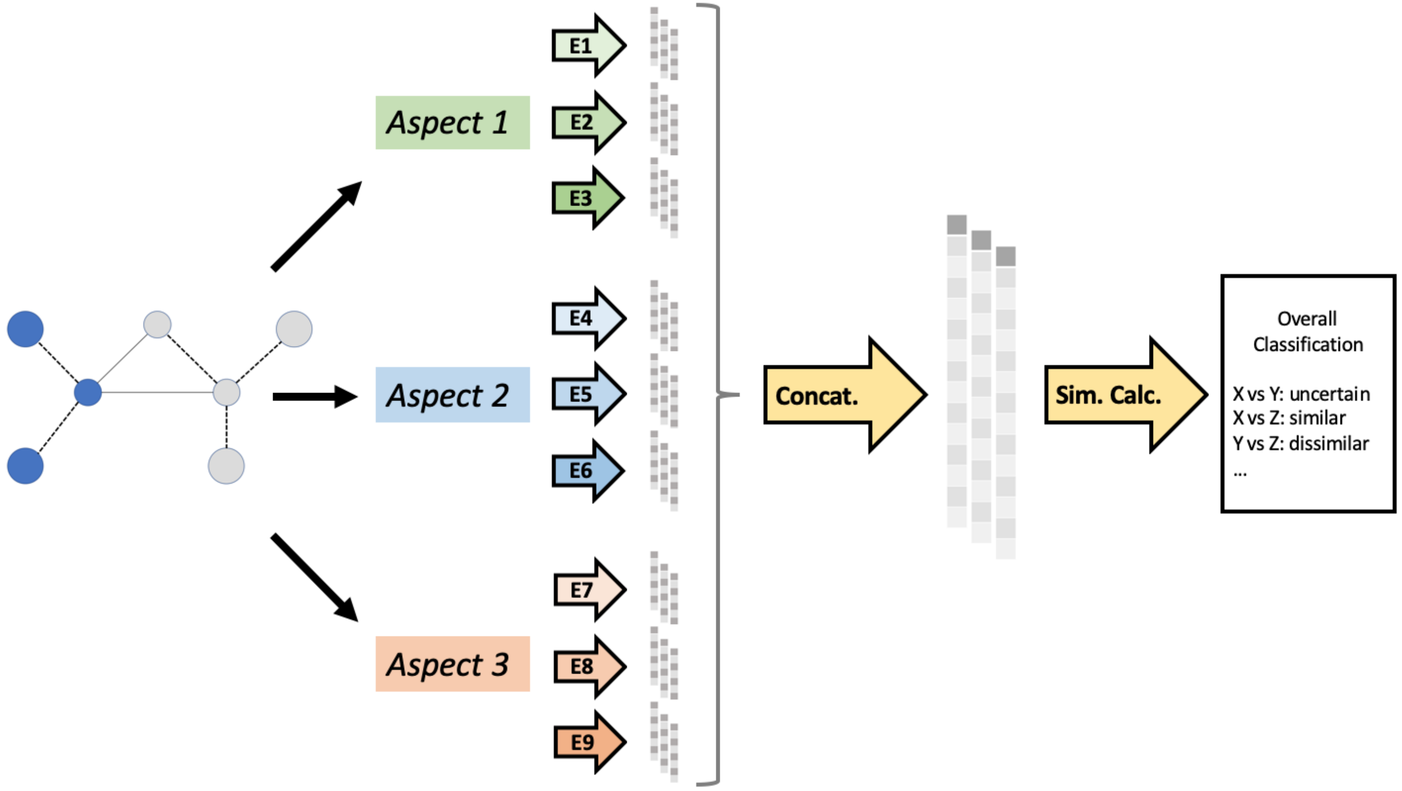


Figure 5.5: *A schematic view of an alternative process for combining multiple embeddings. (Compare to Figure 5.4.) An MVN is split into different aspects which are then separately embedded using multiple embedding algorithms. The embeddings are then concatenated so that one single joint embedding is obtained for each item. Pairwise similarity calculations are then performed, using the joint embeddings, to obtain an overall classification. From the result of the similarity calculations it is not possible to deduce any of the aspect specific similarity values.*

reason about any aspect-specific similarity, and it would not be possible to use the concept of aspect-specific criteria. As a consequence, using the concatenation would result in a more coarse-grained control of the similarity calculations than if we use the different embedding types by themselves—and that is why we have chosen not to use this method in our framework.

5.5 Future Work

In this work, we have mainly focused on combining embeddings for similarity-based analysis of MVNs. Looking ahead, we see several potentially interesting directions for future work which are highlighted in the following.

Multi-embedding clustering As discussed in the introduction of Chapter 1, embeddings are commonly used for clustering. However, most clustering algorithms can not handle several different embeddings for the same data item. Therefore, we see an interesting opportunity in exploring VA solutions for multi-embedding clustering. This would most likely include algorithm-specific adaptation, as well as the development of generic solutions for how to combine partial results to a final, unified outcome. Since this direction would lead to a more general context than that of only MVNs, it will however not be one of our first to explore.

Ensemble optimization As discussed in Chapter 3, the optimization of embedding-ensembles (which is supported by the EEVO tool) is a complex process with many challenges, both regarding computation and visualization. A natural direction for future work would therefore be to explore this process further to find even better ways for optimization, visualization, and interaction. As we have mentioned, one idea would be to explore how to extend the “horizon" of the user guidance in the EEVO tool, and another idea would be to look into the possibilities of more direct computational optimization. This is something that that we are planning to do.

Non-embeddable aspects In this work, we have shown how to make use of embeddable aspects for MVN analysis. However, there could very well exist interesting aspects of an MVN for which no suitable embedding technology exist. Therefore, a direction for future work could be to explore how the methodology framework could be extended to include also these aspects. More specifically, this would mean incorporating the results of similarity calculations that are not embedding-based, and our assessment is that this would be relatively straightforward to do. This is something we will consider to do if such cases present themselves within our ongoing work.

Other data types As discussed in Section 3.2, our proposed methodology is generic in the sense that operates on the level of numeric embedding vectors and makes no assumptions on the underlying data type. Therefore, there is potential for using the same methods on any complex data type that can be broken down to separately embeddable aspects. Our assessment is that this will be straightforward to do since the framework makes no assumption that the underlying data entity must be a MVN. However, since this direction would lead to a more general context than that of only MVNs, it will not be one of our first to explore.

The aspect of time In this work, we have shown how to combine and make use of several common aspects of MVNs. However, we have not covered the aspect of dynamic MVNs that change over time. Therefore, there is interesting potential for future work in exploring what possibilities the multi-embedding approach could bring for VA solutions for dynamic networks. Our assessment is that this direction is the most challenging of the ones that we have presented in this section, and that it is also the hardest to predict in terms of more concrete steps or ideas to start from. However, since this is also one of the directions that we find the most interesting, it is something that we are planning to explore.

Bibliography

[1] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, 6:52138–52160, September 2018. doi:10.1109/ACCESS.2018.2870052.

[2] Charu C. Aggarwal. *Machine Learning for Text*. Springer, 2018.

[3] Felipe Almeida and Geraldo Xexéo. Word embeddings: A survey. *CoRR*, abs/1901.09069, 2019. arXiv:1901.09069.

[4] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35(4):105–120, December 2014. doi:10.1609/aimag.v35i4.2513.

[5] Natalia Andrienko, Tim Lammarsch, Gennady Andrienko, Georg Fuchs, Daniel A. Keim, Silvia Miksch, and Alexander Rind. Viewing visual analytics as model building. *Computer Graphics Forum*, 37(6):275–299, September 2018. doi:10.1111/cgf.13324.

[6] Dustin L. Arendt, Nasheen Nur, Zhuanyi Huang, Gabriel Fair, and Wenwen Dou. Parallel Embeddings: A visualization technique for contrasting learned representations. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, IUI '20, pages 259–274. ACM, 2020. doi:10.1145/3377325.3377514.

[7] Yonatan Belinkov and James Glass. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72, March 2019. URL: https://www.aclweb.org/anthology/Q19-1004, doi:10.1162/tacl_a_00254.

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

[9] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, March 2003.

[10] Mathew Berger, Katherine McDonough, and Lee M. Seversky. cite2vec: Citation-driven document exploration via word embeddings. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):691–700, January 2017. doi:10.1109/TVCG.2016.2598667.

[11] Wolfgang Berger and Harald Piringer. Interactive visual analysis of multiobjective optimizations. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, VAST '10, pages 215–216. IEEE, 2010. doi:10.1109/VAST.2010.5651694.

[12] Wolfgang Berger, Harald Piringer, Peter Filzmoser, and Eduard Gröller. Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. *Computer Graphics Forum*, 30(3):911–920, June 2011. doi:10.1111/j.1467-8659.2011.01940.x.

[13] Angie Boggust, Brandon Carter, and Arvind Satyanarayan. Embedding Comparator: Visualizing differences in global structure and local neighborhoods via small multiples. *CoRR*, abs/1912.04853, 2019. arXiv:1912.04853.

[14] Matthew Brehmer and Tamara Munzner. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385, December 2013. doi:10.1109/TVCG.2013.124.

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

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

[17] Davide Ceneda, Natalia Andrienko, Gennady Andrienko, Theresia Gschwandtner, Silvia Miksch, Nikolaus Piccolotto, Tobias Schreck, Marc Streit, Josef Suschnigg, and Christian Tominski. Guide me in analysis: A framework for guidance designers. *Computer Graphics Forum*, 39(6):269–288, September 2020. doi:10.1111/cgf.14017.

[18] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal Sentence Encoder. *CoRR*, abs/1803.11175, 2018. arXiv:1803.11175.

[19] Angelos Chatzimparmpas, Rafael M. Martins, Ilir Jusufi, Kostiantyn Kucher, Fabrice Rossi, and Andreas Kerren. The state of the art in enhancing trust in machine learning models with the use of visualizations. *Computer Graphics Forum*, 39(3):713–756, June 2020. doi:10.1111/cgf.14034.

[20] Angelos Chatzimparmpas, Rafael M. Martins, Kostiantyn Kucher, and Andreas Kerren. StackGenVis: Alignment of data, algorithms, and models for stacking ensemble learning using performance metrics. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1547–1557, February 2021. doi:10.1109/TVCG.2020.3030352.

[21] Siming Chen, Lijing Lin, and Xiaoru Yuan. Social media visual analytics. *Computer Graphics Forum*, 36(3):563–587, June 2017. doi:10.1111/cgf.13211.

[22] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, November 2011.

[23] Peng Cui, Xiao Wang, Jian Pei, and Wenwu 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.

[24] D3 — Data-driven documents. https://d3js.org/, 2011. Accessed September 21, 2021.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. arXiv:1810.04805.

[26] Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. A survey on ensemble learning. *Frontiers of Computer Science*, 14(2):241–258, 2020. doi:10.1007/s11704-019-8208-z.

[27] Alex Endert, William Ribarsky, Cagatay Turkay, B.L. William Wong, Ian Nabney, Ignacio Díaz Blanco, and Fabrice Rossi. The state of the art in integrating machine learning into visual analytics. *Computer Graphics Forum*, 36(8):458–486, December 2017. doi:10.1111/cgf.13092.

[28] Petri Eskelinen, Kaisa Miettinen, Kathrin Klamroth, and Jussi Hakanen. Pareto navigator for interactive nonlinear multiobjective optimization. *OR Spectrum*, 32(1):211–227, January 2010.

[29] Paolo Federico, Florian Heimerl, Steffen Koch, and Silvia Miksch. A survey on visual approaches for analyzing scientific literature and patents. *IEEE Transactions on Visualization and Computer Graphics*, 23(9):2179–2198, September 2017. doi:10.1109/TVCG.2016.2610422.

[30] Marshall L Fisher. Interactive optimization. *Annals of Operations Research*, 5(3):539–556, October 1985. doi:10.1007/BF02023610.

[31] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *Proc. of the IEEE International Conference on Data Science and Advanced Analytics*, pages 80–89. IEEE, 2018. doi:10.1109/DSAA.2018.00018.

[32] Michael Gleicher, Danielle Albers, Rick Walker, Ilir Jusufi, Charles D. Hansen, and Jonathan C. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, October 2011. doi:10.1177/1473871611416549.

[33] Wael Gomaa and Aly Fahmy. A survey of text similarity approaches. *international journal of Computer Applications*, 68, 04 2013. `doi:10.5120/11638-7118`.

[34] Google Colaboratory. https://colab.research.google.com/, 2014. Accessed September 21, 2021.

[35] Palash Goyal and Emilio 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`.

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

[37] Riccardo Guidotti, Anna Monreale, Franco Turini, Dino Pedreschi, and Fosca Giannotti. A survey of methods for explaining black box models. *ACM Computing Surveys*, 51:93:1–93:42, August 2018. `doi:10.1145/3236009`.

[38] Leonardo Gutiérrez-Gómez and Jean-Charles Delvenne. Unsupervised network embeddings with node identity awareness. *Applied Network Science*, 4(1):82, October 2019. `doi:10.1007/s41109-019-0197-1`.

[39] Jussi Hakanen, Kaisa Miettinen, and Krečimir Matković. Task-based visual analytics for interactive multiobjective optimization. *Journal of the Operational Research Society*, 2020. `doi:10.1080/01605682.2020.1768809`.

[40] William L. Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *IEEE Data Engineering Bulletin*, 40(3):52–74, September 2017.

[41] Jeffrey Heer and Ben Shneiderman. Interactive dynamics for visual analysis: A taxonomy of tools that support the fluent and flexible use of visualizations. *Queue*, 10(2):30–55, February 2012. `doi:10.1145/2133416.2146416`.

[42] Florian Heimerl, Christoph Kralj, Torsten Moller, and Michael Gleicher. embComp: Visual interactive comparison of vector embeddings. *IEEE Transactions on Visualization and Computer Graphics*, 2020. `doi:10.1109/TVCG.2020.3045918`.

[43] Petra Isenberg, Florian Heimerl, Steffen Koch, Tobias Isenberg, Panpan Xu, Charles D. Stolper, Michael Sedlmair, Jian Chen, Torsten Möller, and John Stasko. Vispubdata.org: A metadata collection about IEEE Visualization (VIS) publications. *IEEE Transactions on Visualization and Computer Graphics*, 23(9):2199–2206, September 2017. `doi:10.1109/TVCG.2016.2615308`.

[44] Tobias Isenberg, Petra Isenberg, Jian Chen, Michael Sedlmair, and Torsten Möller. A systematic review on the practice of evaluating visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2818–2827, December 2013. `doi:10.1109/TVCG.2013.126`.

[45] Stefan Jänicke, Greta Franzini, Muhammad Faisal Cheema, and Gerik Scheuermann. On close and distant reading in digital humanities: A survey and future challenges. In *Proceedings of the EG/VGTC Conference on Visualization — STARs*, EuroVis '15. The Eurographics Association, 2015. `doi:10.2312/eurovisstar.20151113`.

[46] Xiaonan Ji, Han-Wei Shen, Alan Ritter, Raghu Machiraju, and Po-Yin Yen. Visual exploration of neural document embedding in information retrieval: Semantics and feature selection. *IEEE Transactions on Visualization and Computer Graphics*, 25(6):2181–2192, June 2019. `doi:10.1109/TVCG.2019.2903946`.

[47] Ilir Jusufi. *Multivariate Networks : Visualization and Interaction Techniques*. PhD thesis, Linnaeus University, Department of Computer Science, 2013.

[48] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *CoRR*, abs/1404.2188, 2014. `arXiv:1404.2188`.

[49] Bogumił Kamiński, Paweł Prałat, and François Théberge. An unsupervised framework for comparing graph embeddings. *Journal of Complex Networks*, November 2019. `doi:10.1093/comnet/cnz043`.

[50] Ashish Kapoor, Bongshin Lee, Desney Tan, and Eric Horvitz. Interactive optimization for steering machine classification. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1343–1352. ACM, 2010. `doi:10.1145/1753326.1753529`.

[51] Daniel A. Keim, Jörn Kohlhammer, Geoffrey Ellis, and Florian Mansmann, editors. *Mastering the Information Age: Solving Problems with Visual Analytics*. Eurographics Association, 2010.

[52] Andreas Kerren, Kostiantyn Kucher, Yuan-Fang Li, and Falk Schreiber. Biovis explorer: A visual guide for biological data visualization techniques. *PLOS ONE*, 12(11):1–14, 11 2017. `doi:10.1371/journal.pone.0187341`.

[53] Andreas Kerren, Helen Purchase, and Matthew O. Ward, editors. *Multivariate Network Visualization*. Springer, 2014.

[54] Andreas Kerren, Helen C. Purchase, and Matthew O. Ward. *Multivariate Network Visualization*. Springer International Publisher, 2014.

[55] Andreas Kerren and Falk Schreiber. Toward the role of interaction in visual analytics. In *Proceedings of the Winter Simulation Conference*, WSC '12. IEEE, 2012. `doi:10.1109/WSC.2012.6465208`.

[56] Atif Khan, Qaiser Shah, M. Irfan Uddin, Fasee Ullah, Abdullah Alharbi, Hashem Alyami, and Muhammad Adnan Gul. Sentence embedding based semantic clustering approach for discussion thread summarization. *Complexity*, 2020:4750871, August 2020. `doi:10.1155/2020/4750871`.

[57] Jaeyoung Kim, Janghyeok Yoon, Eunjeong Park, and Sungchul Choi. Patent document clustering with deep embeddings. *Scientometrics*, 123(2):563–577, May 2020. `doi:10.1007/s11192-020-03396-7`.

[58] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors. *CoRR*, abs/1506.06726, 2015. `arXiv:1506.06726`.

[59] Gunnar W. Klau, Neal Lesh, Joe Marks, and Michael Mitzenmacher. Human-guided search. *Journal of Heuristics*, 16(3):289–310, June 2010. `doi:10.1007/s10732-009-9107-5`.

[60] Kostiantyn Kucher and Andreas Kerren. Text visualization techniques: Taxonomy, visual survey, and community insights. In *Proceedings of the IEEE Pacific Visualization Symposium*, PacificVis '15, pages 117–121. IEEE, 2015. `doi:10.1109/PACIFICVIS.2015.7156366`.

[61] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, ICML '14, pages 1188–1196. PMLR, 2014.

[62] Jiaying Liu, Tao Tang, Wei Wang, Bo Xu, Xiangjie Kong, and Feng Xia. A survey of scholarly data visualization. *IEEE Access*, 6:19205–19221, March 2018. `doi:10.1109/ACCESS.2018.2815030`.

[63] Shusen Liu, Peer-Timo Bremer, Jayaraman J. Thiagarajan, Vivek Srikumar, Bei Wang, Yarden Livnat, and Valerio Pascucci. Visual exploration of semantic relationships in neural word embeddings. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):553–562, January 2018. `doi:10.1109/TVCG.2017.2745141`.

[64] Yang Liu, Eunice Jun, Qisheng Li, and Jeffrey Heer. Latent space cartography: Visual analysis of vector space embeddings. *Computer Graphics Forum*, 38(3):67–78, June 2019. `doi:10.1111/cgf.13672`.

[65] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[66] Krečimir Matković, Denis Gračanin, Rainer Splechtna, Mario Jelović, Benedikt Stehno, Helwig Hauser, and Werner Purgathofer. Visual analytics for complex engineering systems: Hybrid visual steering of simulation ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1803–1812, December 2014. `doi:10.1109/TVCG.2014.2346744`.

[67] David Meignan, Sigrid Knust, Jean-Marc Frayret, Gilles Pesant, and Nicolas Gaud. A review and taxonomy of interactive optimization methods in operations research. *ACM Transactions on Interactive Intelligent Systems*, 5(3), September 2015. `doi:10.1145/2808234`.

[68] Tomás Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013. `arXiv:1310.4546`.

[69] John Mingers and Loet Leydesdorff. A review of theory and practice in scientometrics. *European Journal of Operational Research*, http://authors.elsevier.com/sd/article/S037722171500274X, 04 2015. `doi:10.1016/j.ejor.2015.04.002`.

[70] Jeff Mitchell and Mirella Lapata. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429, November 2010. `doi:10.1111/j.1551-6709.2010.01106.x`.

[71] Mário Popolin Neto and Fernando V. Paulovich. Explainable Matrix — Visualization for global and local interpretability of Random Forest classification ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1427–1437, February 2021. `doi:10.1109/TVCG.2020.3030354`.

[72] Mark E. J. Newman. *Networks: An Introduction*. Oxford University Press, 2010.

[73] Giang H. Nguyen, John Boaz Lee, Ryan A. Rossi, Nesreen .K Ahmed, Eunyee Koh, and Sungchul Kim. Dynamic network embeddings: From random walks to temporal random walks. In *Proceedings of the IEEE International Conference on Big Data*, BigData '18, pages 1085–1092. IEEE, 2018. `doi:10.1109/BigData.2018.8622109`.

[74] C. Nobre, M. Meyer, Marc Streit, and A. Lex. The state of the art in visualizing multivariate networks. *Computer Graphics Forum*, 38:807–832, 06 2019. `doi:10.1111/cgf.13728`.

[75] Carolina Nobre, Miriah Meyer, Marc Streit, and Alexander Lex. The state of the art in visualizing multivariate networks. *Computer Graphics Forum*, 38(3):807–832, June 2019. `doi:10.1111/cgf.13728`.

[76] David Opitz and Richard Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, August 1999.

[77] Deokgun Park, Seungyeon Kim, Jurim Lee, Jaegul Choo, Nicholas Diakopoulos, and Niklas Elmqvist. ConceptVector: Text visual analytics via interactive lexicon building using word embedding. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):361–370, January 2018. `doi:10.1109/TVCG.2017.2744478`.

[78] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '14, pages 1532–1543. ACL, October 2014. `doi:10.3115/v1/D14-1162`.

[79] Hasan Pirkul, Rakesh Gupta, and Erik Rolland. VisOpt: A visual interactive optimization tool for P-median problems. *Decision Support Systems*, 26(3):209–223, September 1999. `doi:10.1016/S0167-9236(99)00032-9`.

[80] Eric D. Ragan, Alex Endert, Jibonananda Sanyal, and Jian Chen. Characterizing provenance in visualization and data analysis: An organizational framework of provenance types and purposes. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):31–40, January 2016. `doi:10.1109/TVCG.2015.2467551`.

[81] Jonathan C. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Proceedings of the Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization*, CMV 2007, pages 61–71. IEEE, 2007. `doi:10.1109/CMV.2007.20`.

[82] D. Sacha, M. Kraus, D. A. Keim, and M. Chen. VIS4ML: An ontology for visual analytics assisted machine learning. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):385–395, January 2019. `doi:10.1109/TVCG.2018.2864838`.

[83] Dominik Sacha, Michael Sedlmair, Leishi Zhang, John A. Lee, Jaakko Peltonen, Daniel Weiskopf, Stephen C. North, and Daniel A. Keim. What you see is what you can change: Human-centered machine learning by interactive visualization. *Neurocomputing*, 268:164–175, 2017. doi:10.1016/j.neucom.2017.01.105.

[84] Dominik Sacha, Andreas Stoffel, Florian Stoffel, Bum Chul Kwon, Geoffrey Ellis, and Daniel A. Keim. Knowledge generation model for visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1604–1613, December 2014. doi:10.1109/TVCG.2014.2346481.

[85] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988. doi:10.1016/0306-4573(88)90021-0.

[86] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, June 1990. doi:10.1007/BF00116037.

[87] Bruno Schneider, Dominik Jäckle, Florian Stoffel, Alexandra Diehl, Johannes Fuchs, and Daniel A. Keim. Integrating data and model space in ensemble learning by visual analytics. *IEEE Transactions on Big Data*, 7(3):483–496, July 2021. doi:10.1109/TBDATA.2018.2877350.

[88] Michael Sedlmair, Christoph Heinzl, Stefan Bruckner, Harald Piringer, and Torsten Möller. Visual parameter space analysis: A conceptual framework. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2161–2170, December 2014. doi:10.1109/TVCG.2014.2346321.

[89] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and Philip S Yu. A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):17–37, 2017. doi:10.1109/TKDE.2016.2598561.

[90] Henry Small. Tracking and predicting growth areas in science. *Scientometrics*, 68(3):595–610, September 2006. doi:10.1007/s11192-006-0132-y.

[91] Daniel Smilkov, Nikhil Thorat, Charles Nicholson, Emily Reif, Fernanda B. Viégas, and Martin Wattenberg. Embedding Projector: Interactive visualization and interpretation of embeddings. In *Proceedings of the NIPS 2016 Workshop on Interpretable Machine Learning for Complex Systems*, 2016. arXiv:1611.05469.

[92] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, EMNLP '13, pages 1631–1642. ACL, October 2013.

[93] Charles D. Stolper, Adam Perer, and David Gotz. Progressive visual analytics: User-driven visual exploration of in-progress analytics. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1653–1662, December 2014. doi:10.1109/TVCG.2014.2346574.

[94] Justin Talbot, Bongshin Lee, Ashish Kapoor, and Desney S. Tan. EnsembleMatrix: Interactive visualization to support machine learning with multiple classifiers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1283–1292. ACM, 2009. doi:10.1145/1518701.1518895.

[95] Martina Toshevska, Frosina Stojanovska, and Jovan Kalajdjieski. Comparative analysis of word embeddings for capturing word similarities. *CoRR*, abs/2005.03812, 2020. arXiv:2005.03812.

[96] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394. ACL, 2010.

[97] Emily Wall, Meeshu Agnihotri, Laura Matzen, Kristin Divis, Michael Haass, Alex Endert, and John Stasko. A heuristic approach to value-driven evaluation of visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):491–500, January 2019. doi:10.1109/TVCG.2018.2865146.

[98] Jiapeng Wang and Yihong Dong. Measurement of text similarity: A survey. *Information*, 11(9), 2020. `doi:10.3390/info11090421`.

[99] Daniel Witschard, Ilir Jusufi, and Andreas Kerren. SimBaTex: Similarity-based Text Exploration. In Jan Byška, Stefan Jänicke, and Johanna Schmidt, editors, *EuroVis 2021 - Posters*. The Eurographics Association, 2021. `doi:10.2312/evp.20211067`.

[100] Daniel Witschard, Ilir Jusufi, Rafael M. Martins, and Andreas 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, pages 219–223. INSTICC, SciTePress, 2021. `doi:10.5220/0010314602190223`.

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

[102] Yingcai Wu, Ka-Kei Chung, Huamin Qu, Xiaoru Yuan, and S.C. Cheung. Interactive visual optimization and analysis for RFID benchmarking. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1335–1342, November–December 2009. `doi:10.1109/TVCG.2009.156`.

[103] Erjia Yan and Ying Ding. Applying centrality measures to impact analysis: A coauthorship network analysis. *Journal of the American Society for Information Science and Technology*, 60(10):2107–2118, October 2009. `doi:10.1002/asi.21128`.

[104] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Network representation learning: A survey. *IEEE Transactions on Big Data*, 6(1):3–28, March 2020. `doi:10.1109/TBDATA.2018.2850013`.