VISUALIZATION AND STRUCTURING OF BIBLIOGRAPHIC RECOMMENDATIONS

by

Matthew Ray McQuaigue

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Computer Science

Charlotte

2019

Approved by:

____
Dr. Erik Saule

____
Dr. Kalpathi Subramanian

____
Dr. Zachary Wartell

ABSTRACT

MATTHEW MCQUAIGUE. Visualization and Structuring of Bibliographic Recommendations (Under the direction of DR. ERIK SAULE)

This study analyzes methods to structure and visualize bibliographic recommendations efficiently while conveying important information for users. Most of the work being developed in the realm of bibliography visualizations is surrounding the question of how authors are related and the network concerning their progression throughout their career. The aspect of conveying information about papers and why to read them have become or are the secondary idea when analyzing these graphs. Why should a user choose a paper over another when a visualization technique or algorithm can aid in the decision-making process? Our visualization should be the reverse/opposite of past work; Co-authorship networks are secondary with a primary emphasis on the network of recommended papers that the user wants to read. This study provides a structured pipeline for better viewing bibliographic recommendations and their relations. This method makes use of important machine learning techniques such as word embeddings and self-organizing maps to take extracted topics/key phrases and map relations to other recommended papers. This extends the typical node-link graph with links representing relations and provides spatial relations of papers that are more intuitive to a user. This study also provides exposure of metadata for customizable aspects of the visualizations for interactive searching. In addition to a 2D view of the recommendations, a side-by-side 3D view is provided for quantitative values.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SOM	Self-organizing maps
GloVe	Global vectors
BMU	Best matching unit
LDA	Latent Dirich Allocation
JSON	Javascript object notation

CHAPTER 1: INTRODUCTION 1. RESEARCH GAP

Within citation/bibliographic research, a primary method of extracting new knowledge and understanding researchers' questions and results is through visualizations. Much of the work visualizing the topics listed above are used to solve and understand the question of how authors are related and the network concerning their progression throughout their careers. As a secondary product, researches also try and map gaps of research within a certain subject area. This research analyzes a mass amount of academic literature and bibliometrics usually derived from databases such as CiteSeerX. Even though metadata about the literature is present, the primary focus is on citations, references, and authors. The graphs of such analysis only reflect a first-level view of authors and citation connections. The aspect of conveying information about the actual literature and why to read them has been a large gap in this research area. Why should a user manually choose a paper over another when the visualization can aid in this decision-making process by stressing the most suitable work based on certain criteria? A visualization for recommendations should be the reverse/opposite of past work; Citation networks should be secondary emphasis while a primary emphasis is on the network of recommended papers that the user most likely should read.

1.2 History

Bibliographic networks have been studied since the late 60's. by scientists like Derek J. de Solla Price. In his work, the goal was to explore the broadest outline of scientific networks. Like co-citation networks and bibliographic coupling today, Derek mapped scientific papers to directly related works through its citations, footnotes, and bibliography. Through analyzing direct relations of 200 papers, tightly bound 'knittings' became prominent patterns in isolated subject areas. He concluded that the knitted strips are the works of a few hundred scientists in that field that correspond to objectively defined subjects [19]. Derek predicted that if future work was focused on such strips, it would lead to determining the topology of scientific literature in a subject area, highlighting the importance of certain research sections based on its position in the map. This beginning work became a foundation and starting point for constructing bibliographic relations and extracting new knowledge of scientific areas that were previously hidden. Since then, many forms of analyzing and visualizing scientific networks have been developed with the help of graph theory and computer graphics advancements. Past research in this field has branched to new areas of work including evaluating scholarly contributions, mapping research fields and scientific papers to respective structures, tracking the flow of knowledge and gaps in research areas, and studying the progression and prediction of author's careers. Majority of the literature contributions relating to bibliographic networks are citation analysis. Citation analysis is the exploration of reference patterns in the scholarly and scientific literature [5]. In recent years, citation analysis has gained significant popularity for the use of determining research impact, and the flow of research within certain subjects. However, citation analysis is only a broad focus within citation research. Other fine-grained research areas and techniques are studying bibliometrics, bibliographic coupling, co-citation associations, and citation/bibliometric graph analysis. Within this research paper the primary contribution will be the use of bibliographic coupling and citation/bibliographic graph analysis.

1.3 Thesis Questions

This project proposes the idea of streamlining a systematic process for how bibliographic recommendations can be effectively visualized. Using graph theory, machine learning algorithms and visualization techniques, how should such a network of papers be visualized? How should the networks be formed and structured to convey the data/information in effective ways? With metadata providing information on ranking, paper title, authors, venue, topics etc. how do these attributes affect the visualization process and how are they useful? Visualizing them can remove the mystery of the information they hold while providing a more modern/natural process of searching. Contributions to these questions could change the way someone conducts literature reviews which has become an outdated process. “Visualizations of the information found in bibliographies and recommendations is a valuable resource for researchers and viewing this information in a text-based environment is among the norm and well adopted within the field of research, some information such as, inter-relationships of papers, authors, research and typology are better viewed in a visual representation. In other words, texted-based views can be hard to read and convey information in too long of a time.” [2]

CHAPTER 2: BACKGROUND 1. THEADVISOR

Theadvisor is an academic recommendation system that provides diversified ranked recommendations to its users. For this project, I will be using theadvisor’s pipeline for visualizing the recommendations from a user's query. Thus, an explanation of the system is needed. Theadvisor specializes in providing more sophisticated literature searches while focusing on three important aspects: personalization, scalability, and exploratory searching. Along with prioritizing these aspects, theadvisor specializes in the use of efficient recommendation algorithms for the construction of personalized results that include result diversification and relevance feedback. The algorithms used in this system allow for the highest accuracy of recommendation querying. Since the personalization, efficiency, and accuracy are provided, response times and interactivity are competitive with a system that is scalable to the data that will be increasing within the system [6]. With all these aspects working together, easy recognition of the correct paper a user should read can be achieved.

In theadvisor’s system, four main components are used, in Figure 1, a paper mapper which uses a citation graph that corresponds to the input from the user, a recommendation engine that recommends the papers, feedback input which intakes comments from the user on the recommendations they received and refines results, and visualizations that use graph drawing techniques for recommended paper relations, which is the subject of this paper.

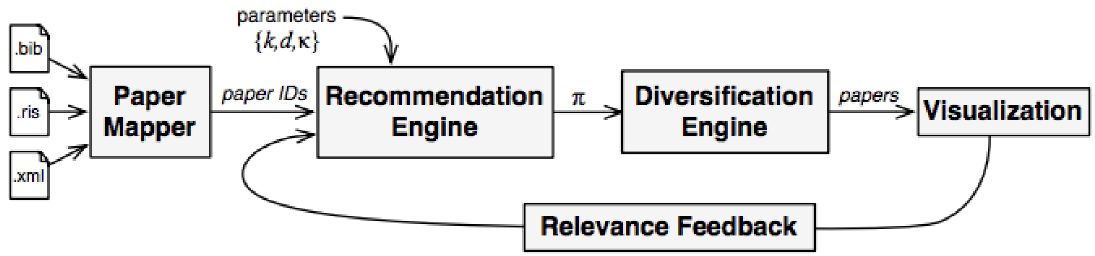

Figure 1 - Overview of theadvisor system. The visualization is the product of this work.[6]

2.2 Bibliographic Networks

Bibliographic networks are comprised of bibliographic objects linked by relations. The objects in this type of network can be scholarly publications, authors, and topics related to these nodes. These relations between objects in a network can be represented in many ways such as nodes, links, colors, shapes etc. For a relation to be distinguished between two objects in a bibliographic network, calculations and comparisons must be derived from metadata provided by the publications. Metadata can be titles, author(s), publisher, language, and sometimes the abstract. Relations can be computed in many ways through co-citation analysis, related authors, and subject/topic relations. These combinations of relations and representations are the building blocks of bibliographic networks.

To begin visualizing bibliographic recommendations, an understanding bibliographic calculations and basic protocol is in order. In a network two main goals can be derived as the motive for this work:

1. Collecting and preprocessing objects for the use of conveying insightful information about papers/authors so users can browse, and retrieve this information for their need

2. Using the objects, meta-data, and relations to discover new information for further analysis and future research

With the former being the primary focus of this research. To understand the architecture of a bibliographic network, we can begin by defining one/two-mode networks.

1. One-mode networks are networks containing one set of nodes with links describing the relations between them. These networks can be derived from two-mode networks.
2. Two-mode networks are networks with two kinds of datasets as nodes with ties between the two datasets representing edges. These two networks are split as either a primary/top level node set or a secondary/bottom level node set.

Bibliographic data falls within a two-mode network because of having paper nodes and author nodes. The primary node set is usually decided by the process direction by which node set dictates the existence of another. For example, an author can choose to write a paper or not thus creating the possible existence of the work as an object in the network.

2.3 Projections

To analyze a two-mode network, a transformation must be applied to take the two-mode network into a one-mode network. This transformation is called a projection (Figure 2). The process describes taking one set from the two-mode network and linking its nodes based on the shared relation they have with the other node set. The product of this transformation allows for the connection of both data sets interchangeably to establish groupings. Once projections are performed on the data and connections are established, weights could then be attached to

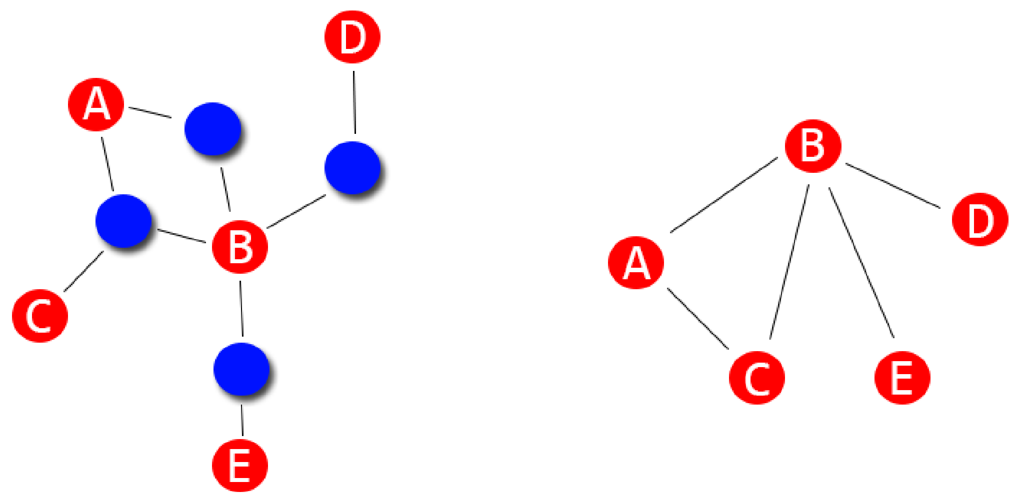

Figure 2 - (left) two-mode network with papers as red nodes and connected authors as blue nodes (right) one-mode projection of the two-mode network

edges based on co-occurrences calculated by some measure of relatedness. The co-occurrences can be calculated simply by $w_{ij} = p1$ where w_{ij} is the weight between node i and j with p being the related node that i and j share occurrence with. The flaw with this co-occurrence calculation is that it doesn't consider the quantity of collaborators. Newman [18] argued that papers containing fewer collaborators have more impact on relationships and relatedness of work than papers with more collaborators. Newman stated that "Even on a paper with four or five authors, the authors probably know one another less well on average than authors from a smaller collaboration." To account for this extra calculation, the equation above is adjusted to: $w_{ij} = p1/N_p$-1 where N_p is the number of authors on paper p. Now accurate weights between two papers can be used to represent relatedness from authors in a projected one-node network. This becomes a binary two-mode network which is the weighted one-mode projection of a two-mode network.

2.4 Word Embeddings

Word embeddings is a broad naming of different sets of language modeling and feature learning techniques in natural language processing. In this process, words are mapped to numerical vectors of real numbers. These vectors can be derived by using a neural network to learn explicit representations of terms within the context it appears or through other methods mentions later. The vector dimension length can be different depending on the model and precision needed. Machine learning models usually take vectors as input and provide a vector as output representing the word. If working with text, a method of representing that text as numerical data is needed to input those words into a model. Converting text into numerical vectors is known as vectorization.

There are different methods of vectorizing text. The first method, originally borrowed from its original use in computer architecture, [20] which is called a one-hot encoding. One-hot encoding takes a sentence and tokenizes its unique words. A zero vector is then generated of equal length to the tokenized sentence of unique words. A matrix of the words is then created and a one is placed at the index corresponding to that word. For example, consider the sentence “he was as fast as light”. Once unique words are extracted, the sentence is {he, was, as, fast, light}. In the table below, the vectors are generated, and a matrix is created.

	he	was	as	fast	light
he	1	0	0	0	0
was	0	1	0	0	0
as	0	0	1	0	0
fast	0	0	0	1	0

light	0	0	0	0	1

Table 1 – A one hot matrix encoding

This may be a fast and simple method but is extremely inefficient. Since only one index of the vector for each word has a non-zero number associated with it, most of the embedding is comprised of zeros. This embedding, in a machine learning model, would assume all words are equally similar.

Another possible method of choice for embedding is encoding each word with a random number. Given two sentences tokenized of their unique words {he, was, as, fast, light} and {he, was, late, last, night}, in the table below, this method would result in a sentence vector where each number is associated with its respective word no matter the context.

he	was	as	fast	light
25	46	87	43	65

he	was	late	last	night
25	46	290	30	40

Table 2 – Random word encoding

As you can see, ‘he’, and ‘was’ are given the same number no matter the context. Without context as a parameter in determining the value associated to the word, the true meaning of the word is not captured by the embedding. This results in meaningless encodings where relations could be inaccurate.

Using word embeddings, an efficient representation of text can be achieved in which similar words have similar encodings. The dense vector encoding is composed of floating-point values that are generated by a trained network of a parameter length. Since a model is learning the word associations and generating the vectors based on the learning associations, words that are the same but come from different contexts can have a different vectorization. Below is an example word embedding vectorization of ‘he’ and ‘was’.

he	1.5	-0.43	0.78	2.45	0.2
was	5.7	0.45	-0.1	1.4	1.6

Table 3 – Example vectorization of a word from an embedding

2.5 Self Organizing Maps

Self -organizing maps are another unsupervised learning neural network used for feature detection. The model takes in high dimensional data to perform dimensionality reduction to produce a low-dimension space for classification, outlier detection and quantization. The system is based on competitive learning where nodes or neurons compete to decide which is activated for a set of inputs called a winning neuron [21] [Figure 3].

To begin, the weights are initialized with random values. All neurons will compute a discrimination function below for the input features where D is the dimension of the input, x is the input, and w is the weight. The neurons whose weight is closest to the input is called the best matching unit (BMU).

$$d_j(\mathbf{x}) = \sum_{i=1}^{D} (x_i - w_{ji})^2$$

Once a neuron has competed for activation and won, neighboring neurons will fire slightly more than neurons of greater distance from the BMU. The BMU weights and the neurons closest are adjusted toward the input vector. This is called topological neighborhood and is calculated below

$$T_{j,I(\mathbf{x})} = \exp(-S_{j,I(\mathbf{x})}^2 / 2\sigma^2)$$

Where S is the lateral distance between neurons, I(x) is the index of the winning neuron and σ is the number of neighbors that decrease over time. This decrease will continue to zero as the distance from the winning neuron increases [21].

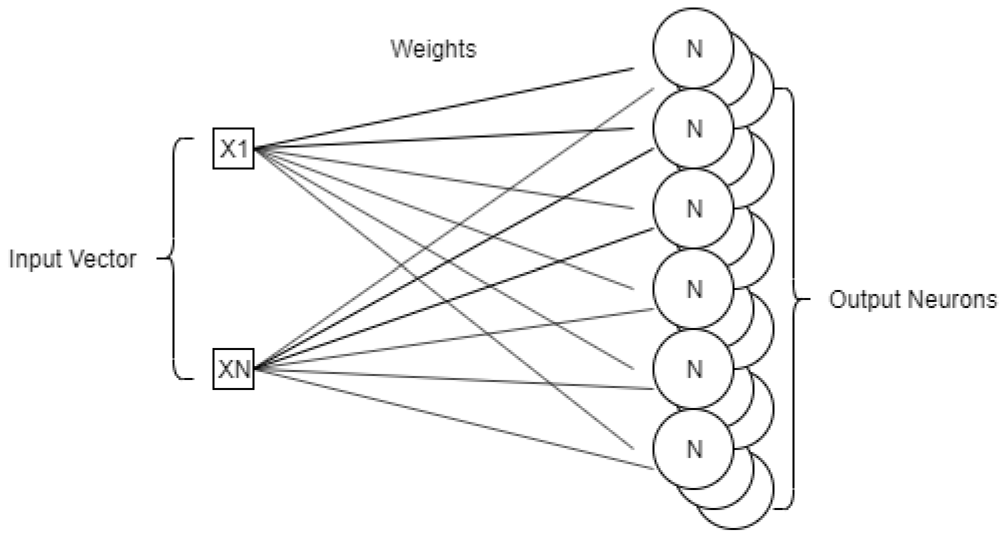

Figure 3 – Self-organizing map model. The left side are input vectors which are the word embeddings. The output are neurons from the topographic map

These formulas and steps are repeated for several epochs that produce a network of associating output nodes with patterns in the input data.

CHAPTER 3: RELATED WORK

The visualization of bibliographic networks is a newer form of the study. With the increased access to bibliographic data, production of efficient visualization techniques and data science grow in popularity. It is only a matter of time before many researchers try to structure scientific papers and collaborations for new knowledge discovery. The relation of bibliographic visualizations and citation analysis are useful for each other's success. Through the process of structuring research fields and organizing the knowledge and published cliques, the methods and understandings of the outcome data can be utilized in visualizations. Just as citation analysis work takes a set of metadata (documents, authors, publishing) from a selected area of work or from a large database and performs calculations using this data, visualizing scientific works require the same process. With the metadata information, a visualization uses them as parameters and variables within formulas to represent their aspects as a high-level abstracted view for quick recognition of the aspect pertaining to object. The aspects can be represented as nodes, colors, thickness, shapes, sizes, and positioning in the visualization space.

In the visualization space, bibliography can be studied and represented as a macro-structure which is a mapping of the entire scientific community with nodes representing a certain topic or discipline, or a micro-structure where each node represents a single document or contributor in that field.[5] The main idea of visualizations is to summarize the data rather than represent specific works in a singular fashion. Two steps are involved in the creation of these visualizations, 1) deciding which data, or subset of data, to be used in representing the nodes and links within the graph as the main information. 2) how should the visualization be structured to convey the relationships. If these two steps are done efficiently, a visualization can provide information to a user much quicker than text-based representations in a natural way. Analyzing previous work in bibliographic/citation visualizations can provide a roadmap to what techniques work and which are unneeded.

3.1 Histcomp

Histcomp is a bibliographic analysis and visualization tool used for the biological bulletin. Histcomp argues that bibliographic networks tend to be scattered and that visualization requires criteria of convergence [1]. The scattered tendencies are tied to the paper to author ratio where more authors are abundant than papers, thus some authors will not have connections with another. Within histcomp's visualizations, they use first and second authors for relations and a local citation score to combat this issue of sparseness. Using a ranked author list and a local citation score, histcomp can judge relation and popularity of a paper.

Along with a citation matrix, hist comp boasts the use of two view network where level 1 is for typology (areas of research for papers) and topic relation, while level 2 is for paper references and commonality between authors. These use of different level views provides a better understanding of the structure of literature and their connections. Their citation matrix allows for clustering using citing nodes (bibliographic coupling group documents) and clustering using cited nodes (co-citation link documents). Using these methods, histcomp can achieve a well-structured topic/subject space for easy recognition of relations in academic subject areas. Along with citation matrices, Histcomp also uses LCS which is the local citation frequency. Using this, and dictating an LCS threshold of greater than N, the visualization will only display works with more cites than the LCS.

3.2 BiblioViz

BiblioViz is a project birthed from the InfoVis 2004 contest lend for the development of bibliographic visualization tools/systems. After analyzing many of the aspects from tools submitted to the contest, BiblioViz used the best to create an efficient bibliographic visualization tool in attempts to give the maximum number of views for the data using a minimum number of constructs, thus the visualization must be cohesive and efficient in the methods used to convey the information [2].

To begin, BiblioViz asked themselves three questions:

1. "What are the similarities and differences, and advantages and disadvantages of each system?"
2. "Which of these systems should I use for my needs?"
3. "Is it possible or feasible to integrate two different methods offered by different systems?"

From any different visualizations submitted, BibliViz analyzed visualizations of type: table, network (2D & 3D), and node placement without network relations. Visualizations were considered based on the efficiency of expressing relationship to data and how effectively the information was conveyed. The goals of BiblioViz's

1. "Characterize the research areas and their evolution,"
2. "Where does a particular author fit within the research areas?" and
3. "What are the relationships between two or more of the authors?"

The conclusion was that network visualizations are better for relationships and tables are better for time-related information. BiblioViz continued to use these two different display formats for their system and included a paper details panel, data menu, and a user control panel. The table view is a 2D table with the x-axis depicting time and year, and the y-axis can be chosen between publication venues, authors, and research area. Each paper is represented as a rectangle placed into a cell. If a paper has multiple authors or multiple research areas, the square in the respective cell is split and colored accordingly. Through the user controls, the user can choose to sort the research areas in terms of their importance or arrange them by relationship.

On the network side of the visualization, it follows the same customizable attributes of the table view. The user can filter components and draw the network consisting of authors papers, publication venues and research area. Highlighting components are used to allow users to choose certain entities in the network and focus on its relations to other entities. The network visualization provides a 2D orthogonal view of the network and a 3D view. "The 2D view looks at the network plane(s) from the top, so if multiple planes exist, they will be blended into one. The 3D view looks at them from an oblique angle. Cylinders, spheres, and lighting are used to enhance depth cues in the 3D view." The 2D view is for paper contents while 3D is for authors related to those papers.

3.3 Managing and Visualizing Citation Network Using Graph Database and LDA Model

Research papers contain basic metadata such as: paper ID, publication year, paper title, and authors. This project proposes an additional piece of metadata for research papers called a "topic vector". This new piece of metadata contains the topics within the paper extracted using the LDA (Latent Dirich Allocation) algorithm which considered each document or paper as a bag of words. "LDA infers a representation of document as a distribution of K topics and each topic is a distribution of V words using Gibb Sampling algorithm" [3]. They also provide a new method of storing and managing the citation network using a graph database. The use of graph query language is then used to perform operations of citation analysis on the network.

CHAPTER 4: MACHINE LEARING IMPLEMENTATIONS 1: DATA COLLECTION

Using theadvisor's API, a collection of paper recommendations can be achieved. Once a bibliography is submitted through its system, a JSON of metadata for each recommendation is provided. This will then require the preprocessing of data to extract the arrangement into the appropriate form for construction into a network and the use of calculations. Theadvisor also supports keyword extraction from the recommended papers. For each paper these keywords will be used and filtered into a topic vector for the aid in structuring and relation operations.

For the construction of a bibliographic network, the assembly of bibliographic data must be organized from a source. Many researchers in the area of creating such networks from citation and bibliographic data get their resources from well-known bibliographic databases such as CiteSeerX. A bibliographic database is a database containing records of bibliographic data in the form of a digitized collection containing journals, newspaper articles, conference proceedings, reports, books, abstracts, etc. (Feather, John; Sturges, Paul, eds. (2003)). The resulting data within these databases are not complete workings of said data but resulting metadata that represents useful and important information about that work. In the area of bibliographic metadata, the information contained can be different depending on the type of work you are viewing. For example, books/journals/conference papers may have metadata listed as: Author name, title of publication, article title, publication date, publisher, publication ID. The objective is to use this data to create a bibliographic network to gain insight on new knowledge and relations among different bodies of work. Theadvisor recommendation system has scraped and contains over 1.9 million computer science articles from DBLP, 740k technical reports on physics, mathematics, and computer science from arXiv, and 40k publications from HAL-Inria. To increase the number of edges (citation relations), theadvisor also obtained data from CiteSeerX. After the merging of data from former datasets and CiteSeerX, the final citation count is ~1M papers and ~6M references [6]. Using this system, the necessary bibliographic information can be gained to construct a network from recommendations.

To begin, the system is presented with a bibliography file in the form of BibTeX, RIS, or EndNote XML Format. Upon requesting for bibliographic recommendations, the system will respond with a list of top 100 recommended academic papers based on theadvisor's recommendation engine and Diversification engine. To visualize these recommendations, a data format must be gathered for further editing. The recommendation data is presented by providing an ID for each paper. This ID is a representation of the paper signifying its publication info such as conference, publication type, and database origin. An example paper ID from DBPL is DBLP:journals/sigcse/Pillay09. These paper IDs map to the papers metadata that is needed in the visualization process.

The information for the paper metadata is in JSON format. The metadata for these papers supply information about title, venue, type, vol, issue, pages, year, authors, ids, all ids, references, and citations. An example of the JSON format for metadata pertaining to a single publication appears as:

```
{
        "title": "Learning difficulties experienced by students in a course on formal languages and automata theory.",
        "venue": "SIGCSE Bulletin",
        "type": "0",
        "vol": "41",
        "issue": "4",
        "pages": "474-476",
```

```
        "year": "2009",
        "authors": ["Nelishia Pillay"],
        "ids": ["DBLP:journals\/sigcse\/Pillay09"],
        "allids": ["DBLP:journals\/sigcse\/Pillay09"],
        "citations: […],
        "references": […]
}
```


All IDs for a given paper is all the IDs from different databases that can map to this metadata. The 'types' object is represented at the end of the JSON representation and describes the type of publication:


```
"types": [{
        "name": "JOURNAL",
        "bibentry": "article",
        "venue": "journal"
}, {
        "name": "CONF",
        "bibentry": "inproceedings",
        "venue": "booktitle"
}, {
        "name": "REPORT",
        "bibentry": "techreport",
        "venue": "institution"
}, {
        "name": "THESIS",
        "bibentry": "phdthesis",
        "venue": "school"
}, {
        "name": "BOOK",
        "bibentry": "book",
        "venue": "note"
}, {
        "name": "MISC",
        "bibentry": "misc",
        "venue": "note"
}]
```


In addition to the objects within the metadata JSON structure, another data object is added which is a topic vector. The topic vector contains strings of text that is representative of the key phrases extracted from the literature associated to it. To extract these key phrases, 3 methods of graph-based ranking were used: textrank, betweenness, and degree centrality. The idea behind graph-based methods is to construct a graph that represents the text and encodes the relationship between words in a meaningful way.

Typically, words appearing in the text will be taken as nodes, and edges represent semantic relationships between words. Then, the key phrase extraction task is transferred into a graph ranking problem based on the importance of nodes. The importance of a word is determined by its relatedness to others. In other words, a word is important if it is related to a lot of words or some words that are important. Each edge can be deemed as a vote from one node to another. After convergence, graph-based methods select top ranked nodes as keywords. [21].

Using these three methods, different calculated key phrases are assigned to each paper. In the derivation, a top N key phrases are generated while becoming less useful for larger values of N. For my implementation of word embeddings, 15 key phrases are an efficient sentence representation for each piece of literature. Since three methods of key phrase extraction is used, one type will be fed into the word embedding for comparisons of vectorization and mapping results.

4.2 Word Embedding Creation

Once the data has been pre-processed and the topic vectors have been generated, it is ready for consumption into the word embedding algorithm. There are different techniques of word embeddings for different purposes. Each word embedding technique has libraries and tools built and developed to implement that method.

4.2.1 Skip-Gram

Skip-gram is one of the learning techniques used to find the most related words for a given input which is the context of a word. The architecture of a skip-gram takes a target word input as w(t). This word is transferred to the one hidden layer that performs a dot product on a weight matrix and the input vector. The number of neurons present in the hidden layer can be denoted by N. Thus, to calculate the dot product between the weight matrix and the input target word vector, the weight matrix for the hidden layer must be of dimensions [|V|, N]. This outputs a vector H[N]. The hidden output vector computes a dot product with the output layer weight W’ [N, |V|]. The final output vector O[|V|] is then given [Figure 4]. Following this architecture, words to be used in this method are first vectorized into a one hot encoding with a dimension of [1, |V|] with |V| being the length of the known unique words in the vocabulary. Since no activation function is within the hidden layer, the dot product weight matrix [|V|, N] and [1, |V|] is passed directly to the output layer for the final H · W’ product. Over each iteration a new output vector O[|V|] is computed. A probability using SoftMax function is found for each output vector to determine highest probability of result. This probability is calculated in the formula below for each target word replicated in the vocabulary. The target words forward propagation is passed k times resulting in |v| * k in each epoch [22].

$$p(w_{c,j} = w_{O,c} | w_I) = \frac{\exp u_{c,j}}{\sum_{j'=1}^{V} \exp u_{j'}}$$

W(c,j) is the jth word predicted on the cth context position. W(O,c) is the word present on the cth context position. W(I) is the only input word and c(c,j) is the jth value in the output vector when predicting the word cth context position [22].

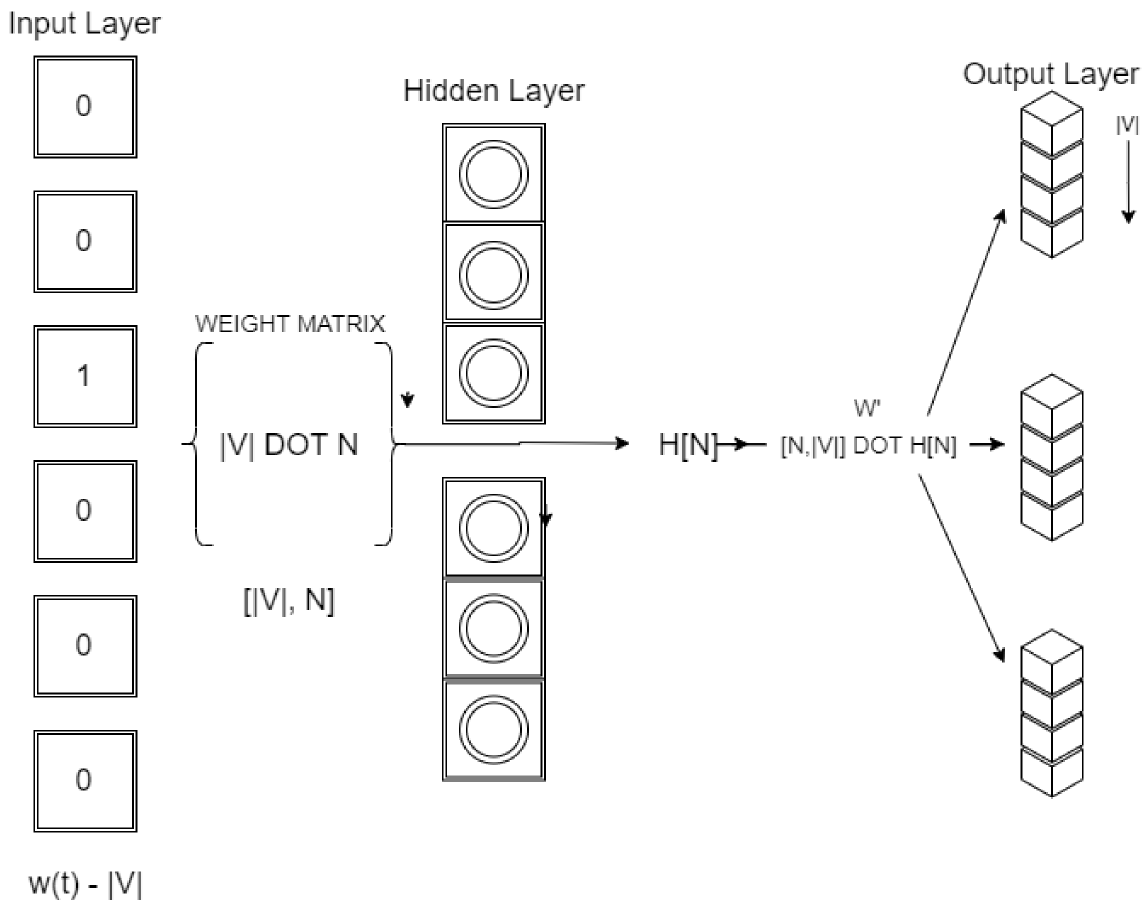

Figure 4 – Word embedding model. The left side are input vectors which are the one-hot encoding of a word. The outputs are words similar by context.

Since the goal of understanding the context of a topic in a topic vector is exactly what is needed to relate papers containing the same target topic surrounded by possible different topics, this model can perform accurate predictions. The input vectors in the Skip-Gram are split in ‘sentences’ that determine the context of a given topic. A corpus or long string of text would be downloaded and fed into the model for training.

For the first implementation of a word embedding the Word2Vec’s Skip-Gram model was used. Word2Vec is a group of word embedding models that provide pre trained shallow neural networks and high-level tools for text corpus implementations for training. In this implementation the text8 corpus was used which is a Wikipedia dump of

253,855 unique words and 17,005,208 total number of words. In addition to using this corpus, some of the sentence contexts may not pertain specifically to the certain computer science jargon or contain it at all. To compensate for this issue, we iterate over each topic vector, cleaning and tokenizing sentences and concatenating the data to the text8 corpus [Figure 5].

```
0     [computer, visualization, tools, science, theo...
1     [computer, visualization, tools, process, obje...
2     [computer, visualization, tools, science, theo...
3     [solid, mechanics, ebook, web, based, multimed...
4     [mechanics, ebook, fluid, solid, education, un...
5     [mechanics, ebook, solid, web, based, multimed...
6     [image, maker, i, abstract, new, les, combined...

[213 rows x 1 columns]
```

Figure 5 – The cleaned topic vectors from each recommended paper. Each array will act as a sentence appended to the text8 corpus

Though, each key phrase extraction method is different, all three are used to triple the amount of context possible sentences. Therefore, some sentence/topic vectors have duplicating words. Once the model is done training, we can see the embedding result below for the sample word ‘learning’ as a vector.

```
[-1.5541319  -1.7148365  -1.7215875  -1.5565938  -0.77502215 -0.57989466
  1.0604031   0.86727494 -1.1548496  -1.4317769  -3.5657969   1.8053453
 -1.0907987  -1.1501547   1.55118     2.5965557   0.47875923  0.8985616
  0.39752457 -0.22825074  3.2378314   1.4757442  -0.1374613  -1.7482446
 -0.92192453  0.23891824 -0.9744908   0.53777504 -1.2941656  -0.60780716
 -1.0364228   0.09502112  1.034104    0.05814997  2.3076177  -1.0648476
 -0.01070223  0.5363094  -1.1342664  -2.9093564  -1.5853978  -0.39045832
  1.435394    2.2275605   1.4375108   0.33395314 -0.33838373 -0.2205911
 -1.0161955  -1.3895769   0.3550209  -0.3390755  -0.60748124 -0.1548637
 -0.40907773 -0.20968722  0.8460224   1.3669626   0.01648097  1.3900192
```

-1.8478782 -0.76736945 -1.7130573 0.05296636 -1.9114523 -0.1322503
-2.252343 -3.8123522 -2.4941251 1.5630381 -0.8679548 -2.218669
-1.27287 0.7512572 1.1250979 -0.32962218 0.58618563 3.8824067
0.90416026 -1.4789664 -1.1762164 0.665379 -0.69387007 0.8051734
1.9957956 -1.9244237 0.13805343 -1.3076774 -0.2581305 -1.2291732
-2.3814228 0.11748409 0.8229781 0.1759938 1.3424524 2.177885
-0.09534218 0.66231436 -3.6083279 0.28906384]

Since this is a Skip-gram model, you can also view the related words along with the cosine similarity value. The cosine similarity is a measure of similarity between two non-zero vectors that measures the cosine of the angle between them. Given two vectors, which are the target word and word of comparison, the cosine can be derived using Euclidean dot product formula.

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \, \|\mathbf{B}\| \cos\theta$$

Given two vectors of attributes, A and B, the cosine similarity, cos(θ), is represented using a dot product and magnitude as

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}},$$

The resulting similarity ranges from −1 meaning exactly opposite, to 1 meaning the same, with 0 indicating orthogonality or decorrelation, while in-between values indicate intermediate similarity or dissimilarity [22]. This is shown below by printing words similar to ‘learning’.

[('teaching', 0.6963135004043579), ('knowledge', 0.630664587020874), ('expertise', 0.6280145645141602), ('cognitive', 0.6205146312713623), ('studying', 0.6131911277770996), ('communication', 0.6081438064575195), ('education', 0.598731517791748), ('creativity', 0.5966799259185791), ('psychology', 0.5942460894584656), ('systematic', 0.5846485495567322)]

Plotting these values, by using t-SNE we can see visually the similarities [Figure 6]. T-SNE works by taking high-dimensional data such as the embedding vector for a target word and compresses it to a 2-dimensional representation in a x, y plane (dimensionality reduction). This keeps similar words close together while dissimilar words have greater distance.

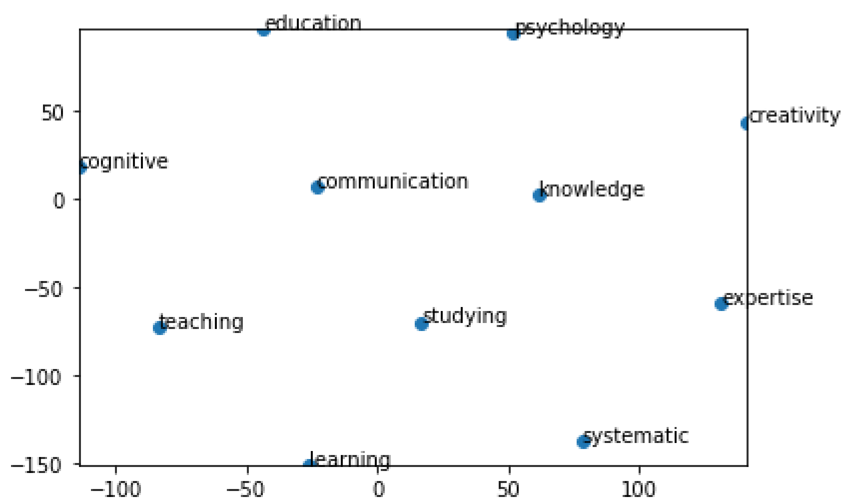

Figure 6 – The t-SNE visualization of the words similar to 'learning' for the Skip-Gram model

4.2.2 GloVe Model

GloVe is another word embedding model that stands for global vectors. Like other word embedding models, it is an unsupervised algorithm for vectorizing words to achieve accurate mappings where the distance between words is the semantic relatedness. GloVe is ultimately an improvement on the Skip-Gram method by, in addition to using local context-based methods, it implements global matrix factorization methods such as latent semantic analysis. The reason for the use of these two methods in combination is for improvement is because global methods do poor with analogy tasks, but context methods do well in such tasks while failing to capture the global context. The two methods combined essentially compensate for each other's weaknesses [23]. To begin, co-occurrence statistics on words in the form of a co-occurrence matrix is constructed as X. Each element X_{ij} of the matrix represents how often I appears in the context of word j. This can be calculated by scanning the entire corpus and for each term within a define context window size is accounted. Using a decay formula, a less weight for distant words is given by:

$$decay = 1/offset$$

Next, for each word, a soft constraint is defined by the formula below where w_i is the vector of the word and w_j is the vector of the context word. b_i and b_j are scaler biases for the main and context words [23]. These soft constraints express required and preferred properties within the model. The preferred properties refer to a scale of elements, vectors and biases, which are ordered to pose different restrictions.

$$w_i^T w_j + b_i + b_j = log(X_{ij})$$

Finally, a cost function is derived to measure the performance of the model for the given batch of data.

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij})(w_i^T w_j + b_i + b_j - \log X_{ij})^2$$

$F(X_{ij})$ represents the weighting function which prevents learning for common words described within the XMAX value [23].

$$f(X_{ij}) = \begin{cases} (\frac{X_{ij}}{x_{max}})^{\alpha} & \text{if } X_{ij} < XMAX \\ 1 & \text{otherwise} \end{cases}$$

For this model we will be using the text8 corpus again and concatenate topic vector sentences into the corpus to compare results later once word vectors are input into the self-ordering maps. To create the co-occurrence, we iterate over the corpus and generate batch sizes. The batch sizes will calculate the labels and co-occurrence weights for the distances from target words. If we iterate over the co-occurrence matrix, we can generate sample chunks.

Target Word: "classification"

```
Context word:"of"(id:2,count:40.58), "the"(id:1,count:39.08), "and"(id:3,count:15.83),
"UNK"(id:0,count:14.58), "a"(id:6,count:13.08), "system"(id:92,count:10.83), "is"(id:11,count:9.83),
"in"(id:5,count:7.83), "for"(id:14,count:7.58), "to"(id:7,count:5.50),
```

Now, evaluating the GloVe model, we can use the loss function calculation above to evaluate the loss at each epoch.

```
Average loss at step 0: 128.720076
Nearest to seven: randle, goddard, nazism, classification, dantzig, construction, hydrocodone, sparc,
Nearest to high: marxian, inflected, unstructured, sundarbans, gast, executing, nanotube, msi,model.
Average loss at step 100000: 0.132654
Nearest to seven: one, eight, four, six, nine, five, three, two,
Nearest to high: school, low, speed, level, quality, levels, education, acid,
```

Finally, we can view the word embeddings with a t-SNE 2D projection [Figure 7].

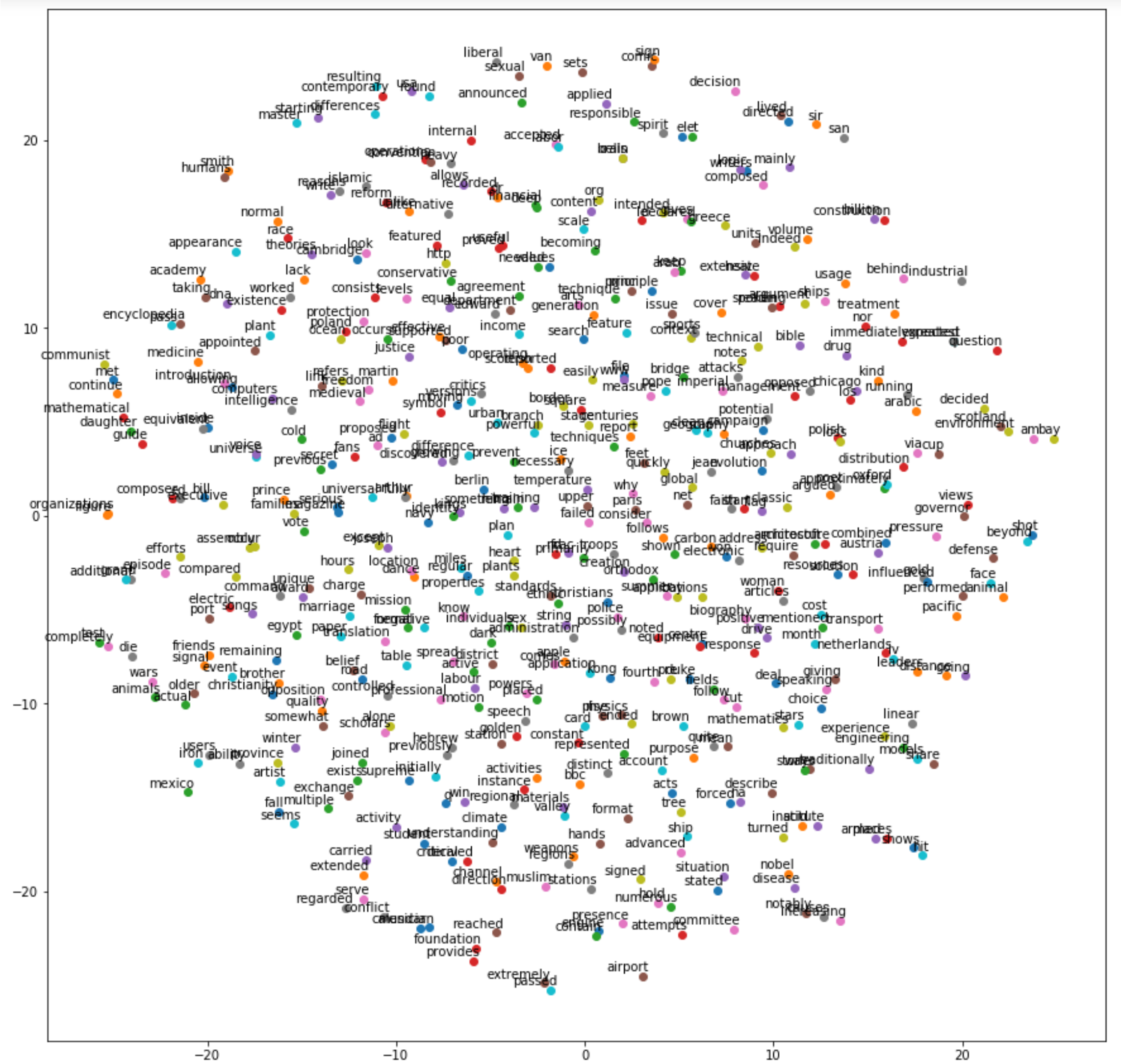

Figure 7 – The t-SNE visualization of the words like ‘learning’ for the GloVe model

4.3 SOM Implementation

To illustrate the self-organizing map, we are going to take the vectorized results from the skip-gram and GloVe word embeddings and project the 15 values or topics in 2D space for each piece of literature. We will use the MiniSom library from creating self-organizing maps.

To begin, we take the paper topic vectors and retrieve the top 15 topics of a given method. Each topic is fed into the web embedding and returned the vectorization of that word. For each vectorization, we average the values to get a single floating-point representation of the word. Then, each word is pushed into a data array for that papers ID:

```
CID:6392694 =
[[-0.02414785884320736], [-0.04227916896343231], [0.0419791080057621], [0.05210078880190849], [-0.02414785884320736], [-0.04227916896343231], [0.07087530940771103], [-0.02414785884320736], [-0.02414785884320736], [-0.10492909699678421], [0.04417248070240021], [-0.02414785884320736], [0.156926691532135], [-0.1351669281721115], [-0.02414785884320736]]
```

Next, a 900 neuron self-organizing map with a learning rate of 0.4 (adjustment in the neuron coefficients) and a random 10000 batch for training is initialized. After Cleaning and initialization, the current steps are conducted within the MiniSom self-organizing map algorithm:

1. Choosing a vector randomly from training batch
2. Every node is visited to calculate which weight is closest to the input vector
3. Obtain the BMU from the computed weights
4. Calculate the neighborhood of nodes within the BMU
5. Adjust the weights of the neighbors to the BMU

Once the training is done, the matrix can be visualized by obtaining the BMU coordinates for each vector and map to a 2D graph shown in Figure 8.

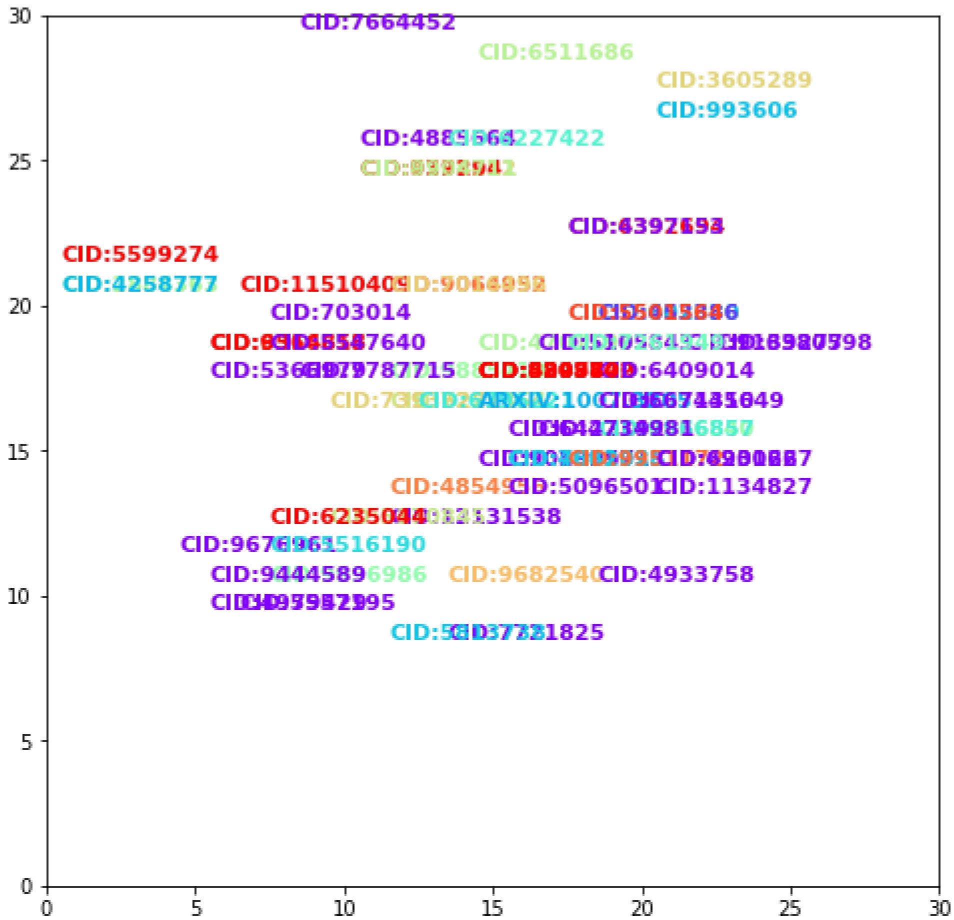

Figure 8 – The topographic map of the papers after propagation through the SOM for the GloVe model (paper IDs)

In figure 8 above, you can see that the resulting papers are visualized into groupings based on color with most similar papers being closer together and having the same colors.

Similarly, we can duplicate this mapping with the word2vec skip-gram word embedding to compare results.

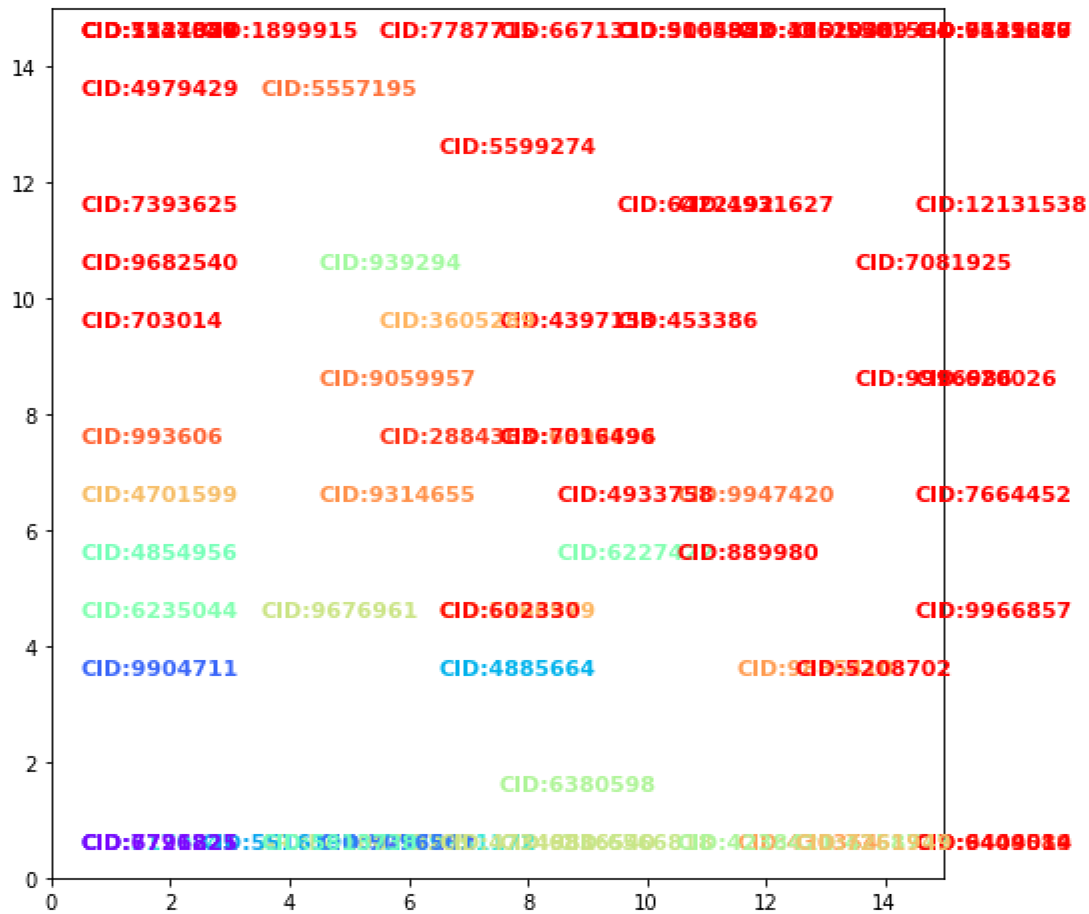

Figure 9 – The topographic map of the papers after propagation through the SOM for the Skip-Gram model

As you can see, the results are more sparse compared to the GloVe model [Figure 9]. A related group is at the upper right limit of the graph and the similarities change and trend down to the lower left limit of the graph. Going forward the GloVe method will be used. We can modify the visualization labels to view a keyword associated with each paper in the GloVe model below.

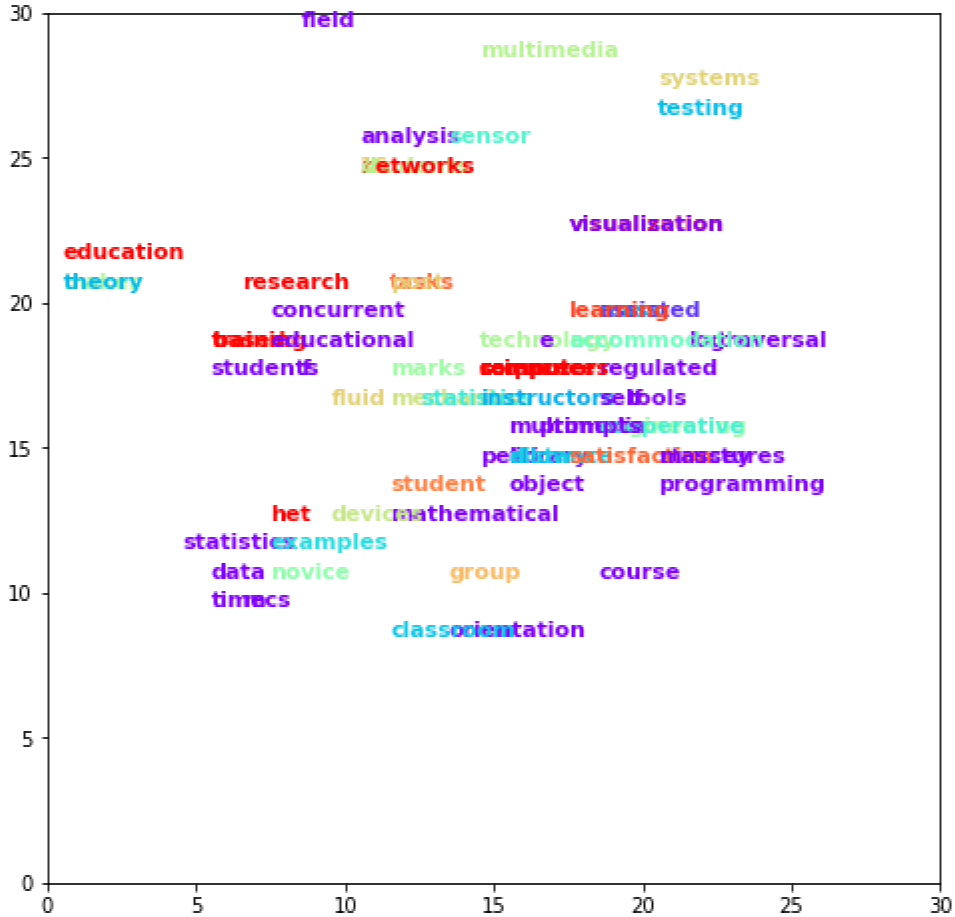

Figure 10 – The topographic map of the papers after propagation through the SOM for the GloVe model (first Topics IDs)

The topographic map has been adjusted and learned from the possible features extracted to alter weights and distance to represent relation within the map. This results in natural clustering of papers and topics related to each other.

CHAPTER 5: RESULTS 1: NODE-LINK GRAPH

After the machine learning models have been implemented and have calculated a relationship of topics for topographic structuring, visualizations of the papers for users within a recommendation system can be formed as a node-link graph. We first begin with projection of the data from a two-mode network to a one-mode network. The one-mode network to be abstracted will be the network of papers as the primary mode is of authors. For each paper, a calculation for relatedness must be performed for weight assignment. This weight can be used within the structuring algorithm for edge count from one paper to another. The preliminary edge weight calculations can be done from bibliographic coupling relations and factored into future calculations.

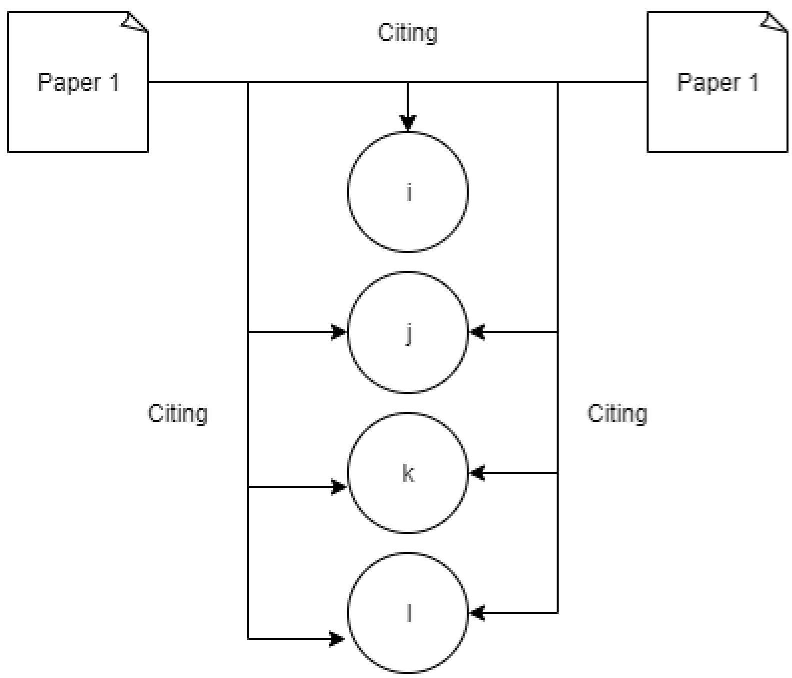

Figure 11 – Bibliographic coupling of two papers

In Figure 11 above, is the typical structuring for bibliographic coupling calculations. Using this method, a simplistic calculation for paper relation can be used.

One problem is that some papers may have a relation to one or many papers while a few may have no relations at all. This fundamental problem is that the frequency or weights connecting two papers tend to correlate with the importance of the paper over others. If following this path, relevant papers that are not connected with another may be clustered/structured incorrectly. Two main techniques can be used to help combat this issue.

- Try to connect the node based on other criteria separate from the rest. This means searching within other metadata attributes to find a potential match and make a connection from it. This can be combining weights for co-author relation and bibliographic coupling.
- Assign each object (paper) in the visualization to exactly one cluster and use factor analysis to relate based on topic vector metrics.

Like implemented in Histcomps design for relating papers, using a co-occurrence matrix and co-authorship connections can provide easy calculations. For a recommendation system like theadvisor, the network graph is limited to few co-authors and coupling relationships. In this new implementation, the use of the key phrase vector can be used to also link relations between papers.

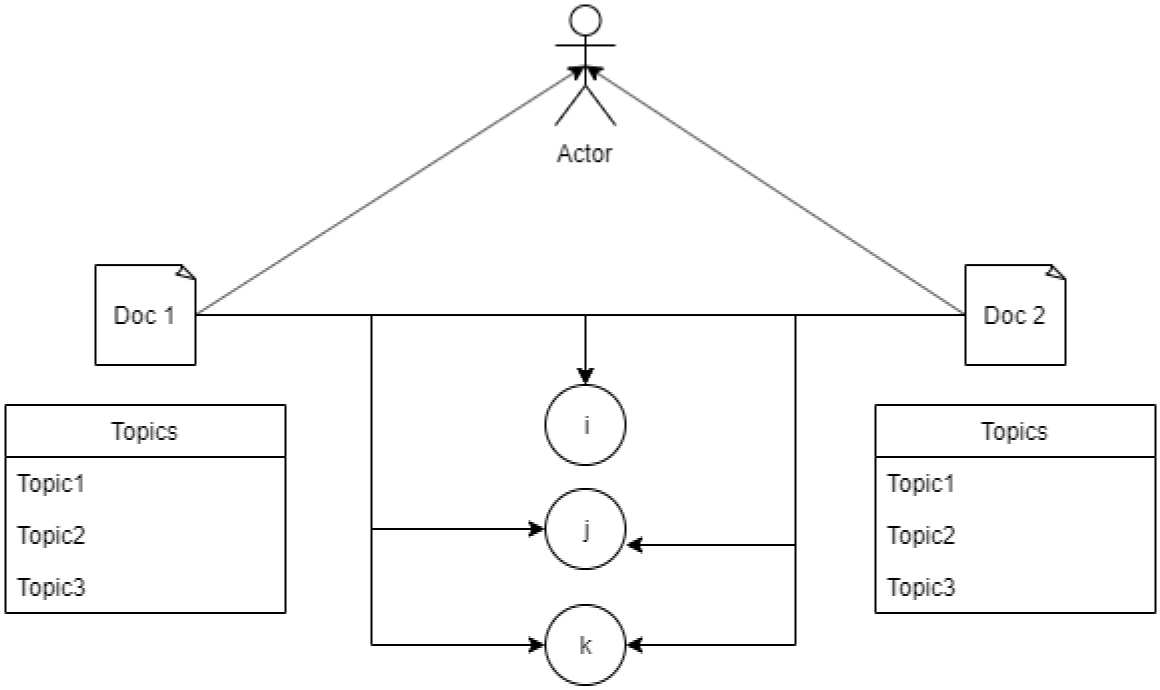

Figure 12 – New relational system for edge creation

In figure 12, a new implementation of bibliographic relations is used to combat sparseness in a small network graph and expose some of the papers metadata to configure new relations. Below Is a visualization of a node-link graph using relations from all three aspects of meta-data mentioned above.

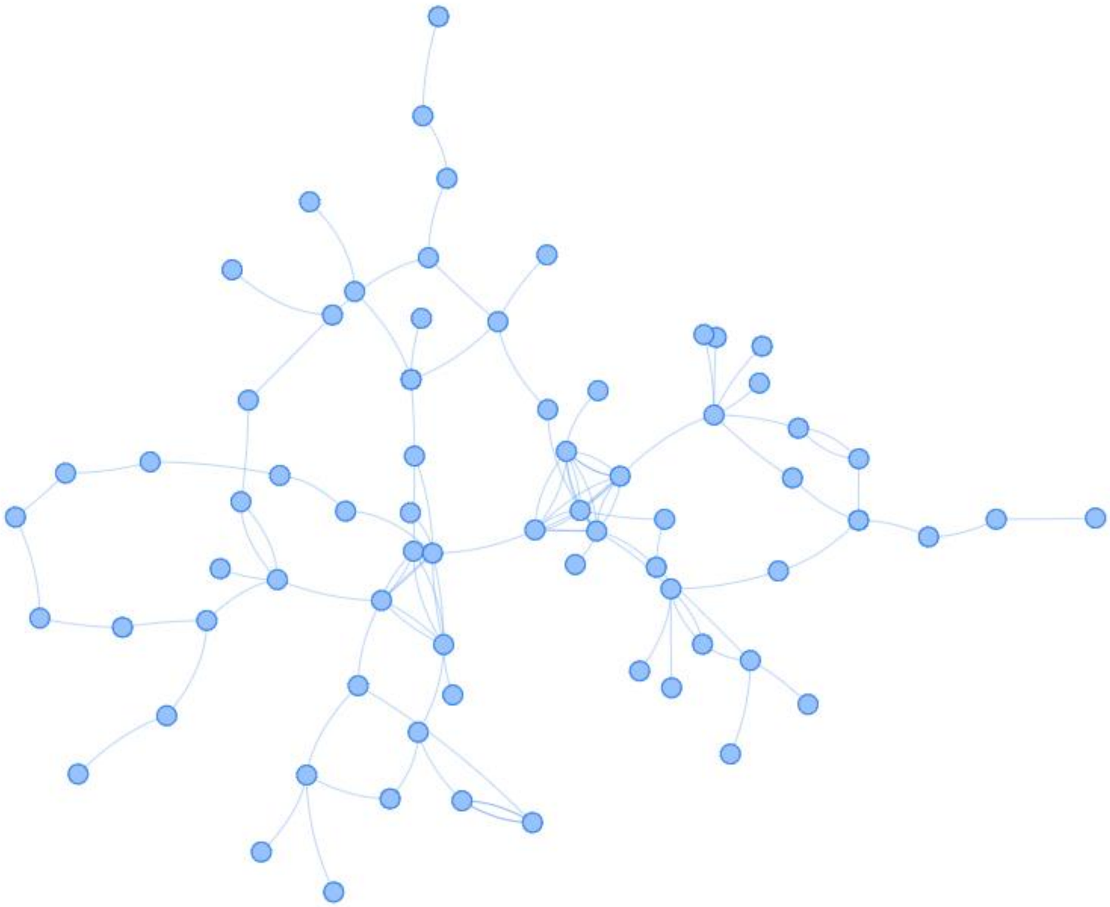

Figure 13 – Resulting node-link graph using a traditional layout with author, citation and topic coupling

Currently all edges look to provide the same type of relation. We can then map colors to edges representing the type of edge it is. A blue edge can represent a key phrase relation. A green edge can represent an author/citation relation. This in turn exposes some of the metadata attributes to the users that can be used as a threshold [Figure 14]. In addition, users can choose the relation path for the next paper to be chosen.

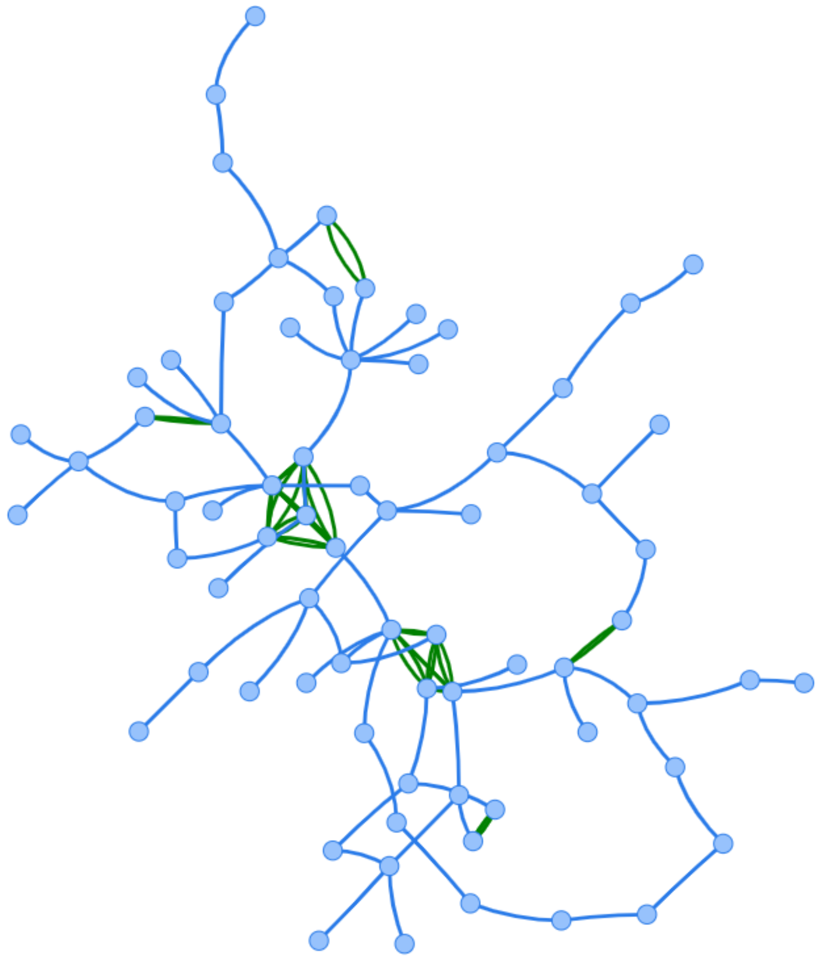

Figure 14 – Resulting node-link graph color highlighting for edges. (Green – author and citation) (Blue – topic relation)

Many bibliographic visualization systems hide metadata from users. Data and information about a paper is extremely useful and should be provided. This information and extra relational data can be provided through click interactions. When a user clicks a node, the node shall be highlighted along with exposed metadata to the user. This data can be the title of the paper, its rank, its authors, and topics related.

A user may want to make a choice of reading multiple papers or base their reading path from a previous paper, so interaction of highlighting related papers in the visualization based on criteria such as related authors and bibliographic coupling is needed to help direct useful papers. The highlighted children of a selected node are the subgraphs of that node. This can be changed for different customizable highlighting to any sub tree depth. This can easily allow for the user to identify potential research within their interest area (Figure 15).

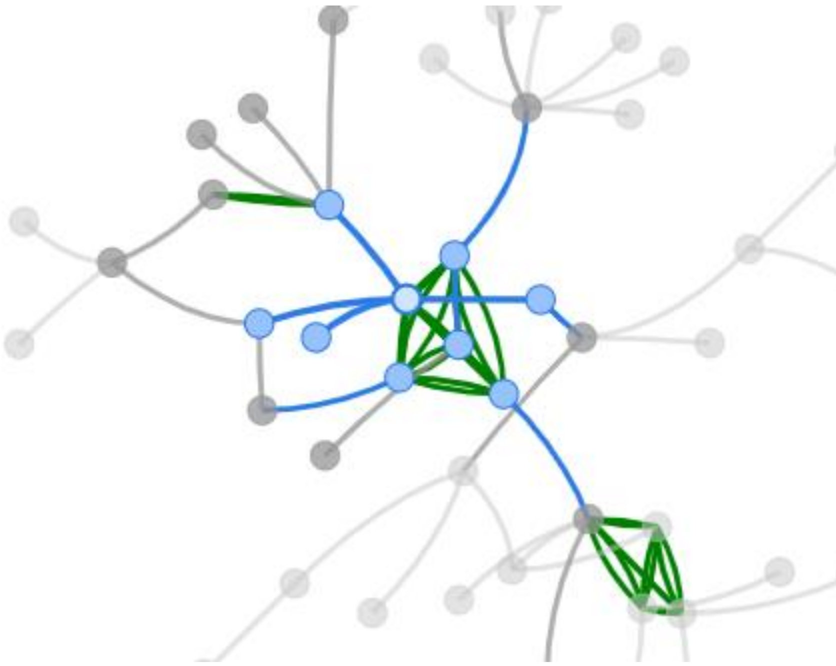

Figure 15 – Highlighting of subtrees for relation recognition

Like mentioned in related works such as HistComp and BiblioViz, only links are associated with relations. Unless the links are labeled or given a color/thickness dictating the edge weight, relations between papers can become a binary view of either not related or related. Even if the visualizations are using thickness and labels do display these weights, it can become too clustered and an unnatural process of comparing thickness across the visualization. Using the SOM topographic map, an x and y value can be associated with each paper. The x and y coordinate are the BMU neurons for each paper creating a relation space. Papers closer together within this 2D space are closer in relation while further away nodes represent the opposite. This method improves upon the binary relation of if its linked/clustered its related to a continuous relation comparison of distances. Carrying over the edges below and applying the positioning, we can have a node-link graph with distant relations along with co-author, coupling, co-topic relations. This provides an extra degree of comparable data to be used.

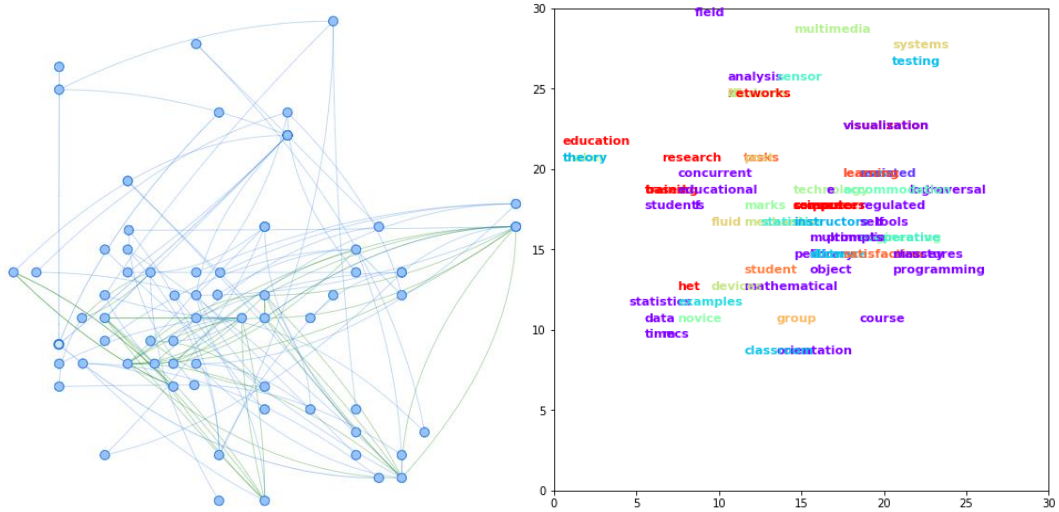

Figure 16 – Comparison of topographic map to node-link graph layout using same coordinates

As you can see, the topographic map from the resulting SOM GloVe model has been implemented with a node-link graph representation. If the node-link graph is zoomed in, the first key phrase from the topic vectors have been used as a label for each node. You can see the nodes key phrase relations based on their distance from each other. Because the SOM performs feature reduction and dynamically changes the topographic map through each epoch, natural clusters of papers are formed.

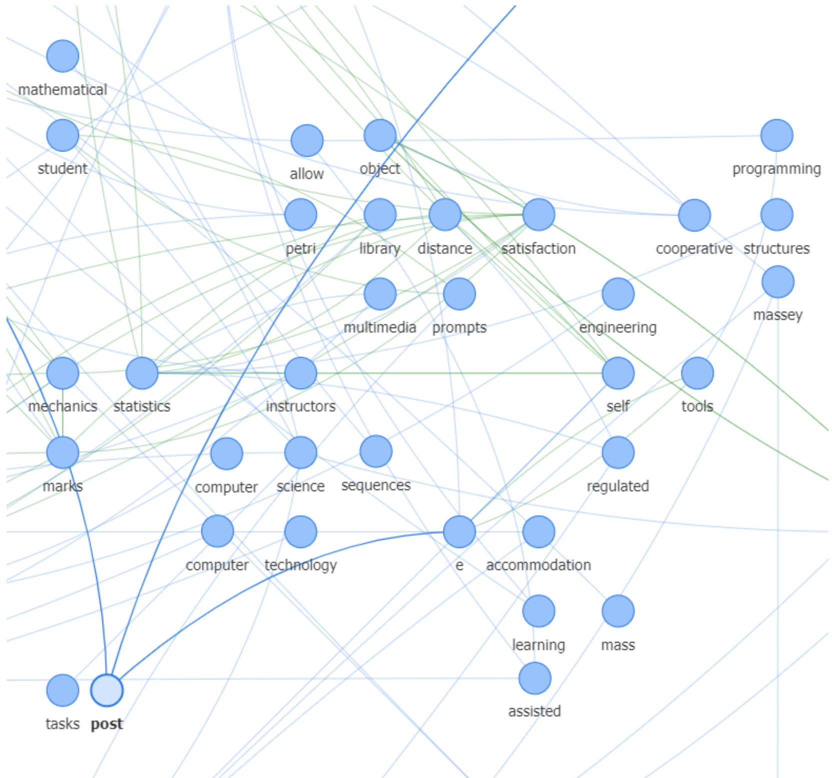

Figure 17 – Zoomed view of new node-link graph with papers labeled as first topic

5.2 3D Plotting

A hard question when visualizing networks is when should 3 dimensions be introduced? Two dimensions can convey a lot of information very quickly and easily, but when the attributes of objects become a large size, two dimensions are not enough. Since theadvisor is a recommendation system that provides the ranking of papers, it should be a significant part of the visualization process. A 2D visualization of the recommendations is good for topics relations to shard metadata and clustering. Once a factor such as rank is presented, a network cannot convey the same information 3D could.

For this work, both visualizations are used together for more information. One that is 2D for the reasons mentioned above, and 3D for ranking each node (Figure 18). This allows for nodes to be structured by quantitative attributes. The x axis can represent citing count, the z axis can represent the author count, and the y axis can represent the paper recommendation rank, and size can represent citation amount. In the current implementation, the x and y axis represent the SOM BMU units while the z axis represents the rank of a paper. Clicking and hovering a node in 3D space should correspondingly highlight the same 2D node from the node-link graph. With two visualization types, double the information can be presented without confusing the user.

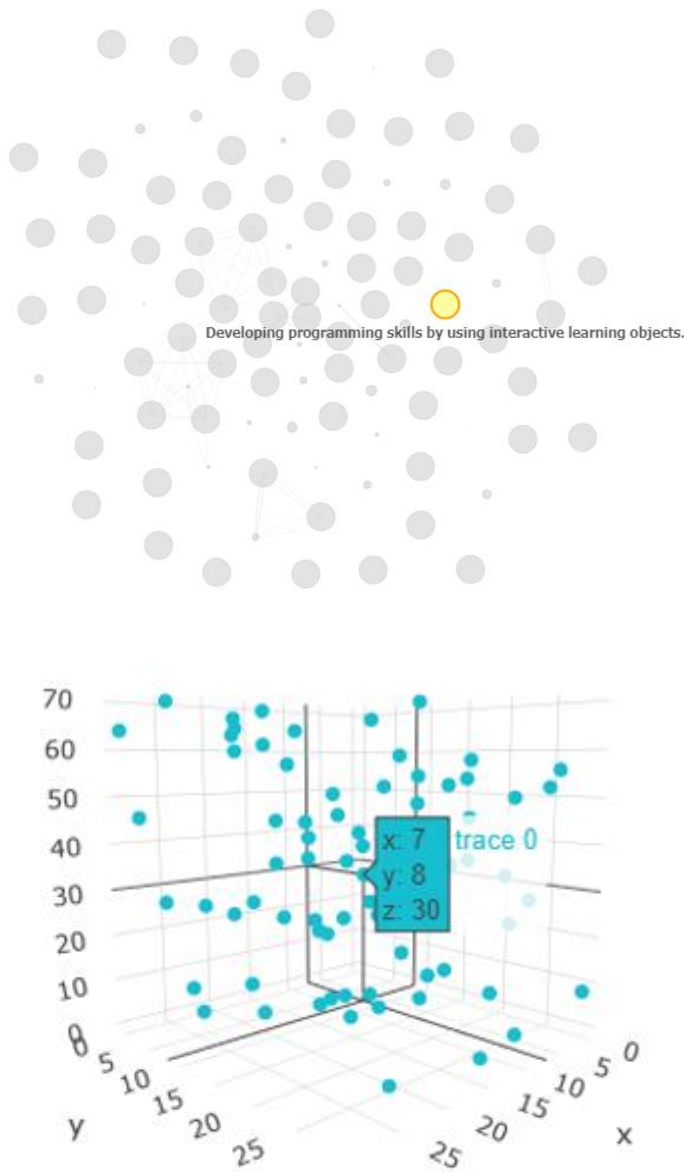

Figure 18 - (top) node-link visual representation and selected node from 3D view (bottom) 3D representation

In BiblioViz's implementation of this, authors were represented as nodes while the top 2D plane connected to those authors. This implementation in comparison argues that author nodes are not as important and such data can be represented as labels rather than nodes themselves. Using the provided 3D view mention below, in combination with SOM results, quick decisions of paper choices can be made. The closer nodes to the top ranked paper or paper chosen within 3D space will be the papers most related for the user to read.

5.3 Conclusion

After many stages in the pipeline added for theadvisor's visualization system, a method for visualizing bibliographic recommendations was achieved. From the beginning, data was received from theadvisor's system and appended with topic vector values. These topics/key phrases were cleaned and supplied as input for two word embedding models: Skip-Gram and GloVe. Each model provided vectorizations of the words in the topic vectors representing relations to context words. These vectors were supplied to a self-organizing map to be clustered together based on the learned associations from the word embeddings the learned associations constructed a topographic map where each input was matched with a BMU neuron and its coordinate. These coordinates were then mapped to a node-link graph to help aid in visualizing relation in a new topic space where only using links can't show as efficiently. The 2D node-link view is also accompanied by a 3D view of the paper data. This view is used to convey more quantitative information while also being tied directly to the 2D visualization through interactions. In comparison to other techniques from BiblioViz, more information and paper relations are formed for quick decision making. In summation, the strengths and weaknesses of these techniques are:

Strengths:

- Exposed paper metadata
- Customizable viewing through parameters
- New spatial relations visualized and formed
- Dual visualization views for more information

Weaknesses:

- More data cleanup
- Multiple data forms for each stage of process
- Longer runtime because of model training
- Doesn’t work without topic vectors for given paper.

Improvements can be made to the visualizations in many ways. Clustering and labeling of the clusters can be achieved by expanding the word embedding models to incorporate a Continuous bag of words model. With this model, the resulting similar words and the topic vectors of each paper can be processed into the model to output topics from their context. Other clustering methods such as k-means and support vector machines can be used to compare efficiency and accuracy. Hyperparameters can be fine-tuned in the word embeddings and self-organizing maps through back propagation to form the most efficient model for the input data.

Three questions were proposed at the beginning of this work. With metadata providing information on ranking, paper title, authors, venue, topics etc. how do these attributes affect the visualization process and how are they useful? The metadata is the most important aspect of a recommended paper because it provides the information to the user. This work created a system for easily visualizing that information in customizable and intuitive ways. Using graph theory, machine learning and visualization techniques, how can such a network of ranked papers be visualized? Incorporating a new way to relate papers in addition to authors and citations can provide a more intuitive way of choosing a paper recommendation. The combination of Word embeddings, SOM and graph structures create a structured system of feeding in recommendations and their key phrases as input and providing efficient relational structures that are more intuitive for a user. How can the networks be formed and structured to convey the data/information in effective ways? The visualizations provide users customizable aspects such as, labels, subgraph highlighting, visible links, top visible papers, thresholding and dual visualizations for easy associations to quantifiable data.

REFERENCES

[1] Wulff, Enrique. (2015). HistComp : bibliographic analysis and visualization of 'The Biological Bulletin'.

[2] Shen, Zeqian & Ogawa, Michael & Teoh, Soon & Ma, Kwan-Liu. (2006). Biblioviz: A System for Visualizing Bibliography Information. 60. 93-102. 10.1145/1151903.1151918.

[3] Iraklis Varlamis and George Tsatsaronis. 2011. Visualizing Bibliographic Databases as Graphs and Mining Potential Research Synergies. In Proceedings of the 2011 International Conference on Advances in Social Networks Analysis and Mining (ASONAM '11). IEEE Computer Society, Washington, DC, USA, 53-60. DOI=http://dx.doi.org/10.1109/ASONAM.2011.52

[4] Westgate, Martin. (2019). revtools: An R package to support article screening for evidence synthesis. Research Synthesis Methods. 10.1002/jrsm.1374.

[5] Zhao, Dangzhi & Strotmann, Andreas. (2015). Analysis and Visualization of Citation Networks. 10.2200/S00624ED1V01Y201501ICR039.

[6] Onur Küçüktunç, Erik Saule, Kamer Kaya, and Ümit V. Çatalyürek. 2013. TheAdvisor: a webservice for academic recommendation. In Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries (JCDL '13). ACM, New York, NY, USA, 433-434. DOI: https://doi.org/10.1145/2467696.2467752

[7] Min Song, "BiblioMapper: a cluster-based information visualization technique," *Proceedings IEEE Symposium on Information Visualization (Cat. No.98TB100258)*, Research Triangle, CA, USA, 1998, pp. 130-136.
doi: 10.1109/INFVIS.1998.729569

[8] Fung, Tsai-Ling & Chou, Jia-Kai & Ma, Kwan-Liu. (2016). A design study of personal bibliographic data visualization. 244-248. 10.1109/PACIFICVIS.2016.7465279.

[9] Habib, Raja & Afzal, Muhammad. (2017). Paper recommendation using citation proximity in bibliographic coupling. TURKISH JOURNAL OF ELECTRICAL ENGINEERING & COMPUTER SCIENCES. 25. 2708-2718. 10.3906/elk-1608-180.

[10] Zhao, Haozhen & Lin, Xia. (2010). A comparison of mapping algorithms for author co-citation data analysis. Proceedings of the American Society for Information Science and Technology. 47. 1 - 3. 10.1002/meet.14504701449.

[11] Klink, Stefan & Ley, Michael & Rabbidge, Emma & Reuther, Patrick & Walter, Bernd & Weber, Alexander. (2004). Browsing and Visualizing Digital Bibliographic Data.. 237-242. 10.2312/VisSym/VisSym04/237-242.

[12] Haydar, Charif & Boyer, Anne. (2017). A New Statistical Density Clustering Algorithm based on Mutual Vote and Subjective Logic Applied to Recommender Systems. 59-66. 10.1145/3079628.3079676.

[13] Portenoy, Jason and Jevin D. West. “Visualizing Scholarly Publications and Citations to Enhance Author Profiles.” WWW (2017).

[14] Sun, Yizhou & Wu, Tianyi & Yin, Zhijun & Cheng, Hong & Han, Jiawei & Yin, Xiaoxin & Zhao, Peixiang. (2008). BibNetMiner: Mining Bibliographic Information Networks. 1341-1344. 10.1145/1376616.1376770.

[15] Nguyen, Thuc & Do, Phuc. (2017). Managing and Visualizing Citation Network Using Graph Database and LDA Model. 100-105. 10.1145/3155133.3155154.

[16] Fekete, Jean-Daniel. (2009). Visualizing Networks using Adjacency Matrices: Progresses and Challenges. 636 - 638. 10.1109/CADCG.2009.5246813.

[17] Yan, Erjia and Ying Ding. “Scholarly network similarities: How bibliographic coupling networks, citation networks, cocitation networks, topical networks, coauthorship networks, and coword networks relate to each other.” JASIST 63 (2012): 1313-1326.

[18] Newman, M.E.J.. (2001). Scientific Collaboration Networks. II. Shortest Paths, Weighted Networks, and Centrality. Physical review. E, Statistical, nonlinear, and soft matter physics. 64. 016132. 10.1103/PhysRevE.64.016132.

[19] Derek J. deSolla Price "Networks of Scientific Papers: The pattern of bibliographic references indicates the nature of the scientific research front" Science 149(3683): 510-515, July 30, 1965

[20] Harris, David and Harris, Sarah (2012-08-07). *Digital design and computer architecture* (2nd ed.). San Francisco, Calif.: Morgan Kaufmann. p. 129. ISBN 978-0-12-394424-5.

[21] Kaski S. (2011) Self-Organizing Maps. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA

[22] Mikolov, Tomas et al. “Efficient Estimation of Word Representations in Vector Space.” CoRR abs/1301.3781 (2013): n. pag.

[23] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation.