Visualizing Similarities between American Rap-Artists based on Text Reuse

Christofer Meinecke*‡, Jeremias Schebera*‡, Jakob Eschrich† and Daniel Wiegreffe*

*Image and Signal Processing Group, Institute for Computer Science, Leipzig University, Leipzig, Germany
E-mail: {cmeinecke,schebera, daniel}@informatik.uni-leipzig.de

†Institute for Computer Science, Leipzig University, Leipzig, Germany
E-mail: je17rady@studserv.uni-leipzig.de

‡Equal contributors

***Abstract*—Rap music is one of the biggest music genres in the world today. Since the early days of rap music, references not only to pop culture but also to other rap artists have been an integral part of the lyrics' artistry. Rappers may use them to introduce their shared personal backgrounds such as where they grew up. In addition, rap musicians reference each other by adopting fragments of lyrics, for example, to give credit. This kind of text reuse can be used to create connections between individual artists. Due to the large amount of lyrics, only automated detection methods can efficiently detect text reuse. In addition, automated methods can also be used to identify similar artists based on their lyrical content. Here, we present a visualization system for analyzing text reuse in rap music lyrics. The system supports the user of detecting text reuse and allusions between songs and exploring connections between artists. For this purpose, we crawled song lyrics and their metadata of selected American rap artists from Genius.com. We also trained a network tailored specifically for rap lyrics, which we named "rapBERTa", to compute similarities in lyrics.**

***Index Terms*—Text Reuse, Intertextuality, Visualization, Rap Music**

I. Introduction

Rap music started as a way for marginalized groups to express their social and economical struggles rhythmically and poetically. In the early years after its inception, the genre stayed mostly within the borders of its corresponding subculture. But in the 80s, with the emergence of "gangsta rap" through groups like N.W.A and artists like Snoop Dogg or Dr. Dre, rap music made its breakthrough into the mainstream [1], [2]. Today, it is one of the biggest music genres with its influence spanning across the globe [3]. Since rap music's early days, references to pop culture but also to other rap artists have been an integral part of its lyrical craftsmanship. Rappers may share personal connections through their backgrounds like the city or neighborhood they grew up in or even gang affiliation. Because of these relations they often reference similar themes, places, or culturally specific phrases. Rivalries have also always played a big part. Controversies between formerly affiliated rappers like the members of the group N.W.A, rappers being affiliated with different gangs, or rivalries spanning the whole genre like the East Coast vs. West Coast clash in the 1990s often result in so-called "diss tracks". In these, the musicians mock each other, often re-using or referencing their adversary's lyrics to use against them. More positively, artists sometimes re-use other musicians' phrases to pay homage to them and their lyrical craftsmanship, be it out of mutual respect of two contemporary artists or in the effort of a newer artist to allude to the ones that inspired them [2].

Yet, an issue that arises with anything connected to commercial success is plagiarism [4]. Websites like Genius.com [5] offer annotated song lyrics while services like Spotify [6] and Soundcloud [7] provide access to millions of songs on demand. Because of tools like these, discovering music has never been easier. This easy access combined with the promise of financial success achievable through rap music may lead aspiring artists to plagiarize successful ones in the hopes of garnering attention. Because of the sheer amount of lyrical content, automated means of detecting text re-use can help find cases of plagiarism. Furthermore, these automated procedures can also be used to identify similar artists based on their lyrical content. This data may then be utilized to help fans of the genre to find new artists similar to the ones they already enjoy. References to other artists as well as commonly used phrases could be retraced to their origin, enabling those interested in rap music to deepen their knowledge.

This work is an extension of [8]. We present a visualization system that enables the user to explore similarities between artists, detect cases of plagiarism and allusions between songs, and discover new artists or songs. Therefore, we crawled lyrics and their metadata of selected American rap artists from Genius.com. These lyrics are then embedded using RoBERTa [9] in order to compute similarities between them. All artists, songs and lines are then saved as nodes in a graph into a Neo4J database. First, we give an overview of related work (Section II) and define the tasks for the application (Section III). After that, the collection of the data (Section IV) and the text alignments of the song lines (Section V), which form the basis for the application, are discussed. In Section VI, the components of the application are described in detail, and in Section VII, some connections to the real world seen in the visualizations are discussed. Finally, the results are examined (Section VIII) and a summary is given (Section IX).

LEVIA'22: Leipzig Symposium on Visualization in Applications 2022

II. Related Work

A. Visualize Artist Similarity

Several previous works have utilized visualization to compare artists and to discover new artists based on similarity [10]. Tools like the ArtistExplorer [11] and Music-Map [12] calculate artist similarity based on user statistics from platforms like Spotify [6]. Musicians with a bigger overlap between listeners are treated as more similar. Both utilize graph-based visualizations to communicate relations between artists but do not explore the content of the artists' lyrics. Other works base the artist's similarity on reviews. Cano and Koppenberger [13] crawled data from Allmusic.com while Gleich et al. [14] use music reviews from Yahoo to generate a densely connected graph of similar musicians. Additionally, Cano and Koppenberger [13] search playlists for co-occurring artists to calculate similarity. Similarly, Schedl et al. [15] rely on the co-occurrence of musicians on websites to generate lists of prototypical artists for different genres in addition to a graph. The "History of Rock in 100 Songs" [16] visualization takes an approach not solely based on textual data. Instead, it analyzes songs' valence (musical positiveness) and energy (intensity and activity). Furthermore, visualizations to show plagiarism are designed by Ono et al. [17] and De Prisco et al. [18].

Other works focus on the lyrics to visualize the size of rappers' vocabulary [19] or the used words and their frequency [20]. Similar to us, Meinecke et al. [21] use Genius data for an automated semantic analysis of songs to generate similarities between artists and to explore their lyrics with several visualizations. However, in order to compute similarities between song lines they use fastText [22], which only provides word vectors. Furthermore, their models are trained on Wikipedia and Urban Dictionary [23] and are not fine-tuned on lyrics. In order to calculate embeddings for whole song lines, additional steps have to be taken. The transformer models used in this work on the other hand are both fine-tuned on the task of semantic textual similarity and natively produce sentence vectors. Additionally, one model is specifically trained on the collected corpus of rap lyrics to include domain knowledge of this specific task.

B. Text Alignment

The goal of an alignment is to find similar and diverging patterns within two or more data objects of the same type. Yousef and Jänicke [24] differentiate between three tasks that can be supported by text alignment, two of which are relevant to this work: collation and text re-use detection.

Collation is the process of comparing and analyzing different variants of the same text based on similarities and differences in their wording. Finding where sentences with the same or similar meaning diverge regarding the exact words used and where they are the same can easily be done by a computer. Yet, only a human expert can analyze these collation results and infer knowledge from them. Therefore, visualizations can help to aid humans in this task.

Text re-use is a broad term that covers many ways of copying the content of one passage of text to another. The most direct form of this is copying the text word for word, but paraphrasing, allusions and even summarization fall under the term of text re-use. Automatic approaches can be utilized to calculate a similarity score between sentences, producing alignment pairs. Aided by visualizations, a human can then analyze the automatically generated pairs to determine cases of plagiarism or other text re-use scenarios.

According to the text alignment visualization survey of Yousef and Jänicke [24] the most popular visualization methods for both tasks are Side-by-side Views, Aligned Barcodes and Variant Graphs. Our prototype application aims to find similar artists by detecting possible occurrences of text re-use in their song lines. To visualize these occurrences we utilize a combination of Side-by-side Views and Aligned Barcodes that aid in pairwise collation as well. For the collation of more than two similar lines, which can be seen as text variants, we use Variant Graphs [25].

III. Tasks

The application is designed for users of the general public interested in rap music. The aim is to offer a tool that supports an exploratory analysis of selected American rap musicians and their lyrics. Therefore, the following three levels of tasks (with corresponding sub-tasks) were developed by the authors for the design of this application:

1. **Analyze Artists:**
   - 1.1. **Find similar artists:** As someone generally interested in American hip hop, a user could reasonably want to discover artists similar to those they are already familiar with or even fond of.
   - 1.2. **Explore an artist:** Knowledge about the artists background, may give the user context for similarities between artists or potential references.
   - 1.3. **Compare different artists:** A user familiar with American hip hop might want to explore groups or pairs of artists they already deem similar from listening to their music and infer which songs and lines are the closest thematically. By doing this, it is possible to find artists directly referencing the other or possibly even copying a particularly witty verse. On top of that, looking at artists that emerged in the same time period the user could discover certain trend words or phrases from that time period and even if the meaning of the phrase or word has evolved over time.
2. **Analyze Songs:**
   - 2.1. **Find similar songs:** Users may also be interested in finding lyrically similar songs to their favorite song or a song of interest.
   - 2.2. **Explore a song:** A user with prior knowledge of influential artists and songs could explore those songs' influence by searching for other songs that reference specific lines. The other way around, commonly used phrases could be traced back to their origins within hip hop.

2.3. **Compare different songs:** When a user found a song of interest they could be interested in comparing the song to other ones of different artists.

3. **Find similar lines:** A user could be interested to find lines that are similar to a line of interest and also to find all occurrences of a line across the whole song corpus.

The described tasks follow a finer and finer order from artists to songs to lines. Thereby, Tasks 1. , 2. & 3. (and their sub-tasks) can each serve as an entry point for one another, but can also be treated separately in the tool. It should also be mentioned that general attention was paid to compliance with the "Visual Information Seeking Mantra" of Shneiderman [26].

IV. Data & Preprocessing

We collected song lyrics and metadata about rap artists from Genius.com [1] (henceforth referred to as "Genius"), which describes itself as a website for "song lyrics & knowledge" with a focus on hip hop and pop music. Other than the lyrics themselves, contributors can also provide annotations including but not limited to: possible interpretations of certain lyrics and explanations of references to pop-culture and the artists' personal life. On top of that, Genius provides meta-information such as featured artists, release dates, labels under which a song was released, etc. We compiled a list of 219 American hip hop artists based on popularity and personal interest to gather Genius data for. Genius provides an API, allowing applications registered with their API Client management page to fetch data from Genius' database. We collected data on each of the artists' most popular songs up to a maximum of 200 songs per artist. Thus, our database includes a total of 25,654 songs with 1,598,466 lines. The data contain information about the artists, like their name, a short description, and the artists' songs including their lyrics. The lyrics were lowercased, all the punctuation and special characters were removed.

Since relationships and similarities are our main focus, we used a Neo4j [2] graph-database to store artists, songs and lines as nodes. We focus on textual alignments between individual lines to establish connections between songs and artists. For this, the text was split up, so that each individual line in a song is represented by its own node in the database. Beyond the lyrics, the aforementioned annotations were used to enrich the line-nodes with information about which part of a song they belong to and who they were performed by. To preserve the order, each line-node also gets an index according to their position within the song. We connect these line-nodes with similarity relationships based on the findings of our search for textual alignments. Each song is represented by a node as well, containing information about the title, release date, associated album, featured artists, etc. Line-nodes are connected to their respective song-nodes via a "part-of" relationship. Thanks to this, it is later possible to calculate song-level similarity and explicitly connect songs through similarity relations. Finally, the same is done for the artists, as those too are represented by their own nodes containing their name, description, alternate names, etc.

[1]https://genius.com

[2]https://neo4j.com

V. Textual Alignment

In order to find lines that are semantically similar, we used RoBERTa [9]. The model takes a word or even a string of words as an input and produces an embedding vector representing the semantic meaning. We utilized two versions of RoBERTa, one ready-to-use version specifically fine-tuned on the task of semantic textual similarity called 'stsb-roberta-base' [3], and the same network additionally fine-tuned on our collected corpus of rap lyrics which we gave the name 'rapBERTa'. The reason for this additional training is the heavy usage of slang, neologisms and pop culture references in hip hop. The hypothesis was that in learning rap-specific language, rapBERTa may also perform better in finding meaningful semantic similarities in a corpus of rap lyrics.

Sentence embeddings were produced by the models for each individual line of the cleaned corpus and were indexed using faiss [27] for efficient similarity search. The index was used to find the 15 nearest neighbors for each line i.e. the most similar lines within the corpus based on cosine similarity. The resulting neighbor relations between song lines are then added to the Neo4j Graph-Database with their corresponding similarity value as "neighbor of" relationships. This forms the basis for the entire application.

A. Artist Similarity

The user should be able to discover similarities between artists (Task 1.1.) in a graph connecting each artist to the one most similar to them. Thus, it is necessary to calculate a similarity score between artists based on the relationships between their respective song lines. Two values are important in order to calculate an artist-to-artist similarity score; one being the amount of "neighbor of" relationships found between their respective song lines, and the other being the line to line similarity scores in those relationships.

Not all artists have the same amount of songs stored in the database and therefore the amount of lines for each artist varies as well. Especially newer musicians may not have recorded that many songs or not yet have had all of their songs added to Genius by users. To account for this fact, the relative amount of "neighbor of" relations from artist a_i to a_j ($p_{a_{ij}}$) is used for the calculation of artist-to-artist similarity metrics. Of course there is also the counterpart from the artist a_j to a_i where the relative amount of "neighbor of" relations is defined by $p_{a_{ji}}$.

The second component to calculate artist-to-artist similarity is the cosine similarity of the lines connected by the "neighbor of" relationships. Intuitively, if there are two pairs of artists with the same relative amounts of similar lines between them, the artist pair with the higher average line-to-line similarity is closer. For the remainder of this section, the average line-to-line similarity between two artists shall be referred to as sim.

[3]https://huggingface.co/cross-encoder/stsb-roberta-base

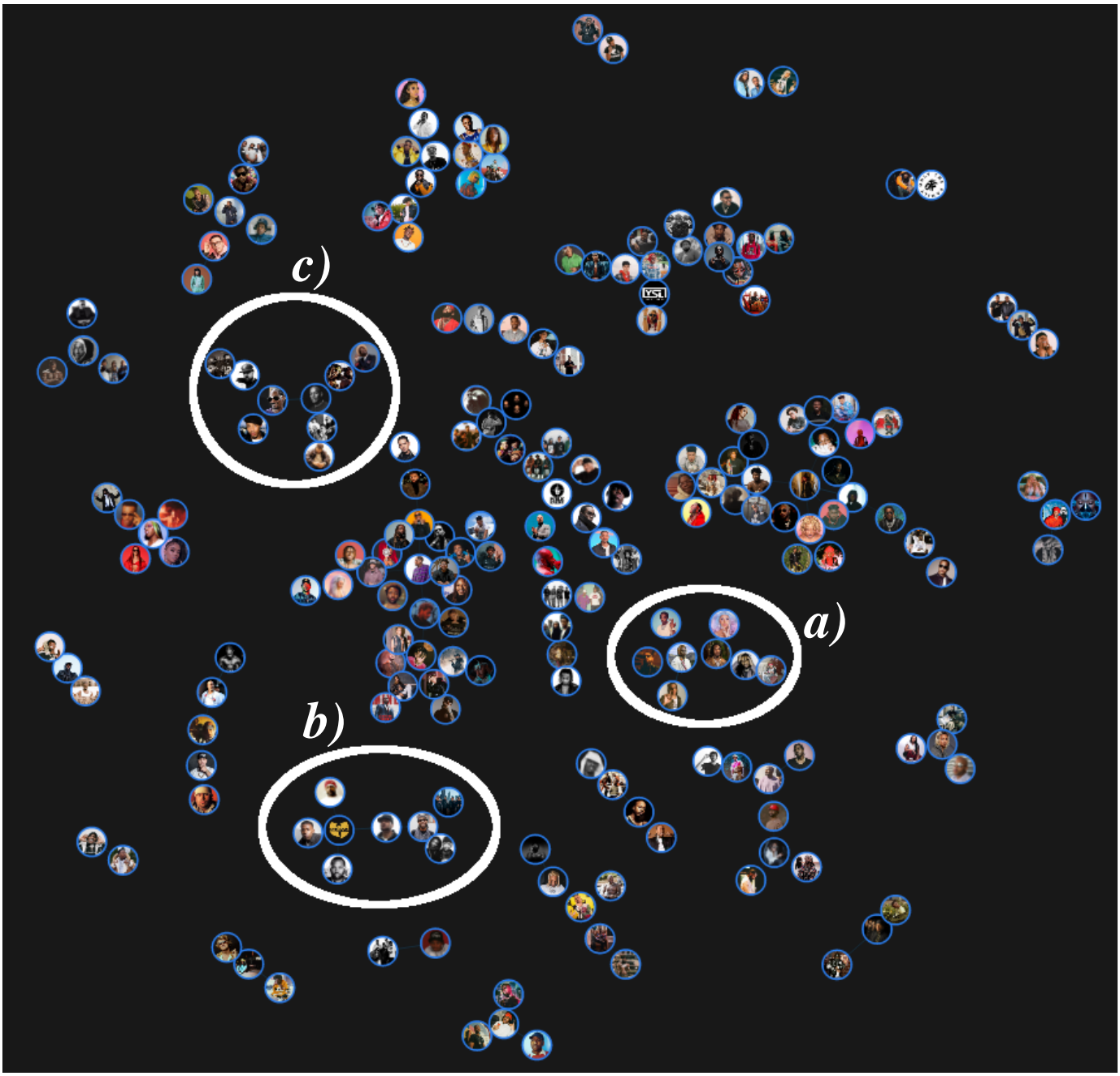

Fig. 1. The artist graph, artists that are similar based on their lyrics are connected. Different kinds of clusters can be observed. **a)** Shows one subgraph with a cluster containing Atlanta based rappers Offset, Quavo and Take-off. **b)** Shows a subgraph containing the artists Raekwon, Ghostface Killah, Method Man, Redman, and GZA, all part of the Wu-Tang Clan. **c)** Shows N.W.A members Dr. Dre and Ice Cube together with several artists connected to them.

There are different viable approaches to calculating an artist-to-artist similarity score. One is a weighted sum, which takes both the average line-to-line similarity and the relative amount of similar lines into account while also allowing for a weight to be assigned to each of them. Through this weight, it can be adjusted how big of an influence the corresponding component will have on the final artist-to-artist similarity score. Further, the weighted sums are normalized by dividing them by the sum of weights. With this method, two directed similarity scores can be calculated between two artists. While this may seem unintuitive at first, it is simply due to the fact that $p_{a_{ij}}$ and $p_{a_{ji}}$ indicate how much of an artist's entire corpus of song lines the found similarities account for. The directed similarity score for an artist a_i to another artist a_j is calculated as follows:

$$\frac{w_p \cdot p_{a_{ij}} + w_{sim} \cdot sim}{w_p + w_{sim}} \tag{1}$$

Another approach to calculating an undirected artist-to-artist similarity score is to take the minimum of $p_{a_{ij}}$ and $p_{a_{ji}}$ and

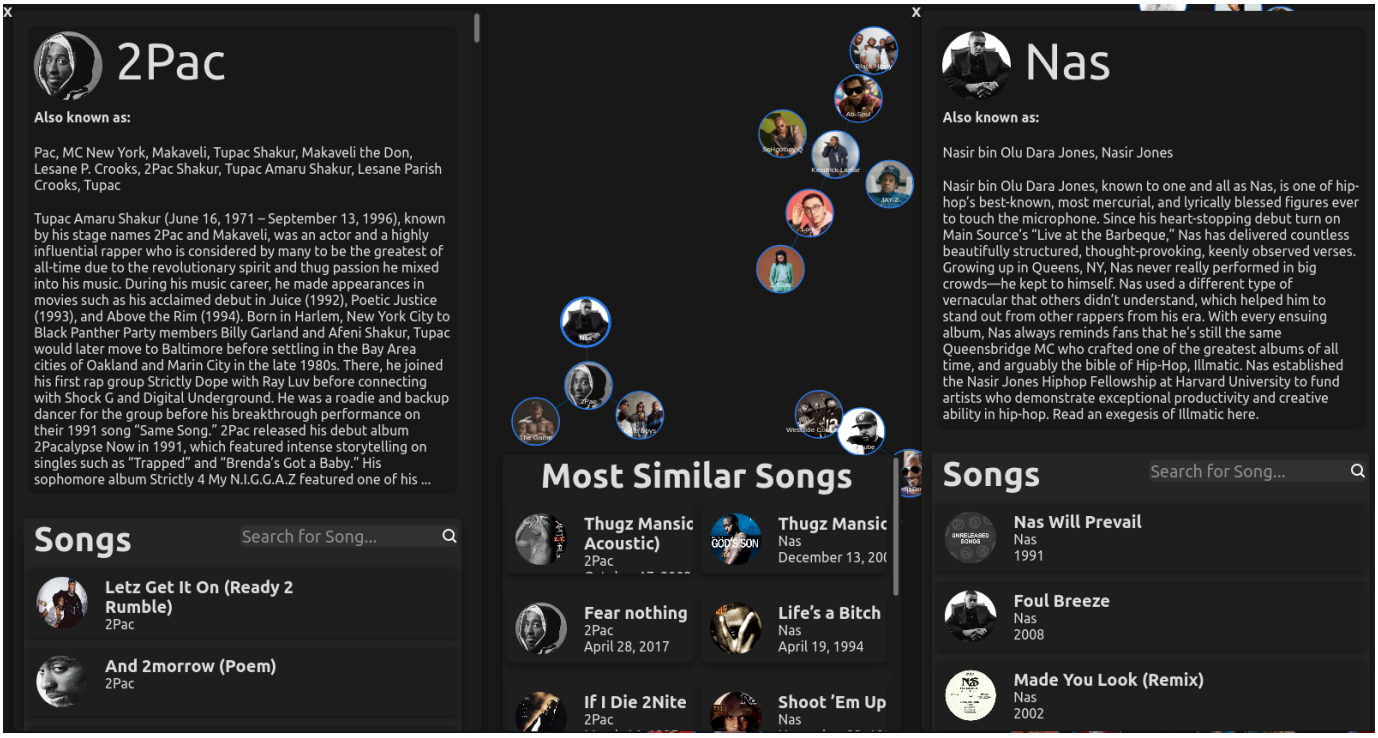

Fig. 2. The artist view of 2Pac and Nas shows biographical information, a list of the songs and the most similar songs of both artists.

combine that with sim instead of constructing an average over two directed similarity scores.

$$min(p_{a_{ij}}, p_{a_{ji}}) \cdot sim \tag{2}$$

The rationale behind using the minimum of $p_{a_{ij}}$ and $p_{a_{ji}}$ is that a high $p_{a_{ij}}$ indicates that a large number of artist a_i's lines are similar to lines of artist a_j. This is, however, not enough to indicate that both artists closely resemble each other. It could simply mean that artist a_i re-uses themes a lot that artist a_j only features in some of their songs. If a $p_{a_{ji}}$ that is smaller than $p_{a_{ij}}$, however, results in a similarity score between artist a_i and artist a_j that is higher than the ones between artist a_i and any other artist or artist a_j and any other artist, it makes a strong case for the close relationship between them. This approach is the one that is used in the application as it subjectively yielded the most interesting connections.

VI. Visual Interface

The user can explore the acquired data through a web application that provides several visualizations to aid in discovering patterns and relationships.

A. Artist Graph

The *artist graph* shown in Figure 1 represents the core of the application. This force-directed graph layout provides the user an overview of all the artists in the database and their similarities (Task 1.1.). For the graph layout the library 'vis.js' [4] is applied, which uses the "Kamada Kawai algorithm" [28] for the initial layout and the "Force Atlas 2 algorithm" by Jacomy et al. [29] for the final layout. Each artist is represented by a circle containing an image of the artist. An edge between two artists indicates that they are the most similar based on their lyrics. This leads to the formation of subgraphs consisting of lyrically related artists. Additionally, the length of an edge represents the value of the similarity score. Through this, denser clusters within those subgraphs manifest, indicating even more closely connected artists. The connections between the artists within the subgraphs and the spatial closeness of artists within the clusters help the user quickly identify groups of similar artists. With this baseline of information, the user can then explore the lyrical connections of artists within these groups (Task 1. & 2.).

B. Exploring and Comparing Artists' Lyrics

From the artist graph, the user can select an artist by double-clicking their image. This opens a popup (*artist view*) containing information about the artist and a list of their songs, which supports Task 1.2. ("Explore an artist"). Upon selecting the first artist, their corresponding artist view opens on the left side. Selecting an additional artist will open a second artist view on the right side, which can be seen in Figure 2. As both of these artist views are shown together with the Artist Graph, the user never loses context, as they can still see the area of the graph they were exploring. Any subsequent selections of an artist will change the right-hand artist view to display the newly selected artist, while the left-hand artist view stays the same. At the top of the artist view, the user can find a short text about the artist which was collected from Genius along

[4]https://visjs.org

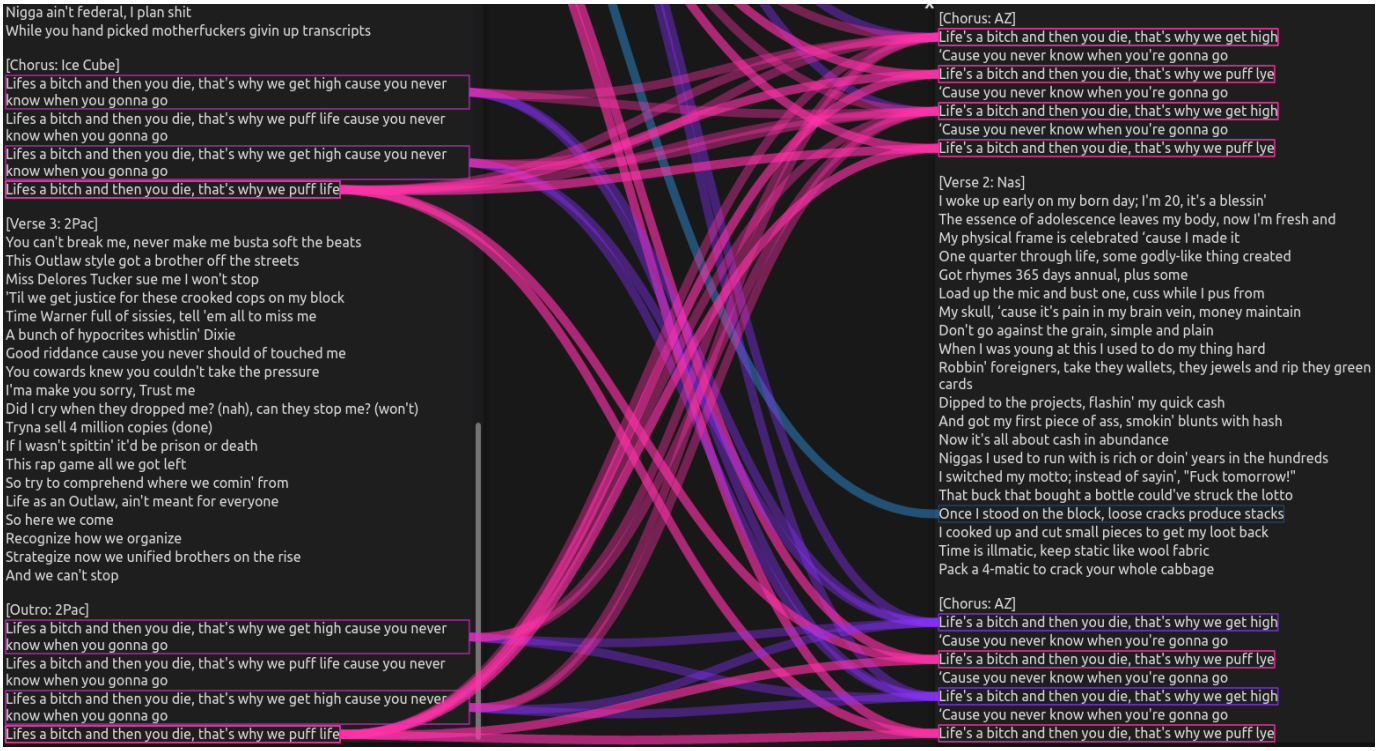

Fig. 3. Side by Side View of the songs "Fear Nothing" by 2Pac and Ice Cube and "Life's a Bitch" by Nas and AZ. The former song reused the chorus of the later song. Each group of lines that are similar to each other is assigned a unique color so that the user can easily distinguish them.

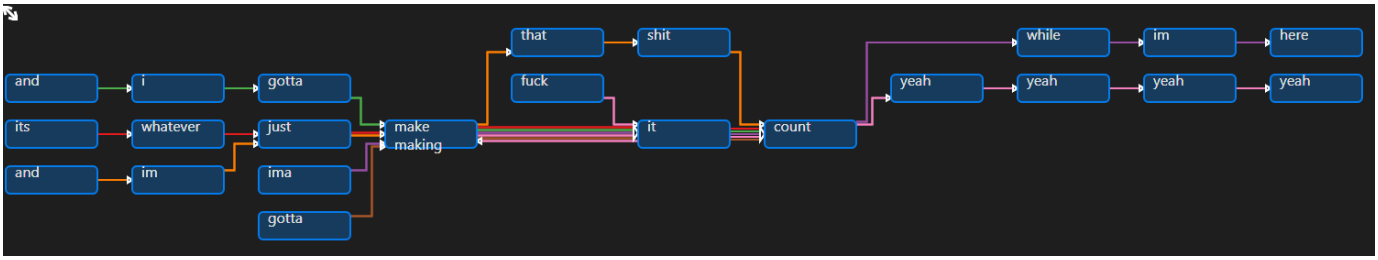

Fig. 4. The Text Variant Graph, the different lines are color coded and shared words are merged into one node.

with the other data. These descriptions offer knowledge about the artists background, giving the user context for similarities between artists or potential references.

Opening two artist views offers the first methods of direct comparison (gives an entry point too Task 1.3. "Compare different artists"). An additional popup will appear in the middle between the two artist views, showing pairs of the artists' most similar songs. This also allows to support Task 2.1. ("Find similar songs"). Selecting one of these pairs will open a *song view*, in place of their corresponding artist view. If the user wants to compare two specific songs (Task 2.3.), the artist view also allows them to search for and select any of the corresponding artists' songs in the database. Whenever there are two song views open at the same time, all their songs' textual alignments are shown via a visualization called *Aligned Barcodes*. All pairs of similar lines are marked and connected by colored Bézier curves (Task 3.). This visualization can be thought of as a graph, where the song lines are vertices with edges connecting them to similar song lines. A group of lines that are all similar to each other form a subgraph. Each of these subgraphs has its own color, so the user can easily differentiate between the groups. The colors are equidistant in regard to their hue but the same in saturation and brightness. This color scheme was chosen to highlight that the groups are qualitatively different. Figure 3 shows "Fear Nothing" by 2Pac and Ice Cube and "Life's a bitch" by Nas and AZ. The former reused the chorus of the later song.

Each song remains individually scrollable, so different parts of both songs can be compared and explored (Task 2.2. & 2.3.). Each song view offers further options to explore the data. By clicking on the artist name at the top of the song view, the user can go back to the artist view to compare another of the artists' songs to the one still open on the other side. It is also possible to not just compare two artists, but use one specific song as a

starting point to traverse the data (Task 2.2.). If the user wants to find references to a song, opening only one song view makes a list of similar songs appear in the middle of the screen. To get even more specific, each individual line of a song view is clickable. Selecting one line opens a list of all similar lines from other songs on the opposite side of the screen (Task 3.). This enables the user to explore the usage of certain phrases between different artists and possibly trace who is referencing who. Additionally, a visualization is provided that aids in the comparison of all the similar lines which also supports Task 3. ("Find similar lines"). It is an adaptation of a Text Variant Graph and can be seen in Figure 4. Each word, or group of words that the language model deemed similar meaning, is represented by a box. Colored arrows connect the boxes to form the sequences of words as they appear in the song lines. Each path of one color represents one song line. The song line selected as consensus is displayed in the center as a sequence of nodes aligned horizontally on a line (red edges). Thus, the sequences diverge where the choice of words differs between the lines, and converge where the chosen variant words are the same. Having found a particularly interesting line similar to the one originally selected, the user has the option to click on it in the list. Thus, the list of similar lines is replaced by the song view corresponding to the clicked-on song line, once again enabling the comparison of the two songs. Furthermore, a user can do a fulltext search for a specific song name or for occurrences of a specific line. For this, the Neo4J fulltext search returns exact matches and also approximate matches. An example for the famous quote "Each one teach one" can be seen in Figure 5.

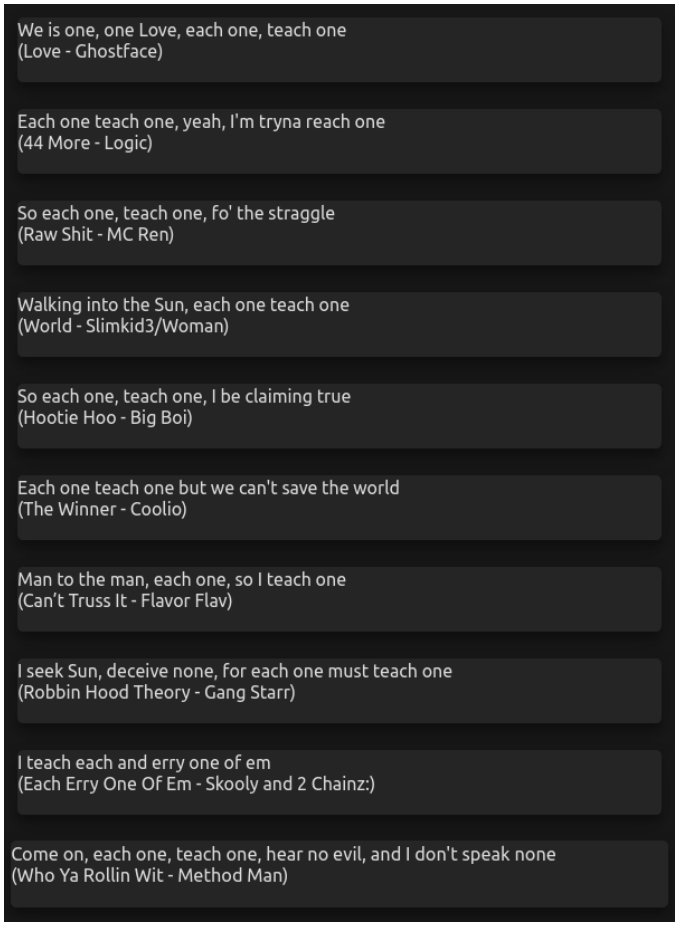

Fig. 5. Top search results for the phrase "Each one teach one".

VII. Real world application

Taking into account knowledge about the individual artists, their style and history, it becomes apparent that the Artist Graph does show meaningful connections. Not only can we observe subgraphs of artists that share thematic and even stylistic similarities, but sometimes even clusters within those subgraphs that point to a deeper connection between artists. Figure 1 a) shows one such subgraph with a cluster containing Atlanta based rappers Offset, Quavo and Take-off. As the graph shows, these three are quite closely related lyrically. This makes sense, because they are also related in the literal sense and form the rap trio known as 'The Migos'. We can also see a close connection between Offset and Cardi B, who are married in real life and thus regularly feature on each other's songs. Three of the other rappers in this subgraph are also based - or at least born in - Atlanta. Similarly, Figure 1 b) shows a subgraph containing the artists Raekwon, Ghostface Killah, Method Man, Redman, and GZA, all part of the Wu-Tang Clan, which is also part of the subgraph. The additional artists featured in the subgraph; Cypress Hill and Heltah Skeltah, also emerged in the same time period as the Wu-Tang Clan, around 1990. Furthermore, apart from Cypress Hill, they all come from New York, influencing and being influenced by 1990s era Eastcoast-HipHop. Figure 1 c) shows N.W.A members Dr. Dre and Ice Cube together with several artists connected to them. Including Snoop Dog and Warren G two artists that collaborated with Dr. Dre and groups where Ice Cube was a member of.

VIII. Discussion & Future Works

The approaches to calculate the artist similarity have different advantages and drawbacks. The average weighted sum enables the assignment of different levels of importance to favor either the amount of similar lines or their average similarity. Assigning bigger weight to the average similarity favors the connection of artists that quote each other directly, or use the same wording for other reasons. For the goal of detecting plagiarism or direct references, this behavior might be desirable. Yet, in many cases, this approach just connects artists that repeat the same or similar phrases multiple times throughout their songs meaningful connections that could be observed using the minimum as a similarity score are not present in the graph. Assigning a bigger weight to the amount of similar lines favors connections between artists that often use thematically similar lines, even if the likeness of these lines is not that big. Some groupings shown previously are present with this score as well. For the cluster in Figure 1 a) the Wu-Tang Clan would be still connected to some of its members, however, the connections to other rappers from the

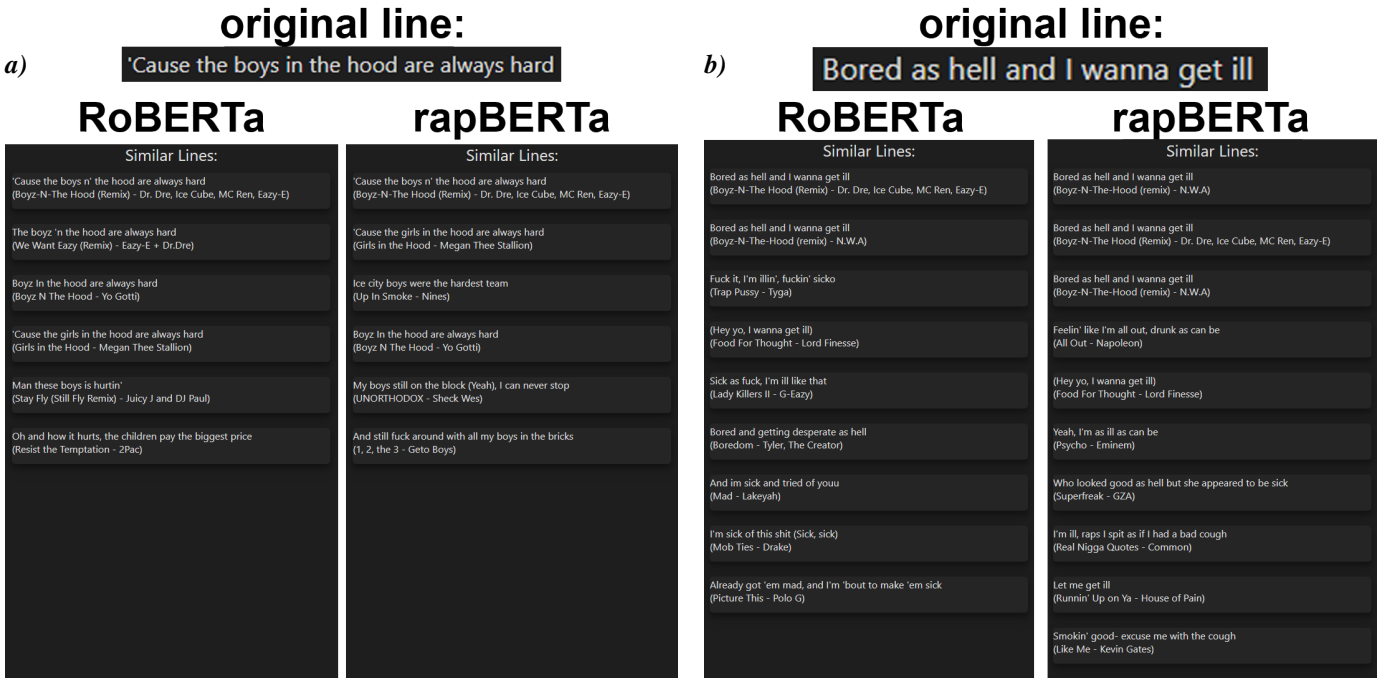

Fig. 6. The most similar lines based on the two different models for the line **a)** "Cause the boys in the hood are always hard" and **b)** "Bored as hell and I wanna get ill".

same era would be missing. Similarly, the cluster in Figure 1 b) would only show Atlanta-based rappers Offset, Gunna, Lil Gotit and the connections to the other Migos members Takeoff and Quavo are not present. Yet, an argument can be made that this could lead to the discovery of similar artists that do not share immediately obvious real-life connections.

Some "songs" collected through the Genius API were in fact not songs. Such oddities include body maps of artists' tattoos and recipes. 2013, rapper 2Chainz released a cookbook, which is listed on Genius as an album of his. Thus, some recipes from his book exist in the Genius database and are treated just like regular songs. All of this is a byproduct of the crowdsourced nature of Genius.

Exploring the data has made clear that the lyrical nature of rap sometimes poses a problem for the models' understanding of song lines. While handling lines that contain words found in the dictionary well, text passages that make use of neologisms and slang are prone to misinterpretation by the models. It can be observed that RoBERTa often succeeds in finding lines with similar meaning as the first few nearest neighbors. However, not all neighbors always match well. Additionally, it seems that rapBERTa has fewer problems in understanding words that are not meant literally based on their context. Examples can be seen in Figure 6. Figure 6 a), shows how "hood", "block" and "bricks" can be used interchangeable and in Figure 6 b), "ill" means drunk which becomes apparent when reading the following lines. Based on the similar lines found by RoBERTa and rapBERTa, it appears that rapBERTa has at least partially learned this meaning while RoBERTa only knows the word literal meaning.

Yet, it still finds semantic similarities where there are none. In many cases, this can be attributed to the fact that the surrounding lines have to be taken into account to understand one line's meaning. Additionally, considering the amount of data that the standard model of RoBERTa is trained on to achieve such high scores on the Semantic Textual Similarity Benchmark [30], the corpus that rapBERTa was additionally trained on is very small. Training it on a much larger corpus of rap lyrics may yield better results, as its understanding of the specific slang, neologisms and pop culture references utilized in rap music will improve. Moreover, employing an approach where the context that the model can use to learn the meaning of words is not limited to the one line containing the word but expanded to its surrounding lines could improve the performance as well. To improve the performance on the task of detecting similar lines, a manually assembled dataset of similar and dissimilar lines could be used to fine-tune the model. This could be supported by a Visual-Interactive Labeling approach or an Active Learning setting [31] for example in a crowd-sourced environment. Despite their shortcomings in regard to the specific language of rap, the data generated by both models often found artists that share real-life connection as well to be similar, pointing to the solidity of the approach. Building on the prototype and taking advantage of the expandability of the used graph database, the application could be expanded to include a much larger amount of artists and cross-genres. Thus, users would be enabled to discover more artists, especially with the inclusion of lesser known ones. Another opportunity lies in the detection of multilingual alignments with multilingual models like LASER [32].

IX. Conclusion

The proposed prototype offers visual tools for users interested in American rap music to discover similarities between American rap artists and their lyrics. For this, a force-directed graph layout was used to show artists that are similar to each other based on their lyrics. A user can further explore a corpus of rap lyrics through visualizations that aid in collation and the detection of text re-use. Two sentence transformers that produce sentence embeddings for each individual line were utilized to automatically detect semantically related lines. Furthermore, we highlighted limitations and possible future directions to improve the methodology. The application can be accessed here: https://git.informatik.uni-leipzig.de/je17rady/rapvis.

Acknowledgment

This work was partially funded by the German Federal Ministry of Education and Research (BMBF) within the project "Competence Center for Scalable Data Services and Solutions (ScaDS) Dresden/Leipzig" (BMBF grant 01IS14014B) and the Development Bank of Saxony (SAB) within the project "Data Mining and Value Creation" (project number 100335729).

References

[1] D. P. Alridge and J. B. Stewart, "Introduction: Hip hop in history: Past, present, and future," *The Journal of African American History*, vol. 90, no. 3, pp. 190–195, 2005. [Online]. Available: http://www.jstor.org/stable/20063997

[2] A. Light, *The Vibe History of Hip Hop*. Three Rivers Press, 1999. [Online]. Available: https://openlibrary.org/books/OL42726M/The_Vibe_history_of_hip_hop

[3] "Hip-Hop Becomes Most Popular Genre In Music For First Time In U.S. History – VIBE.com." [Online]. Available: https://www.vibe.com/music/music-news/hip-hop-popular-genre-nielsen-music-526795/

[4] K. D. Ricardo, "Legal Writing, the Remix: Plagiarism and Hip Hop Ethics," no. ID 1914857, Aug. 2011. [Online]. Available: https://papers.ssrn.com/abstract=1914857

[5] G. M. G. Inc., "Genius.com," 2014, https://genius.com/ (Accessed 2021-10-27).

[6] S. AB, "Spotify," 2008, https://www.spotify.com/ (Accessed 2021-10-27).

[7] S. Limited, "Soundcloud," 2007, https://soundcloud.com/ (Accessed 2021-10-27).

[8] C. Meinecke, J. Schebera, J. Eschrich, and D. Wiegreffe, "Visualizing similarities between american rap-artists," *Poster EuroVis*, 2022.

[9] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[10] R. Khulusi, J. Kusnick, C. Meinecke, C. Gillmann, J. Focht, and S. Jänicke, "A survey on visualizations for musical data," in *Computer Graphics Forum*. Wiley Online Library, 2020.

[11] Spotify, "Spotify Artist Explorer," 2018, https://artist-explorer.glitch.me/ (Accessed 2021-10-27).

[12] M. Gibney, "Music-Map," 2011, https://www.music-map.de ((Accessed 2021-10-27).

[13] P. Cano and M. Koppenberger, "The emergence of complex network patterns in music artist networks," in *Proceedings of the 5th international symposium on music information retrieval (ISMIR)*. Citeseer, 2004, pp. 466–469.

[14] M. D. Gleich, L. Zhukov, and K. Lang, "The world of music: Sdp layout of high dimensional data," *Info Vis*, vol. 2005, p. 100, 2005.

[15] M. Schedl, P. Knees, and G. Widmer, "Discovering and visualizing prototypical artists by web-based co-occurrence analysis." in *ISMIR*, 2005, pp. 21–28.

[16] S. Lu and J. Akred, "History of Rock in 100 Songs," 2018, https://svds.com/rockandroll/#thebeatles (Accessed 2021-10-27).

[17] J. Ono, D. Corrêa, M. Ferreira, R. Mello, and L. G. Nonato, "Similarity graph: visual exploration of song collections," in *SIBGRAPI*. IEEE, Institute of Electrical and Electronics Engineers United States, 2015.

[18] R. De Prisco, N. Lettieri, D. Malandrino, D. Pirozzi, G. Zaccagnino, and R. Zaccagnino, "Visualization of music plagiarism: Analysis and evaluation," in *2016 20th International Conference Information Visualisation (IV)*. IEEE, 2016, pp. 177–182.

[19] M. Daniels, "The largest vocabulary in hip hop," 2014, https://pudding.cool/projects/vocabulary/ (Accessed 2021-10-27).

[20] M. D. The Data Face, "The language of hip hop," 2017, https://pudding.cool/2017/09/hip-hop-words/ (Accessed 2021-10-27).

[21] C. Meinecke, A. D. Hakimi, and S. Jänicke, "Explorative visual analysis of rap music," *Information*, vol. 13, no. 1, p. 10, 2022.

[22] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," *arXiv preprint arXiv:1802.06893*, 2018.

[23] A. Peckham, "Urban dictionary," 1999, https://www.urbandictionary.com/ ((Accessed 2021-10-27).

[24] T. Yousef and S. Janicke, "A survey of text alignment visualization," *IEEE transactions on visualization and computer graphics*, 2020.

[25] S. Jänicke, A. Geßner, G. Franzini, M. Terras, S. Mahony, and G. Scheuermann, "Traviz: A visualization for variant graphs," *Digital Scholarship in the Humanities*, vol. 30, no. suppl_1, pp. i83–i99, 2015.

[26] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Proceedings 1996 IEEE symposium on visual languages*. IEEE, 1996, pp. 336–343.

[27] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *IEEE Transactions on Big Data*, 2019.

[28] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Information Processing Letters*, vol. 31, no. 1, pp. 7–15, 1989. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0020019089901026

[29] M. Jacomy, T. Venturini, S. Heymann, and M. Bastian, "Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software," *PLOS ONE*, vol. 9, no. 6, pp. 1–12, 06 2014. [Online]. Available: https://doi.org/10.1371/journal.pone.0098679

[30] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation," *arXiv preprint arXiv:1708.00055*, 2017.

[31] J. Bernard, M. Hutter, M. Zeppelzauer, D. Fellner, and M. Sedlmair, "Comparing visual-interactive labeling with active learning: An experimental study," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 298–308, 2017.

[32] M. Artetxe and H. Schwenk, "Margin-based parallel corpus mining with multilingual sentence embeddings," *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3197–3203, 2019.