[CSE6242 Project Wikability] Final Report

Team001 - Brian Popp, Victoria Murray, Kyle D’Souza, Evan Burchard

Introduction: Traditional text recommendation systems evaluate readers’ reading levels and compare them to that of the text to make level-appropriate recommendations. These systems focus solely on textual features such as sentence length and frequency of words to determine readability, without regard for inter-document similarity. Wikability aims to address these weaknesses of existing text recommendation system by providing an interactive graphical tool that combines the calculated textual feature-based readability metrics with content similarity metrics to recommend Wikipedia articles that are reading-level and topic-appropriate. Considering the wide user base of Wikipedia and based on research that shows the benefit of visualization of document clustering, we believe our project offers a new dimension by which language learners and related personnel can discover challenging yet comprehensible articles that are similar to the articles that they had found interesting.

Problem Definition: For language learners, Wikipedia could be a great tool for learning. However, there is no existing Wikipedia article recommendation system that takes user’s reading level into consideration. Wikability provides an intuitive tool to find interesting and reading-level appropriate articles; its interactive article card component visualizes the relative difficulty and similarity between the recommended Wikipedia articles and the article that they initially select.

Survey: We surveyed 12 articles in the proposal process of our project. Omisore et al., (2014) share the techniques used in a digital book recommendation system. The study lacked an in-depth description of how the LBM (Lexile Book Measure) was determined. In our implementation, we may base the basic architecture but need to modify CBI (Content Based Interference) to gather more meaningful measures from the users. Chen & Meurers (2019) propose complexity feature vectors based on learners’ recent writing output and perform various analyses to prove the efficacy of their methods. Despite the complexity of the suggested techniques, we can still implement various statistical analyses and visualizations used in their research to design and measure the success of our implementation with simplification. Sherkat et al., (2019) propose a key-term-based visual interactive document clustering method that involves user input. The visualization aspect of their research with the highest user interaction and rating could be our focus for implementation. We will need additional metrics to determine and cluster articles based on desired lexical levels. White & Ramesh (1996) provide a good overview of traditional techniques used to index high dimensional feature vectors for filtering based on similarity. Given our schedule limitations, we will find a similar commercial or open-source implementation. Crossley et al., (2019) and Feng et al., (2010) both demonstrate the weaknesses of traditional readability models and provide new features and techniques for measuring readability using Natural Language Processing and Machine Learning. We hope to leverage this research and improve it by using visualization to evaluate and improve performance. Shmueli et al., (2018) describe document processing and analysis such as computing TF-IDF, and stemming, which is useful to us even though the specifics of our approach may differ. Whissell & Clarke (2013) suggests that among non-learning-based models, cosine similarity (with TF-IDF to measure document similarity remains popular but proposes BM25 as an alternative. Agrawal et al., (2009) demonstrate techniques for displaying diverse results, which can be useful to us even though our primary concerns are similarity and readability. Nakamura et al., (2018) proposed semantic relatedness methods for multilingual short text clustering and measuring document relatedness using Wikipedia articles. The datasets utilized Twitter hashtags for four different languages and solved the semantic gap problem by incorporating inter-language links of Wikipedia into Extended Naive Bayes. This will aid in our algorithm implementation for recommendations, and we plan to improve by using a more diverse and robust dataset. Wenskovitch et al., (2017) discuss problems with dimensionality reduction and clustering visualization along with various solutions. This will aid in the key portion of our visualization and extend their conclusions for text document clustering. Kim E.HJ.&Kim S., (2016) proposes how to identify word associations that indicate a specific sentiment polarity and semantics. Their method extracted word co-occurrences and converted them into a cosine adjacency matrix. A co-word network was constructed by applying Pathfinder scaling. The context score was measured and presented context paths from the context structure in the review texts. This will aid in our recommendation algorithm, and we plan to expand the limit scope from review polarity to different hierarchical categories.

Approach: For the Wikability project, our big data comes from the Wikipedia database dump "enwiki-20220101-pages-articles-multistream." We were able to download 19.21GB compressed (85GB uncompressed) file from https://en.wikipedia.org/wiki/Wikipedia:Database_download, which contains information of roughly 6.5 million English article entries in XML format. Each article is written in a complicated, proprietary markup with thousands of citations, categories, and cross-references. A subset of 200,000 articles is being used for our project. Given the short implementation period of the project, we focused on the English language, but a similar approach could be used for other languages in future implementations.

The Wikipedia XML files were then parsed in our python-based parser for necessary information such as title, article texts, and categories. Instead of reinventing the wheel, we relied on several open-source libraries to handle some of the more tedious, complicated, and time-consuming aspects of the project, such as parsing, TF-IDF vectorization, similarity score calculation, and readability calculations. To parse the data, we used the 'mwxml' and 'wikitextparser' libraries. With these libraries, we were able to parse the MediaWiki's XML files quickly and reliably for the initial parse. Afterward, we used regular expressions and the 'spacy' library for the additional cleanup and Natural Language Processing (NLP) such as removal of stop words, symbols, whitespace, and tokenization. The 'pandas' library was used for data cleanup, storage, and retrieval.

Once the data was cleaned, TF-IDF vectorization and cosine similarity scores were calculated with 'sklearn' library. This calculation shows how similar each article is to another. Initially, the TF-IDF algorithm worked well with zero feature engineering; its term weighting essentially ignores common stop words. However, some feature engineering was needed once we began working with larger datasets, since word counts for a 100k article corpus were over 119,000,000 words.

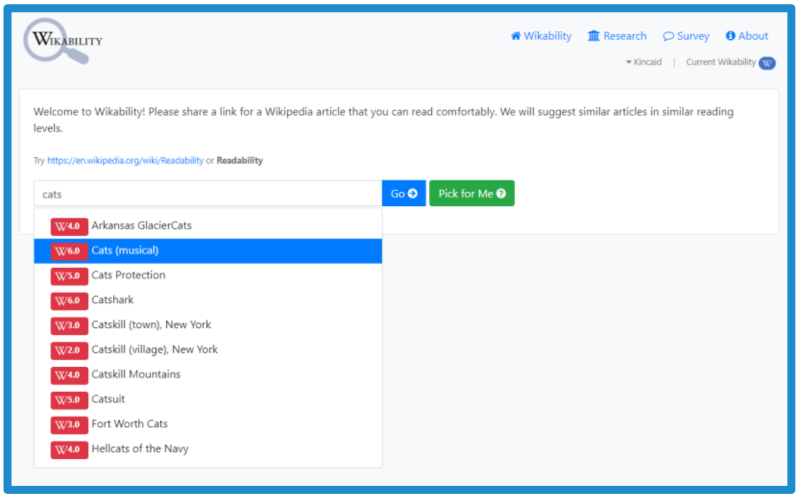
Processing and storing such a large volume of data in this step posed several challenges for us to overcome. The TF-IDF algorithm builds a sparse matrix of documents (m) x words (n). For a 200k corpus, the storage requirements were initially around (200,000, 2,525,731, 8b) or 3.7TB. After analyzing our corpus, we learned that the mean word count for the articles is 1200, with the standard deviation of 1600 words. The longest article had around 25,000 words. Since around 80% of articles had less than 1500 words, we opted to limit the number of words for each article to 1500. This improved performance and significantly reduced storage requirements.

Similarly, calculating the cosine similarity scores posed challenges. The cosine similarity algorithm normally returns a matrix of documents (m) x documents (n). For a 200k corpus, the storage requirements would be (200,000, 200,000, 8b) or a daunting 320GB of memory. We instead compared each document to every other document, then eventually reduced the range to the top 50 highest cosine similarity scores. This is a more cluster-friendly algorithm and reduced overall storage requirements to (200,000, 50, 8b). We struggled with finding a balance between our desired features and performance. However, we were able to employ several tools to find the right balance including feature optimization, caching, and multi-threading.

Multiple readability scores, such as 'Flesch-Kincaid,' 'Gunning Fog,' 'SMOG index,' along with different norms of the TF-IDF word vectors (L0, L1, L2, Infinity norms) were calculated for each article using the 'numpy' library. Because cleaning the text such as removing stop words and stemming interfered with calculating the readability score, the original text was used for readability score calculations. To calculate and study the readability of the articles, we measured additional features such as number of words, sentences, syllables, paragraphs and stored them in our database. We studied the relationship between the readability scores vs. cosine similarity scores, and norms of TF-IDF vs.

readability scores to explore the possibility that these norms can act as proxies for extant readability metrics. Our findings are discussed in the research section of our website.

After some research, we found the 'readability' python library allows us to calculate eight readability metrics (Flesch-Kincaid, ARI, Coleman-Liau, Gunning Fog Index, LIX, RIX, SMOG index, Dale-Chall Index) more easily and used the library for the implementation. We have also developed a readability scoring system called 'Wikability Score,' which ranges from 1 to 10 for each readability metric. Based on min and max score for each readability scale, the readability scores were divided into 10 decile groups. This Wikability score is used to show which articles are 'harder' to read vs. 'easier' to read than the original seed article and to search for other articles with similar reading difficulty. This information, along with 14.4 million features were temporarily exported as CSV files to test accuracy, then were imported into our SQLite database tables. We have cross-referenced (joined) multiple tables to query the necessary information needed to make suggestions for the Wikability.



Welcome to Wikability! Please share a link for a Wikipedia article that you can read comfortably. We will suggest similar articles in similar reading levels.

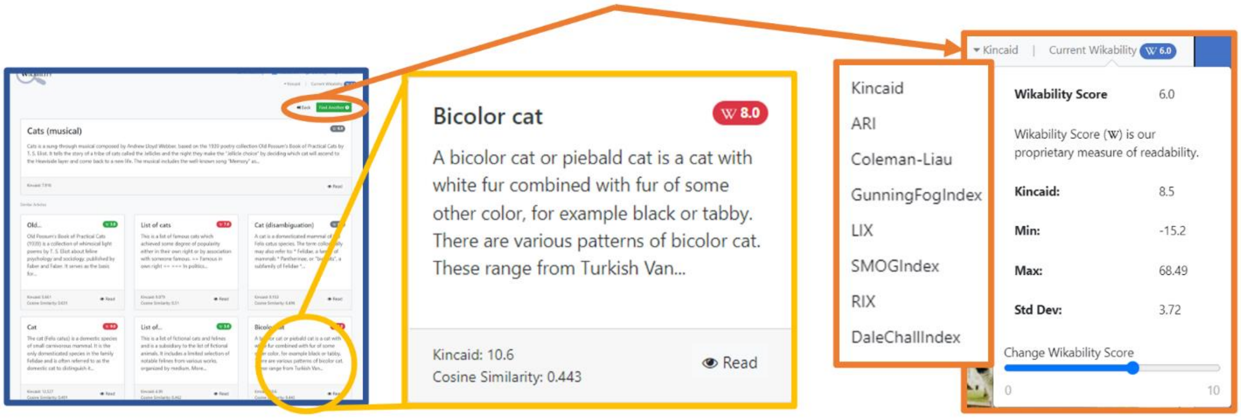
Try https://en.wikipedia.org/wiki/Readability or **Readability**

cats

W4.0 Arkansas GlacierCats
W6.0 Cats (musical)
W5.0 Cats Protection
W6.0 Catshark
W3.0 Catskill (town), New York
W2.0 Catskill (village), New York
W4.0 Catskill Mountains
W5.0 Catsuit
W3.0 Fort Worth Cats
W4.0 Hellcats of the Navy

Figure 1. landing page

Our website is hosted on www.Wikability.com. The landing page as shown in figure 1, prompts the user to enter a URL for a Wikipedia article that they can comfortably read on a topic that interests them. Since we are using a subset of English Wikipedia articles, our keyword search bar gently nudges users into looking up articles extant in our database by showing auto-populated suggestions from our database. 'Pick for me' features also allow the user to select an initial article (seed article) to start.



Bicolor cat W 8.0

A bicolor cat or piebald cat is a cat with white fur combined with fur of some other color, for example black or tabby. There are various patterns of bicolor cat. These range from Turkish Van...

Kincaid: 10.6
Cosine Similarity: 0.443

Wikability Score 6.0

Wikability Score (W) is our proprietary measure of readability.

Kincaid:	8.5
Min:	-15.2
Max:	68.49
Std Dev:	3.72

Change Wikability Score

Figure 2. Suggested Article Card and Wikability Score slider

In the next webpage as shown in figure 2, the seed article is displayed on the top, followed by the six suggested articles in the form of cards. Wikability score is set to the seed article's Wikability score.

This score is used to generate future suggestions when user clicks on ‘Find Another’ button and for color coding the suggested articles. Six articles with the highest cosine similarity score in relation to the seed article are suggested, which means that these articles are the most closely related articles to the seed. Each suggested article card displays the title, short excerpt from the article, selected readability metric score, cosine similarity score in relation to the seed article, and the Wikability score with distinct colors. Wikability score in green indicate the articles that are easier to read than the seed article (or the Wikability score that they choose using the slider), grey indicate the articles that are in similar reading difficulty, and red indicate the articles that are more difficult to read. Users can change the desired readability metric that they want to display the scores in or change the Wikability score using the slider. Initially, we had proposed the network graph for the visualization. However, after several discussions with potential users regarding their preference, we decided that the suggestion cards were cleaner, more readable, and user-friendly visualization than the graph format.

When the user clicks on the card, it will lead the user to ‘read’ page with embedded article. In future phases of implementation, we can add tracking capabilities in each page to study how much time user spends on reading articles of certain readability level. ‘Flask’ was used for the backend development with ‘jinja’ for templating for front end.

Our team performed extensive research and data exploration in order to develop Wikability. Some of these research works such as Readability vs. Cosine similarity relationship, principal component analysis, and TF-IDF regression study performed in Python and R are shown in our ‘research’ section.

List of Innovations / Intuition: The main innovation of Wikability comes from combining TF-IDF similarity and readability scores to suggest interesting and reading-level appropriate Wikipedia articles. We have also explored relationships between TF-IDF norms and existing readability metrics, as well as relationship between cosine similarity and readability. There are only a handful of recommendation systems available for Wikipedia; and we could not find any existing system that uses readability scores or provides an interactive visualization tool.

Regarding the innovations in visualization, Wikability provides a simple yet readable visualization that shows how the suggested articles are related to the original article in terms of similarity and readability. Furthermore, it provides a series of interesting graphical visualizations to allow users to learn about the readability of Wikipedia articles in the ‘Research’ section. For example, we show the difference of words in ‘easiest’ to read article vs. the ‘hardest’ to read article in word bubbles, cosine similarity vs. Readability metric values, PCA plot users can examine from different angles, and colored correlation heat map for readability metrics vs. TF-IDF norms.

Design of Experiment / Evaluation: Wikability was evaluated in three parts to measure scalability, accuracy of readability and similarity measures, and usability. The first evaluation was done by the Wikability QA team. Scalability of our algorithm was checked during implementation; and human-judged qualitative criteria (i.e., Article A and B are related based on these keywords, Article A seems to be easier to read than B, based on its shorter length and easier vocabulary), combined with quantitative criteria such as correlations between similarity score and readability scores was used to check for accuracy of readability and similarity of suggested articles.

The second evaluation was performed using the information gathered from the user feedback. The users/testers were recruited by sharing our project link on internet forums like Piazza. A brief five-question survey designed to assess how users felt regarding the similarity and the readability of the suggested articles, as well as the functionality of visual cards were collected. We understand that with such a limited amount of user feedback gathered in a short testing period, the findings may not accurately represent the success or failure of our project compared to a larger dataset. To gather more reliable results, we can add code to track behind-the-scenes measures, such as the time users spent reading the article embedded on our website, the proportion of number of times the easier or harder articles were clicked to compare with the survey results in the future phases of implementations.

The last evaluation was done by checking and comparing the distribution of Wikipedia readability levels calculated by us to the results of a few existing studies on the topic and comparing our calculated readability scores with other factors such as similarity scores and TF-IDF norms.

Conclusion and Discussions: For the first evaluation method, we encountered scalability issues with the cosine similarity calculations. We had to find creative solutions such as reducing the number of articles to process, number of words per article, amount of cosine similarity calculations per article, and using multi-threading etc., to work around the scalability issues. Using our proprietary human-judged qualitative criteria, the cosine similarity comparison worked very well to suggest articles that are related. Although it was difficult to accurately assess the difference in readability per se with no background in linguistic studies, through our research and data exploration, we determined that some features such as a high number of total words per article or a high number of technical or obscure terms (i.e., names of regions) corresponded with very high Wikability scores (8-10: hard to read) while articles with fewer total words or fewer technical terms corresponded with low Wikability scores (1-3: easy to read). This difference is easily observable in the word bubble shown in figure 3, although more obscure or technical words such as 'Yakutsk,' 'crepuscular' are displayed in smallest bubbles that are hard to observe. The html version of these bubbles that show each word and frequency in bubbles can be found in our research section.

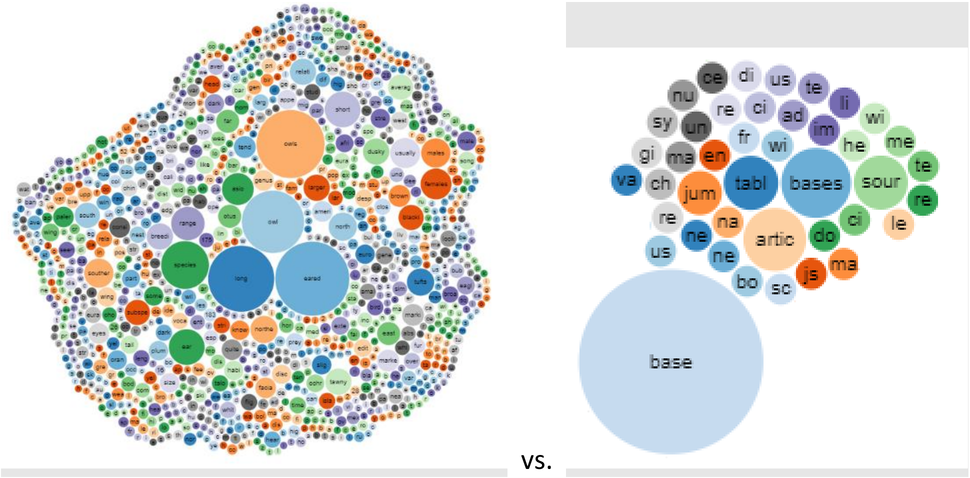


Figure 3. Word Bubbles for Hardest to Read (left) vs. Easiest to Read(right) Articles in Flesch-Kincaid metric

At the time of drafting this report, we received 15 responses in the five-question user survey. The four multiple-choice questions and percentage distributions for the answers are displayed in table 1. From the user feedback, we can conclude that users were highly satisfied with our visualization in interactive card format, with 100% of users answering 'yes' to question 1. 93.7% of users showed high satisfaction in similarity of the articles suggested, based on the cosine similarity score in relation to the seed article.

Mutiple Choice Questions	Answer %
1. Did the interactive cards help you see the relationship between the suggested articles and the original?	Yes: 100%
	No: 0%
2. Were the readabilities of the articles suggested in grey cards similar to the initial article you've provided?	Yes, provided similar reading challenges: 86.7%
	No, easier to read:13.3 %
	No, harder to read: 0%
3. Did you find that suggested articles are similar to the original article?	Yes: 93.7%
	No: 6.3%
4. Do you think Wikability was able to recommend interesting (similar to the article you provided) articles that are in the reading level you desired?	Yes: 86.7%
	No: 13.3 %

Table 1. Product Evaluation Survey User Feedback

Although a similarly high percentage (86.7%) of users have responded that suggested articles with the same Wikability scores (shown in grey) provided similar reading challenges compared to the seed article, the remainder of users have mentioned that it was easier to read. 86.7% of the users so far have mentioned that Wikability was able to recommend interesting (or similar) articles that are in the reading level that they had desired, while 13.3% has answered that it did not. Based on this positive user feedback, Wikability was successful in accurately evaluating similarity and readability of the Wikipedia articles. A user has provided feedback on additional features such as saving the titles and links of the Wikipedia articles that they had read and displaying it when user clicks the ‘done’ button. This feature can be added in future implementations to deliver higher user satisfaction.

For the third evaluation criteria, we were unable to recreate the distribution of Flesch-Reading Ease scores of Wikipedia articles from an existing study which approximately followed a gaussian distribution. This is mostly due to the selection error, where we did not randomly select the 200K articles to process for Wikability (we selected the first 200K entries). Instead, we have performed comparison between article readability and cosine similarity. Interestingly, it shows that articles with similar readability tend to have higher cosine similarity. Since readability considers sentence/document structure, it could imply that the word vector frequencies can give insight into sentence structure. Also, the articles with higher readability tended to have higher cosine similarity. This means that the easier a document is to read, the more in common it has with other documents. In our PCA evaluation, we had found that when plot is colored based on Flesch Reading Ease score, some of the most extreme outliers were not the darkest bubbles, indicating a difference between Flesch Reading Ease score and the other readability scores, which is true as its scoring works in inverse of other metrics. Highest scores in Flesch Reading Ease means the article is easy to read, not harder.

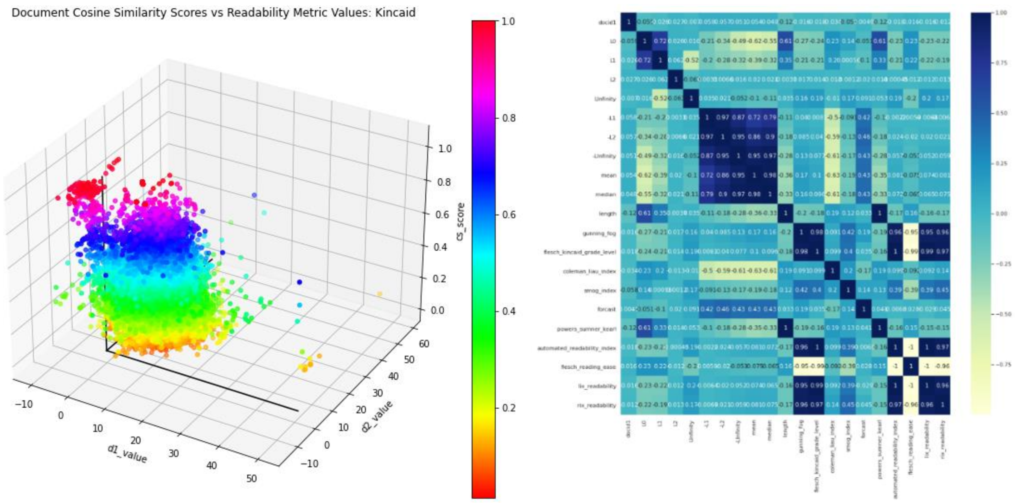


Figure 4. Cosine Similarity Score vs. Flesch-Kincaid Readability Metric & Readability vs. TF-IDF Norm Correlation

In our exploration to find the correlation between the norms (L0, L1, L-Infinity, -L1, -L2, -L-Infinity) of an article’s TF-IDF word vectors and readability metrics, we used regression for the norms (and their interaction parameters) on the Flesch-Kincaid scores (which was chosen for its high correlation with other metrics) of a sample of 1K articles. A log model showed some promise with an R-squared value of .4, but the regression assumptions did not hold. As we expected, there were high correlations with some of the more exotic readability measures (i.e., Powers Sumner Kearl) with certain norms due to sharing metric components. To see the visualization and our research in detail, please refer to the Research section of our website: www.wikability.com/research/view.

Efforts: All team members have contributed a similar amount of effort.

References

Agrawal, Gollapudi, S., Halverson, A., & Ieong, S. (2009). Diversifying search results. *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, 5–14. https://doi.org/10.1145/1498759.1498766

Chen, & Meurers, D. (2019). Linking text readability and learner proficiency using linguistic complexity feature vector distance. *Computer Assisted Language Learning*, *32*(4), 418–447. https://doi.org/10.1080/09588221.2018.1527358

Crossley, Skalicky, S., & Dascalu, M. (2019). Moving beyond classic readability formulas: new methods and new models. *Journal of Research in Reading*, *42*(3-4), 541–561. https://doi.org/10.1111/1467-9817.12283

Feng, L., Jansche, M., Huenerfauth, M., & Elhadad, N. (2010). A comparison of features for automatic readability assessment.

Kim, E.HJ & Kim, S. (2016). An Effective Approach to Finding a Context Path in Review Texts Using Pathfinder Scaling. *Social Informatics*, 376–388. https://doi.org/10.1007/978-3-319-47880-7_23

Nakamura, Shirakawa, M., Hara, T., & Nishio, S. (2019). Wikipedia-Based Relatedness Measurements for Multilingual Short Text Clustering. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *18*(2), 1–25. https://doi.org/10.1145/3276473

Omisore, O.M. & Samuel, O. (2014). Personalized Recommender System for Digital Libraries. *International Journal of Web-Based Learning and Teaching Technologies*, *9*(1), 18–32. https://doi.org/10.4018/ijwltt.2014010102

Sherkat, Milios, E., & Minghim, R. (2019). A Visual Analytics Approach for Interactive Document Clustering. *ACM Transactions on Interactive Intelligent Systems*, *10*(1), 1–33. https://doi.org/10.1145/3241380

Shmueli, Bruce et al., (2018). Data Mining for Business Analytics: Concepts, Techniques, and Applications in R, 477-492. John Wiley & Sons. Inc.

Wenskovitch, Crandell, I., Ramakrishnan, N., House, L., Leman, S., & North, C. (2018). Towards a Systematic Combination of Dimension Reduction and Clustering in Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics*, *24*(1), 131–141. https://doi.org/10.1109/TVCG.2017.2745258

Whissell & Clarke. (2013). Effective measures for inter-document similarity. *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, 1361–1370. https://doi.org/10.1145/2505515.2505526

White & Ramesh. (1996). Similarity indexing with the SS-tree. *Proceedings of the Twelfth International Conference on Data Engineering*, 516–523. https://doi.org/10.1109/ICDE.1996.492202