WIKABILITY: A WIKIPEDIA RECOMMENDATION SYSTEM FOR LANGUAGE LEARNERS

www.WIKABILITY.com

Team001: Brian Popp, Victoria Murray, Evan Burchard, Kyle D'Souza

MOTIVATION / INTRODUCTION

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. Also, 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 the textual features such as sentence length and frequency of words to determine readability, without regard for inter-document similarity.

Wikability aims to address weaknesses of existing text recommendation systems by providing an interactive graphical tool that combines the calculated textual feature-based readability metrics with content similarity metrics, recommending Wikipedia articles that are reading-level and topic-appropriate. Considering the wide user base of Wikipedia and research that shows the benefit of visualization of document clustering, Wikability will offer 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.

DATA

400.8M Words | **200K** Articles | **337K** Categories | **14.2M** Features

For the Wikability project, the big data comes from the Wikipedia database dump "enwiki-20220101-pages-articles-multistream." 19.21GB compressed (85GB uncompressed) file from https://en.wikipedia.org/wiki/Wikipedia:Database_download was downloaded, 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 was used for the English language implementation, but a similar approach can be used for other languages in future work.

APPROACHES

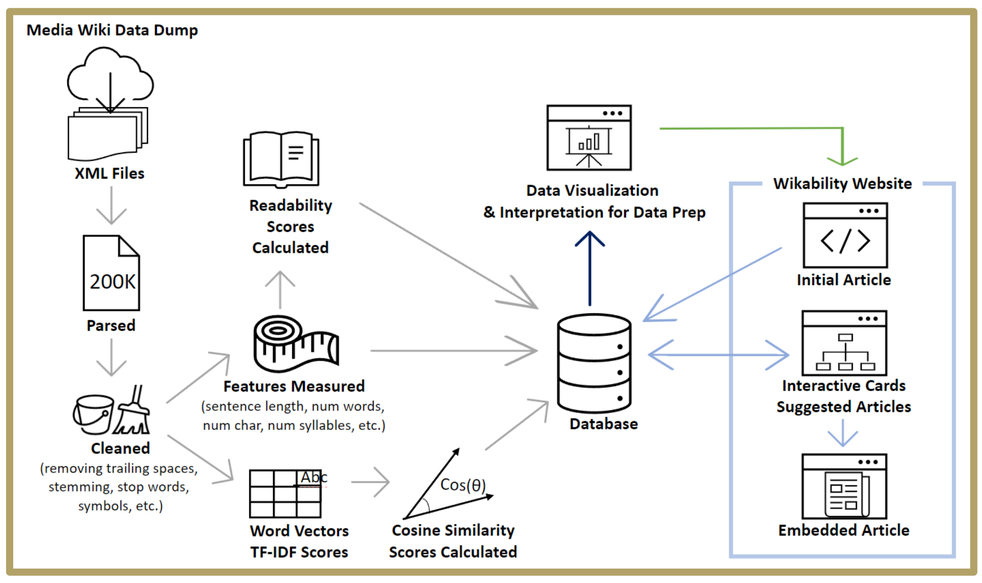

Figure 1. Wikability Architecture

After the Wikipedia database dump files were downloaded, the Wikipedia XML files were parsed in our python-based parser as necessary. 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.

Features such as sentence length, number of words, syllables, and paragraphs were measured to calculate readability scores and for data exploration. Readability scores with 8 different readability metrics (Flesch-Kincaid, ARI, Coleman-Liau, Gunning Fog Index, LIX, SMOG Index, RIX, Dale Chall Index), along with different norms of the TF-IDF word vectors (L0, L1, L2, Infinity norms) were calculated for each article using the 'numpy' and 'readability' libraries. All of these data were stored in our SQLite database.

Data regarding cosine similarity and readability scores were used for data visualization and extensive data exploration and research in order to develop Wikability. Some of these research such as relationship between Readability scores vs. Cosine similarity, PCA, TF-IDF norm vs. readability regression study performed in Python and R are shown in our 'research' section of the website. Another interesting visual to look for is the word bubbles comparison between the articles with lowest vs. highest readability.

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

The next page displays the seed article on the top, followed by 6 suggested articles as cards. These articles have the highest cosine similarity scores, which means that they are most closely related to the seed article. The Wikability score (between 1 - 10) slider is set to the seed article's Wikability score. This score, which divides each readability metric scores into decile, is used to generate future suggestions when user clicks on 'Find Another' button and for color coding the suggested articles. Each suggested article card displays the title, a short excerpt from the article, the selected readability metric score, the cosine similarity score in relation to the seed article, and the Wikability score with 3 distinct colors:

- **Green:** Easier to Read
- **Grey:** Similar in Reading Difficulty
- **Red:** More difficult to Read

Users can change the desired readability metric to display the scores or change the Wikability score using the slider. When the user clicks on the card, it will lead the user to 'read' page with embedded article.

'Flask' was used for the backend development with 'jinja' for templating for the front end.

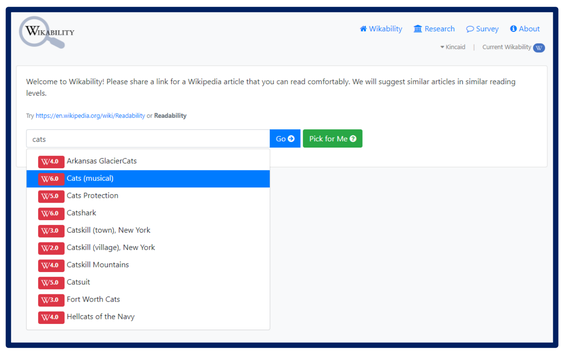

Figure 2. Landing Page with 'auto complete' and 'Pick for Me' features

PRODUCT DEMO

Watch 1min Demo Video

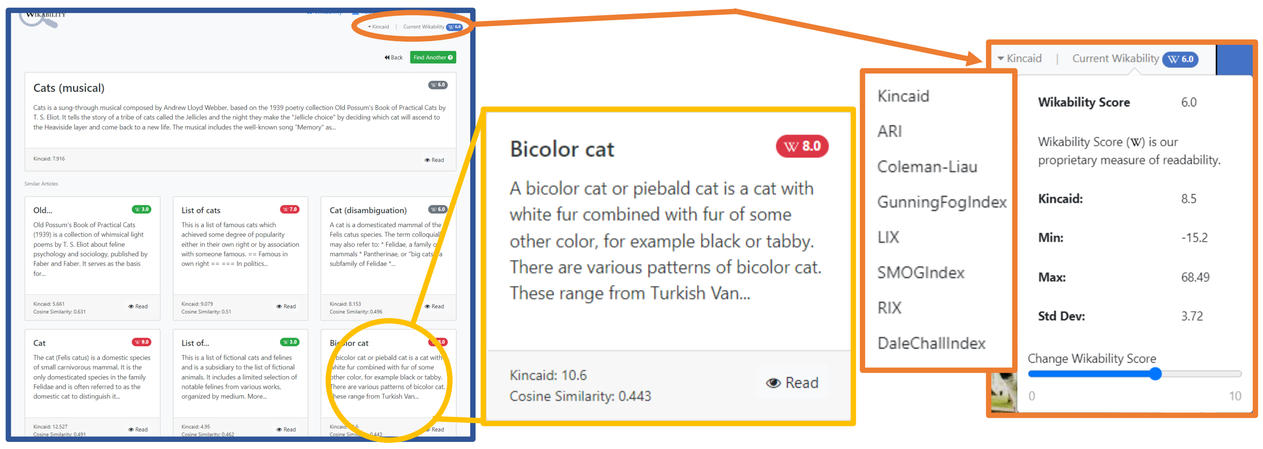

Figure 3. Suggestions Page with 'Article Card' (in yellow), 'Readability Metrics' and 'Wikability Score' features (in orange)

EVALUATIONS

3 Evaluation Tactics:

- Human-Judged Qualitative Criteria (for scalability, accuracy of readability and similarity)
- User Feedback (for usability, accuracy of readability and similarity)
- Comparison with Existing Studies and Our Own Research

FINDINGS / RESULTS

- **Challenges with scalability:** 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. 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.

- **Accuracy of Similarity and Readability by Human Judged Qualitative Criteria**: Cosine similarity worked very well with document comparison. This is later confirmed by user feedback. Accuracy of Readability also seemed to correspond with difference in number of words and use of obscure or technical terms. Word bubbles of highest scored in Flesch-Kincaid metric article (most difficult to read) vs. word bubbles of lowest scored article (easiest to read) article show this difference very well.

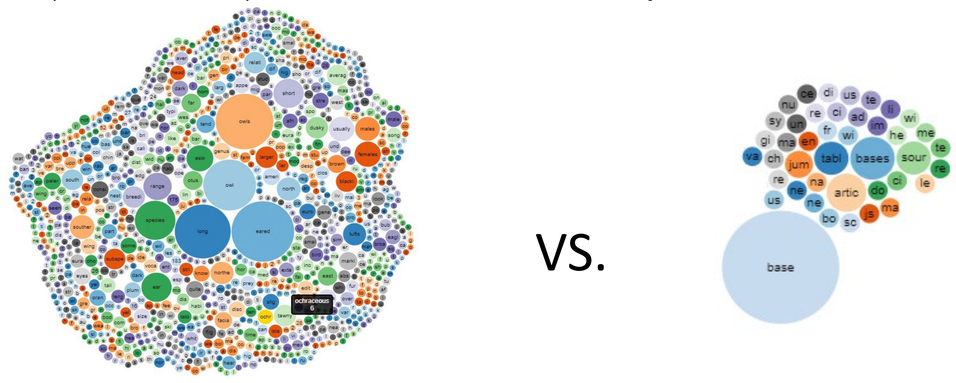

Figure 4. Hardest Article: Long Eared Owl (Kincaid Score: 68.49) vs. Easiest Article: Table of Bases (Kincaid Score: -14.99)

- **Measuring Usability Through User Feedback:** (Based on 15 responses)

Multiple Choice Questions	Answer %	
1. Did the interactive cards help you see the relationship between the suggested articles and the original?	Yes: 100% No: 0%	☑ Helpful Visualization
2. Were the readabilities of the articles suggested in grey cards similar to the initial article you've provided?	Yes: 86.7% No, easier to read:13.3% No, harder to read: 0%	☑ Accuracy of Readability
3. Did you find that suggested articles are similar to the original article?	Yes: 93.3% No: 6.7%	☑ Accuracy of Similarity
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%	☑ Usability

- **Research Findings**: 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 it has in common with other documents.

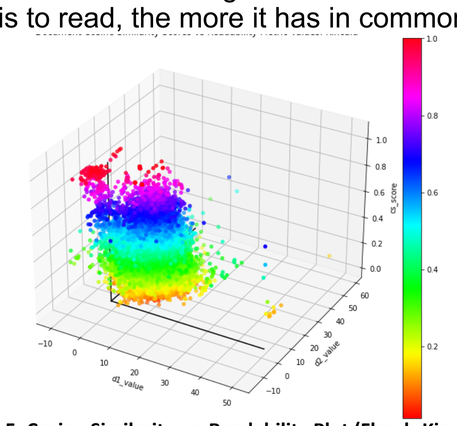

Figure 5. Cosine Similarity vs. Readability Plot (Flesch-Kincaid)

RESEARCH FINDINGS

- Cosine Similarity vs. Readability
- TF-IDF Norm vs. Readability
- PCA Analysis