“How can a system be developed to automate the creation and update of the 'See Also' section in Wikipedia articles by utilizing article features and NLP-generated semantic vectors?”

This project is submitted to complete the requirements for the Data Science Group Project Module, led by [redacted] at the University of Birmingham.

[redacted]

[redacted]

Abstract

This study presents a new system that aims to automate the process of creating "See Also" sections in Wikipedia articles. The system improves the relevance of links and makes it easier to navigate through the most popular 60,000 English Wikipedia articles by using article features, Natural Language Processing (NLP), and semantic vector analysis. Our approach relies on BERT (Bidirectional Encoder Representations from Transformers) to generate semantic vectors. These vectors are then combined with article-specific features to provide recommendations that are tailored to the context. Surveys among Wikipedia editors were conducted to evaluate the effectiveness of this automated system. Based on the results, it was found that a majority of the editors, specifically 58.36%, preferred the automated "See Also" lists generated by the system over current manual ones. The results highlight the system's ability to improve the connectivity between articles on Wikipedia. Although the performance shows promise, some areas can be further improved to fully maximize its potential in terms of coverage and precision. This study offers valuable insights into how AI, ML, and NLP technologies can improve digital informational ecosystems. This report also sets the foundation for future studies focused on enhancing automated content linkage.

Contents

1. Introduction

This report provides an overview of our project focused on automating the "See Also" section in Wikipedia articles. Our project employs advanced Natural Language Processing (NLP) techniques and article feature analysis to revolutionize the way readers navigate and explore interconnected topics on Wikipedia. The main goal of this project was to create a system that could improve the reader's experience by connecting related articles based on their content and context, and save the volunteers' time.

This project was initiated to address the need for enhancing the efficiency and relevance of the "See Also" sections on Wikipedia, a source that is being accessed more than 25 billion times a month.[1] Nowadays, these sections are manually created by editors, but this method can sometimes be biased and inconsistent. Our goal was to automate this process and guarantee that article recommendations are consistent in quality, relying on measurable semantic relationships and reader engagement metrics.

We conducted a study to address the following question: "How can a system be developed to automate the creation and update of the 'See Also' section in Wikipedia articles by utilizing article features and NLP-generated semantic vectors?"

During this project, we conducted extensive data collection and analysis. We utilized a dataset of the top clicked 60,000 English Wikipedia articles. The development process revealed various areas that could benefit from innovation, such as choosing the most efficient NLP techniques and finding the best methods to integrate article data points into a cohesive recommendation system. We greatly appreciate the contributions of Wikipedia administrators and the global editor community in shaping the project's direction and ensuring its alignment with user needs and Wikipedia's editorial standards.

This report provides a comprehensive overview of our project, covering all the key stages, including data collection, model selection, implementation, and community feedback. This report presents our findings and methodologies, serving as both a record of our work and a valuable resource for future research and development in the field of automated content linking and digital knowledge management.

2. Background Research and Related Work

There are numerous studies focused on automating the 'See Also' section or other processes in Wikipedia by leveraging recent technologies. These studies aim to enhance user experience and increase content accessibility. They explore a range of methodologies and frameworks, utilizing diverse aspects of machine learning to accomplish this goal.

2.1. Semantic Vector Analysis and Ensemble Approaches

A notable study by Sahiti Labhishetty et al. (2017) titled "WikiSeeAlso: Suggesting Tangentially Related Concepts (See also links) for Wikipedia Articles" explores an ensemble-based approach that integrates category knowledge, backlink information, and ESA concept vector similarity with external web search knowledge to automate 'See Also' recommendations. The proposed method emphasizes the importance of combining various data aspects to refine the suggestions, thus enhancing the browsing experience for users. [2]

2.2. Natural Language Generation for Article Bootstrapping

Another study by Lucie-Aimée Kaffee and colleagues (2022) examines the use of Natural Language Generation (NLG) to automate the creation of missing Wikipedia articles, a concept that can be extended to the automation of 'See Also' sections. This approach utilizes a human-centric perspective to evaluate the generated content's fluency and appropriateness, which could be pivotal in generating contextually relevant 'See Also' links. [3]

2.3. Unsupervised Infobox Template Identification

An unsupervised method to automate the identification of infobox templates in Wikipedia articles is explored by Hanif Bhuiyan et al. (2015). This method uses NLP to process article text and identify the most suitable template, which supports the automation of structural elements in Wikipedia articles, indirectly contributing to the automation of 'See Also' sections. [4]

2.4. Contextual Features in NLP Models

Research on Contextual Long Short-Term Memory (CLSTM) models for large-scale Natural Language Processing (NLP) tasks by Shalini Ghosh et al. (2016) discusses the integration of contextual features into Long Short-Term Memory (LSTM) networks. This approach significantly improves the semantic understanding of text, which is crucial for accurately linking related articles in Wikipedia. [5]

These studies highlight the potential of NLP and ML in improving user interactions with digital encyclopedias such as Wikipedia. The authors emphasize the use of creative methods that integrate article features and advanced machine learning models to streamline and improve the 'See Also' sections, which are crucial for navigating related topics. The incorporation of these technologies enhances user engagement and guarantees the relevance and variety of content discovery on the platform.

3. Question Development

The process of refining our research question was characterized by continuous improvements, lively discussions, and a strong dedication to addressing an important requirement in the digital knowledge ecosystem. At first, our vision was quite ambitious. We aimed to use technology to improve the usefulness and accessibility of Wikipedia, particularly by automating the "See Also" section in articles. The main focus of our challenge was to find a framework that could meet this need and also be precise and adaptable to keep up with the ever-changing nature of human knowledge.

Our initial exploration into formulating the research question led us to consider the potential of using Natural Language Processing (NLP) in clustering articles by topics. Drawn to the idea, we imagined a system that would use NLP to cluster articles together according to the semantic vectors of their content, making it easier for editors by providing automatic suggestions for related content, our question was: "How can we utilize Natural Language Processing (NLP) to automatically cluster Wikipedia articles by topic, providing editors with suggestions to be put in the "See Also" section?"

This concept, although it took a unique approach to organizing content, quickly showed notable drawbacks when examined more closely. We came to a clear realization: despite having articles clustered by semantic vectors, the task of editors to go through these recommendations was still challenging. In addition, the current infrastructure of Wikipedia already organizes articles into topics (categories), which raises questions about the additional value our initial approach provided.

At this crucial point, a series of brainstorming sessions took place that had an important impact. As we explored Wikipedia's content ecosystem, we realized the significance of considering not only semantic analysis but also incorporating article features like size, quality ratings, and viewer engagement metrics. We made a breakthrough when we started viewing these features as interconnected elements instead of separate data points. By combining them with NLP-generated semantic vectors, we were able to gain a comprehensive understanding of content relevance.

As our research question got clearer, our project grew from a simple idea of using Natural Language Processing (NLP) to cluster articles by topic to a more complex goal of using NLP along with specific article features to automate the creation of Wikipedia's "See Also" sections. Our knowledge of both the technical challenges and the ways that technology could have a big effect on content creation grew as we changed our focus. Through this process, we came up with a more specific research question: "How can a system be developed to automate the creation and update of the 'See Also' section in Wikipedia articles by utilizing article features and NLP-generated semantic vectors?" This question shows our progress from a general idea to a focused study of how to effectively link articles. It also shows our dedication to making a tool that could improve the user experience by suggesting relevant, popular, and high-quality articles that fit the situation.

4. Collecting and Preparing the Data

Preparing and collecting data for our project was quite challenging from the beginning. The dataset we needed wasn't readily available, presenting us with the significant hurdle of obtaining the necessary data for our system. This required us to invest considerable effort not only in gathering the required data but also in developing the skills necessary for its collection. We divided the data collection process into two stages to manage our efforts efficiently and effectively.

4.1. Stage One: Selecting Articles' Titles

The code relevant to this section is available in Section 1.1 of our Google Colab file

We started by identifying the Wikipedia articles that would be included in our system. Our focus was on the English Wikipedia, which has a significant number of readers and receives over 12 billion visits every month. Nevertheless, creating our system for the extensive number of articles, which amounted to more than 6.7 million articles, proved to be an overwhelming task that needed too much computational resources and time. To handle this, we relied on the clickstream data from January 2024, as our project started in February. Our aim was to identify the top 60,000 clicked articles. This selection was carefully chosen to ensure that our study remains highly relevant, while also taking into consideration the limitations of our computational resources. These articles, which attract over 4 billion and 30 million visits per month, make up a substantial portion of Wikipedia's traffic, accounting for approximately 40%. [1]

4.2. Stage Two: Delving into Article Features

The code relevant to this section is available in Section 1.2 of our Google Colab file

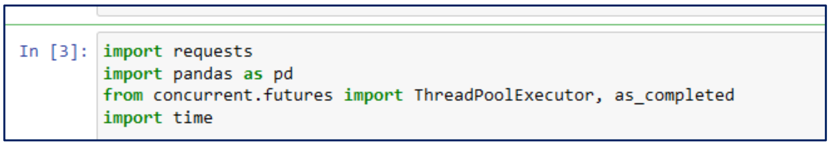
Following the identification of the 60,000 most-clicked articles, our project moved on to a more detailed phase: extracting specific features for each article. This process was vital for the "See Also" automation system we had in mind, which heavily relied on a sophisticated comprehension of article content through Natural Language Processing (NLP). We had a comprehensive plan in place: generating the semantic vectors using the title, introduction, and categories of each article, and implementing filters based on article size, monthly views, and quality. Please refer to Figure 3 in Appendix A (List of Figures) for an overview of the dataset preparation process.

The Learning Curve

Getting started was quite challenging as we had to overcome a significant learning curve. Our team delved into the realm of APIs, web scraping, and data processing tools. Our approach involved extracting the article's introduction and categories via web scraping. These elements, along with the titles, were then utilized in our NLP model to generate semantic vectors. The article size and quality metrics were obtained from Xtools API, while the view counts for January 2024 were taken directly from Wikipedia's views tool API.

Implementing the Data Collection

We started the data collection process using our newly acquired knowledge and skills. The task was quite challenging, requiring intricate requests to Wikipedia's API, we were able to speed up the process by parallelizing our requests using the *concurrent.futures* library, all while being considerate of the load on Wikipedia's servers.



```
In [3]: import requests
        import pandas as pd
        from concurrent.futures import ThreadPoolExecutor, as_completed
        import time
```

Figure 1. Demonstration of using the time and concurrent.futures libraries, including the ThreadPoolExecutor function.

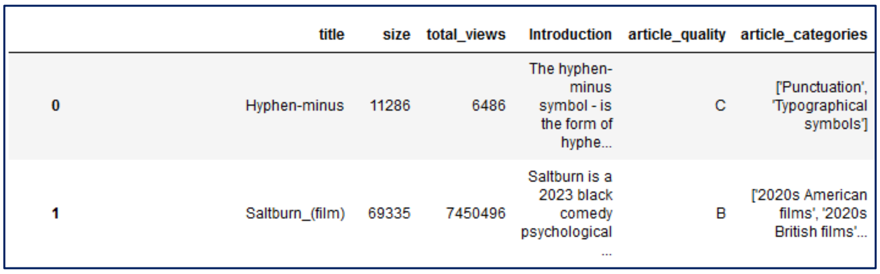
Overview of Challenges and Solutions

As we moved forward, obstacles were bound to arise. We faced challenges with rate limits, scraping pitfalls, and the overwhelming amount of data, which pushed us to constantly refine the fetching process. We utilized the time library to monitor our progress, dividing the task into 60 portions to handle the computational requirements more effectively. We implemented a batching strategy that helped us simplify the process.

4.3. The Final Outcome

The code relevant to this section is available in Section 1.3 of our Google Colab file

The final result of this stage was a dataset that contained the needed features for 60,000 Wikipedia articles. This dataset was crucial for our project and the creation of the automated "See Also" system. Looking back at this phase, it's evident that our team's determination, flexibility, and smart use of technology helped us overcome the challenges we encountered. This experience not only helped us make progress toward our goal but also taught us important lessons and skills in data collection, data fetching, and data science.



	title	size	total_views	Introduction	article_quality	article_categories
0	Hyphen-minus	11286	6486	The hyphen-minus symbol - is the form of hyphe...	C	['Punctuation', 'Typographical symbols']
1	Saltburn_(film)	69335	7450496	Saltburn is a 2023 black comedy psychological ...	B	['2020s American films', '2020s British films'...

Figure 2. First Two Rows from The Final Dataset

5. Rationale For the Group's Approach To Exploring the Data

The code relevant to this section is available in Section 2 of our Google Colab file

5.1. Overview of Exploratory Data Analysis (EDA) for Our Project

Our project utilizes Exploratory Data Analysis (EDA) to thoroughly examine and understand the intricacies of our dataset. Our EDA strategy is designed to simplify the system creation by focusing on the different distributions of important features, such as article category, size, quality, and viewership. Our approach guarantees that we have a dataset that is both comprehensive and diverse across various dimensions. This is essential for the unbiased functionality of our "See Also" recommendation system. Our goal is to provide recommendations to Wikipedia readers that are not only relevant but also popular and packed with valuable content to enhance their reading experience.

5.2. An In-depth Examination of Article Size

The code relevant to this section is available in Section 2.1 of our Google Colab file

Exploring the article size as our initial point of focus, we came across interesting patterns within the distribution of sizes. There were about 12K articles that were between 13 and 26 kilobytes (KB) in size, and approximately 10K articles fell within the 26 to 40 KB range. Our analysis of the data uncovered a consistent pattern of medium-sized articles in our collection. It is worth mentioning that the importance of article size goes beyond just data points. It can be seen as an indicator of the depth of the content. We wanted to make sure that the articles recommended in the "See Also" sections were valuable and provided a thorough and informative browsing experience for the reader.

During our EDA, we encountered a crucial decision point when examining articles with less extensive content. Based on our analysis, we discovered that 1,089 articles fell below the 6 KB threshold. This threshold was used to identify articles that could be considered very short stubs that had very limited information. After careful evaluation, we determined that these articles are not suitable for recommendation. Their content is limited and lacks the necessary depth of information and context that we believe is essential for the reader. We made the decision to exclude these articles, as explained in section 7.3.

For a visual representation of our analysis, with the dashed line indicating the exclusion point at 6 KB. Please refer to Figure 4 in Appendix A (List of Figures).

5.3. An In-Depth Examination of Article Quality Distribution

The code relevant to this section is available in Section 2.2 of our Google Colab file

As we delved into our Exploratory Data Analysis (EDA), we found that the quality of Wikipedia articles played a crucial role in improving our "See Also" recommendation system. The quality ratings provided by Wikipedia Xtools offer a straightforward way to evaluate the depth and reliability of articles, which in turn affects their suitability for recommendation. We carefully evaluated each quality type to ensure that the content met our standards for rigor and editorial completeness. These factors were essential in our selection process. Provided is a concise overview of each quality category displayed in the provided chart above, as defined by Xtools:

- **FA (Featured Articles):** These articles are considered the best of the best on Wikipedia, having gone through a rigorous review process. These articles showcase the finest work on Wikipedia and adhere to the highest standards of content, references, and presentation. [6]

- **A:** Articles in this category are well-organized and essentially complete, providing extensive coverage of their topic, without the exhaustive review process required for Featured Article status. [6]
- **GA (Good Articles):** These articles have been acknowledged for their quality, although they have not yet reached the status of Featured Articles. The articles have undergone review and meet the necessary standards, although they may not be as extensive as Featured Articles or A-class articles. [6]
- **B:** B-class articles are decently comprehensive, covering major aspects of the topic without the depth found in higher-rated articles. They are generally free of major issues and are well-cited but might require some further work to reach Good Article standards. [6]
- **C:** C-class articles possess a basic level of content development and are fairly structured but lack some important content and may be under-referenced. This category makes up the bulk of our dataset, suggesting a moderate quality that balances comprehensiveness with potential for improvement. [6]
- **Start:** These articles are developing but still quite incomplete and poorly structured, often lacking sufficient sources. They represent the starting point for articles in the process of being expanded. [6]
- **Stub:** Articles classified as Stub are typically quite basic, often consisting of only a few sentences, and they offer basic details and act as placeholders to indicate the need for further development. [6]
- **Others:** This category encompasses articles that do not fit into the conventional quality scale, including lists and articles yet to be assessed for quality. They may vary widely in content depth and organization but lack a formal quality rating. [6]

During our analysis, we took into account the possibility of excluding stub articles due to their potential lack of sufficient content depth for our recommendation system. Upon closer inspection, it was discovered that certain stub articles had unexpectedly good content. After conducting a thorough analysis, we decided to adjust our exclusion strategy by considering the article size only. We implemented a threshold for articles smaller than 6 kilobytes, as discussed in section 5.2.

The most frequent quality rating in our dataset was 'C', which indicates a reasonable and acceptable standard. These C-class articles offer a straightforward and basic overview of the topic, leaving plenty of room for improvement. The dataset contains around 21K articles that belong to this class.

For a visual representation of the article quality ranking analysis, please refer to Figure 5 in Appendix A (List of Figures).

5.4. An In-Depth Examination of Article Category Distribution

The code relevant to this section is available in Section 2.3 of our Google Colab file

Wikipedia's content is incredibly diverse, covering a wide range of topics from the natural sciences to the arts. The articles are carefully categorized to create a comprehensive web of knowledge. These categories are essential for our project, serving not only as a way to organize information but also as important data points that inform the "See Also" recommendation system. We focused on extracting article categories for two important reasons:

1. To ensure our dataset reflects the diverse landscape of Wikipedia, encompassing a wide array of subjects rather than being confined to a niche field.
2. To pair these categories with the text of the article's introduction and title, forming a singular, rich text corpus from which BERT vectors can be derived.

We needed to come up with a systematic way to classify articles based on the keywords that are in their categories, and dedicated a significant amount of time to determining a collection of words that would enable us to do so. We used a wide range of terms in our taxonomy, including specific ones like 'democracy' and 'election' in politics, as well as 'biochemistry' and 'ecology' in science. We made sure to include articles from various fields to create a well-rounded and diverse dataset for our system. Here are the main categories we identified, along with a sample of the defining keywords we used for each:

- **Entertainment**: Keywords included 'movies', 'music', 'television', 'theater', 'comedy', 'dance', 'celebrities', and 'video games'.
- **Literature**: For this category, terms like 'novel', 'poetry', 'writer', 'drama', and 'literary genre' were used.
- **Sports**: This category was tagged with words such as 'olympic sport', 'football', 'basketball', 'tennis', 'athletics', and 'soccer'.
- **History**: Here, we used 'historical event', 'ancient history', 'medieval history', 'world war', and 'renaissance'.
- **Technology**: Keywords like 'software', 'hardware', 'internet', 'ai', and 'robotics' were selected.
- **Politics**: We classified articles with terms including 'politician', 'election', 'government', 'democracy', and 'political party'.
- **Health**: 'Medicine', 'nutrition', 'disease', 'psychology', and 'wellness' were among the keywords.
- **Science**: Words such as 'biology', 'physics', 'chemistry', 'astronomy', and 'environment' defined this category.

After examining the distribution of general categories, we discovered that a significant portion of our dataset, roughly 19,000 articles, belonged to the 'Entertainment' category. 'Literature' came in second place with approximately 10,000 entries. Our collection encompasses a wide range of articles, including those in the categories of 'Politics,' 'History,' and 'Technology,' which further demonstrates its depth and variety. Our system's wide variety of recommendations ensures that users receive diverse and comprehensive suggestions that span across different genres and disciplines.

For a clear visualization of the article counts by general category, please refer to Figure 6 in Appendix A (List of Figures).

5.5. An In-Depth Examination of Article Introduction

As part of our data collection process, we started by collecting the introductions of Wikipedia articles. We were able to extract this important section of each article efficiently using web scraping. The introduction is a crucial part of any article. It provides an overview of the main points and gives the reader a sense of the context. Our initial investigation into these introductions uncovered a notable difference in length: while certain articles provided extensive overviews, others simply gave concise preludes to their subjects. The variation in the length of introductions revealed a difficulty in guaranteeing that our "See Also" recommendations would consistently offer a concise overview of related articles.

Enhancing Contextual Understanding Through Category Integration

Based on this observation, we made a significant decision in our methodology. In order to provide a comprehensive system, we needed to go beyond just the article introductions. By integrating categories into the natural language processing phase of our semantic vector generation, we can address the conciseness of certain introductions and enhance the contextual information. The integration of categories ensured that even if an article's introduction was short, the semantic vectors generated would still capture the themes and subjects it was associated with. The full process is detailed in section 6.3.

5.6. An In-Depth Examination of Article Views

The code relevant to this section is available in Section 2.4 of our Google Colab file

Our automated "See Also" system was developed with the goal of providing dynamic and up-to-date recommendations. To achieve this, we integrated monthly view data from Wikipedia into our system. This method enables the system to stay up-to-date with the current trends and interests of readers. We collected articles' views for January 2024 since we started working on the system in February 2024.

The popularity of an article is influenced by various factors, including the quality of content, the relevance of the topic, and the level of interest from Wikipedia readers. Our system was designed to prioritize articles with higher view counts, ensuring that we stay in tune with the public's knowledge demands. In Section 7.5 of this report, we provide an overview of how we utilized views data as a filter.

As we explored the views data, we observed a clear distribution through a histogram analysis. The dataset we used mainly included articles that were widely read. Figure 7 in Appendix A clearly displayed that most articles had view counts significantly exceeding 20,000 in January

2024. It is also worth mentioning that our collection contains less than 500 articles that have less than 20K views in the same month.

6. Rationale for Data Modelling to Generate Semantic Vectors

6.1. Exploration and Selection of NLP Models for Generating Semantic Vectors

During the creation of our automated "See Also" section system for Wikipedia, one important aspect was to generate precise semantic vectors that could effectively capture the essence of each article in a comprehensive manner. Once our dataset was ready, we began a thorough investigation of different Natural Language Processing (NLP) models. We aimed to choose a model that could effectively interpret and process the text, extracting semantic vectors that capture the subtle semantic meanings found in the articles. Some of the models we explored:

BERT (Bidirectional Encoder Representations from Transformers)

Google developed BERT to process words in context, enabling it to capture contextual nuances more effectively than other models. The model incorporates a technique known as Masked LM (MLM) and Next Sentence Prediction to enhance its comprehension of context and sentence relationships. This is particularly important for tasks such as summarization and question answering, where contextual understanding is crucial. [7]

GPT (Generative Pre-trained Transformer)

Although GPT is incredibly efficient at generating text using its transformer-based architecture, it has certain limitations when it comes to extracting semantic meaning like BERT does. OpenAI's GPT excels in content creation but faces challenges when it comes to tasks that demand a deep understanding of context and the relationships between words. This understanding is essential for accurately connecting articles that are semantically similar. [8]

RoBERTa

RoBERTa, also known as the Robustly Optimized BERT approach, is a powerful tool. This version of BERT, created by Facebook AI, makes adjustments to important hyperparameters, eliminates the pretraining objective of predicting the next sentence, and undergoes training with larger mini-batches and learning rates. The modifications have been made to enhance BERT's reliability and precision, resulting in RoBERTa's improved capability to handle diverse datasets and minimize the risk of overfitting on specific tasks or datasets. [9]

DistilBERT

DistilBERT, developed by Hugging Face, is a simplified version of BERT. It aims to improve processing speeds by reducing the number of parameters by 40%. However, this reduction may come at the expense of its ability to comprehend complex information. This model is designed to prioritize resource efficiency while still being capable of handling complex NLP tasks. [10]

XLNet

In contrast to BERT, XLNet utilizes a training method based on permutations, which avoids assuming word independence. XLNet, a collaboration between Google and Carnegie Mellon University, has demonstrated superior performance compared to BERT on various NLP benchmarks. It achieves this by effectively managing the intricacies of language structure. However, for our specific requirement to rank similarity and relevance, BERT's directional approach was more suitable because of its straightforward and efficient handling of sentence context. [11]

ERNIE (Enhanced Representation through kNowledge Integration)

Baidu has developed ERNIE, which incorporates external knowledge bases into its pre-training tasks to improve understanding. This feature is especially beneficial for recognizing specific entities. Nevertheless, when it came to our basic semantic analysis needed for connecting Wikipedia articles, the added intricacy of ERNIE did not result in substantial advantages, making it less crucial for our specific use case. [12]

6.2. Choosing BERT

We decided to implement BERT based on a number of important factors that came up during our evaluation of different models: [7]

- **Contextual Awareness:** BERT's design to read text bidirectionally (considering all surrounding context simultaneously) allows it to understand the context of each word in a way that is superior for tasks requiring a deep understanding of semantic relationships.
- **Accuracy in Semantic Similarity:** BERT has demonstrated high effectiveness in identifying semantic similarities, a core requirement for accurately linking articles in the "See Also" section based on content relevance.
- **Computational Efficiency:** Despite its complexity, BERT offers a good balance between performance and computational efficiency, which is essential given the large volume of data we needed to process.

During the investigation of these NLP models, it became clear that selecting the right tool is crucial. The extensive capabilities of BERT in handling contextual data and its strong support ecosystem made it the perfect choice for our project. This ensures that our automated "See Also" suggestions are not only relevant but also insightful, greatly improving the user experience on Wikipedia.

6.3. Data Modeling and Challenges in Semantic Vector Generation

The code relevant to this section is available in Section 3.1 of our Google Colab file

During the data modeling phase of our project, we encountered several challenges when using BERT to generate semantic vectors. We developed a straightforward method where we combined the title, introduction, and categories of each article into one string. This string was then used to generate semantic vectors.

Overview of the Process

The first step was to preprocess the data, ensuring its completeness and consistency. This involved removing articles with missing introductions and restructuring the DataFrame for accurate indexing. We have split our dataset into two sets of 30,000 articles each to improve processing efficiency and effectively manage system resources.

We used the *bert-base-nli-mean-tokens* model from BERT for the semantic analysis. This model is well-known for its ability to generate strong embeddings. Our model was instrumental in converting the text of each article into semantic vectors, each consisting of 768 dimensions. These vectors play a vital role in capturing the profound semantic meanings required to efficiently connect related articles in the "See Also" section.

Difficulties Faced

One significant obstacle we encountered was the high computational demand of BERT, despite its reputation for being efficient. The processing time for generating embeddings was significant. For example, it took seven hours to create embeddings for a batch of 30,000 articles. We incorporated Python's time library into our monitoring process to effectively track the duration and ensure uninterrupted operations.

Resource management was also a notable concern. The output vectors had a high dimensionality and the computations required significant memory and processing power. We had to carefully consider these requirements to avoid system overloads or crashes, which could result in data loss.

BERT Embeddings Validation through PCA

The code relevant to this section is available in Section 3.2 of our Google Colab file

During our analysis of BERT embeddings using Principal Component Analysis (PCA), we curated a list of articles from various categories. We were able to evaluate the effectiveness of BERT by ensuring a comprehensive representation of Wikipedia content. The scatter plot (Figure 8 in Appendix A) generated from the PCA analysis clearly illustrated the clustering of articles that share common themes and categories. It is worth noting that there were clear clusters in the articles related to sports, universities, data and computer science, and social media. This suggests that BERT was able to accurately capture semantic relationships. This observation emphasizes the strength and potential of BERT for improving the automation of creating "See Also" sections in Wikipedia articles.

7. Detailed Workflow of the Automated "See Also" System

The code relevant to this section is available in Section 4 of our Google Colab file

Please refer to Figure 9 in Appendix A for an Overview of the System Workflow

7.1. Input Article

The system begins with an input article, which serves as the foundation for the following recommendations.

7.2. Calculating Cosine Similarity

The code relevant to this section is available in Section 4.1 of our Google Colab file

The system uses BERT embeddings to determine the cosine similarity between the semantic vectors of the input article and other articles in the dataset. The cosine similarity metric is a useful tool for gauging the level of similarity between articles. It plays a crucial role in identifying the most relevant articles to include in the "See Also" section.

Reasons Behind Choosing Cosine Similarity

In the complex realm of semantic analysis, choosing the most suitable measure of similarity between high-dimensional vectors is crucial. Our system utilizes cosine similarity as the primary metric for a variety of reasons that make it a compelling choice.

The cosine similarity is a measure that calculates the cosine of the angle between two vectors. It provides information about the orientation of the vectors in relation to each other, regardless of their magnitude. This property makes it ideal for text analysis, especially when dealing with articles of varying lengths, which is a common occurrence due to the natural variation in article lengths in Wikipedia. The cosine similarity remains unaffected by the length of the vectors, placing its emphasis solely on the directionality of the vectors. This feature was crucial for our system because articles often have different introduction lengths. What matters most is the direction or semantic orientation of the content, rather than its amount. [13]

Throughout our initial investigation, we delved into various metrics of similarity, including the Jaccard similarity and Euclidean distance. For our high-dimensional and sparse BERT vectors, the Jaccard similarity may not be the most suitable measure of similarity. This is because it calculates similarity based on common attributes, and in our case, the absence of common attributes could result in an underestimation of similarity. The Euclidean distance calculates the direct distance between two points in a multi-dimensional space. It is important to note that it can be heavily influenced by the magnitude of the vectors, which, as we have discussed, can vary significantly between articles. [13]

7.3. Filtering Out Small Articles

To ensure that the recommendations have substantial content, the system excludes articles that are less than 6,000 bytes in size. It is important to include this exclusion criterion to prevent readers from being directed to incomplete or insufficient articles. After that, the system chooses the top 20 articles that have the highest cosine similarity scores compared to the input article.

7.4. Sorting by Number of Shared Categories

The code relevant to this section is available in Section 4.2 of our Google Colab file

Afterward, the system calculates the number of shared categories between the input article and the other articles and sorts these 20 articles according to it. This is done because a higher number of common categories generally suggests a closer topical connection. Following this categorization process, the system proceeds to narrow down the list to the top 10 articles for additional assessment.

7.5. Arranging by Views and Quality, and Final Selection

The code relevant to this section is available in Section 4.3 of our Google Colab file

At the end of the process, the system organizes the top 10 articles by total views and assigns a numerical ranking based on their quality. The ranking is determined by a pre-defined quality mapping system (Explained in detail in section 5.3). Different article quality categories, ranging from Featured Articles to articles of unspecified quality, are assigned scores. The highest score is given to Featured Articles, while categories that are not recognized receive a score of 14. The system selects the top 5 articles based on views, and sorts them by their quality score, to be the final recommended articles for the "See Also" section.

8. Approach for Evaluating the Model

8.1. Overview of Early Obstacles in Model Evaluation

Assessing the efficiency of our automated "See Also" system posed considerable difficulties at the beginning. Our team dedicated a significant amount of time to thoroughly researching and selecting the most suitable methodology for accurately assessing the model. Throughout this phase, we reached out to the research team at the Wikimedia Foundation to get advice. [redacted]

8.2. Methods for Evaluation Consideration

After careful consideration and collaboration with the Wikimedia Foundation's research team, we have identified three possible methods to assess our model:

1. **Clickstream Data Analysis:** This approach entails implementing the system and examining shifts in the clickstream data to observe differences in the number of clicks on "See Also" links before and after our recommendations are integrated. Despite its effectiveness in showcasing the system's influence on user behavior, we decided against using this approach due to time constraints.
2. **Feedback from Wikipedia Specialists:** An alternative approach was to seek input from dedicated editorial teams within Wikipedia, like those involved in the China Wiki

Project. These teams possess a wealth of knowledge on specific topics and are capable of offering well-informed opinions on the significance and effectiveness of our system's recommendations for related articles. Unfortunately, this approach would involve a lot of back-and-forth communication and would take up a significant amount of time, which is not feasible given our project timeline.

3. **Surveying Wikipedia Editors:** The most feasible option for our timeline involved creating anonymous questionnaires comparing our system's "See Also" suggestions against existing ones in randomly selected articles. We planned to distribute these surveys to the 5,000 most active Wikipedia editors, asking them to evaluate which set of recommendations was more relevant and useful.

8.3. Chosen Approach and Implementation Overview

Based on our analysis, we have chosen to move forward with the third evaluation strategy, which involves the use of surveys. This approach was a perfect fit for our project's limitations and provided a straightforward way to gauge the perceived value and relevance of our recommendations from a wide range of Wikipedia contributors. Through a comparison of two anonymous lists of suggestions, we can obtain impartial feedback regarding the effectiveness of our system in enhancing the "See Also" sections in various articles.

Outline of Chosen Methodology for Evaluation

Please refer to Figure 10 in Appendix A for an Overview of the Evaluation Methodology

Choosing Articles

Our evaluation started by choosing 200 articles from our dataset. This selection was chosen at random to provide a broad representation of different Wikipedia categories, including Sports, Politics, Law and Government, Health and Medicine, History and Geography, Science and Education, Business and Companies, Entertainment, Arts and Media, Technology and Internet, and Religion. This was important in order to thoroughly evaluate the system's performance across various subjects.

Surveys' Designing

We created a survey that included two lists of "See Also" articles for each of the selected articles. List A consisted of five articles that were generated by our automated system, while List B comprised articles from the existing manual "See Also" section of the Wikipedia page. This setup was designed to evaluate the efficiency of automated and manual methods in a controlled environment.

Surveys' Questions

The survey asked participants two main questions. The first question was about their preference between two lists (A or B) for finding related information. The second question asked participants to select one or more reasons for their choice from a set of predefined options, which are relevance, popularity, and diversity.

Creating Surveys

The code relevant to this section is available in Section 5 of our Google Colab file

The process of setting up these surveys was filled with difficulties, mainly because of the intricacies involved in creating 200 unique surveys. At first, we developed a code that used web scraping to collect the current articles in the "See Also" section for the randomly chosen articles. A considerable amount of effort went into compiling the lists of articles for each survey. Secondly, we used a Google API script to generate the surveys automatically. This involved investing a significant amount of time in learning how to code using Google's scripting API.

Editors Engagement

We conducted a comprehensive survey by reaching out to the top 5000 Wikipedia editors through a mass messaging campaign. We invited them to participate in surveys that were relevant to their specific areas of expertise, ensuring a wide range of valuable feedback. The purpose of this strategy was to make sure that the feedback collected accurately represented experienced contributors who had well-informed perspectives on the content.

This evaluation was designed to assess the effectiveness of our automated "See Also" system and involve the Wikipedia community in a collaborative and ongoing improvement process. Through an analysis of feedback from a wide range of editors, our aim was to improve our system in order to better support Wikipedia's mission of sharing knowledge in a thorough and precise manner.

8.4. Overview of Survey Participation and Results

The code relevant to this section is available in Section 6 of our Google Colab file

The analysis of the automated "See Also" section yielded very good user engagement, as evidenced by the participation metrics and response distribution.

Analysis of User Participation and Timeline

There were a total of 807 responses recorded in response to the surveys distributed among Wikipedia editors. It is worth mentioning that this figure does not reflect the count of distinct participants, as an individual may have taken part in multiple surveys. A mass messaging campaign through email was used to invite participants. The initial invitation was sent on April 1, followed by a reminder on April 7.

The response timeline was focused primarily on the dates of these communications. There was a noticeable increase in responses right after each message was sent. A total of 218 responses were gathered between April 1 and April 2, and an additional 198 responses between April 7 and April 8.

Please refer to Figure 11 in Appendix A for a visual representation of responses per day.

Overview of Survey Participation Distribution

Of the 200 surveys that were distributed, 165 of them received responses, while unfortunately, 35 surveys did not receive any feedback. Out of all the surveys, the ones on 'COVID-19_pandemic' and 'Tiktok' had the most participation, with 27 and 22 respondents respectively. The variation in response rates across different topics underscores the wide range of interests within the Wikipedia editor community.

Please refer to Appendix B for additional information on surveys and their detailed responses.

Summary of Survey Findings

Based on the data collected from the surveys, it was found that out of the 807 responses, 471 respondents (58.36% of the total) preferred to List A over List B. On the other hand, List B was preferred by 336 participants.

Please refer to Figure 12 in Appendix A for the graph related to this part.

Preferences Influencing Choice

Upon examining the preferences influencing these preferences, clear patterns emerged. Around 25% of the participants who favored List A mentioned 'relevancy' and 'popularity' as their rationale. In contrast, a significant portion of List B supporters (approximately 31.5% of those who chose it) chose 'relevancy', 'popularity', and 'diversity' in their decision-making process.

Please refer to Figure 13 in Appendix A for the graph related to this part.

8.5. Detailed Surveys Responses

Please refer to this link for full details

For a more detailed analysis of the survey data, a comprehensive breakdown can be found in Appendix B of this report. The response data for each survey is listed fully, including the number of participants per survey. In addition, we have created a digital resource that can be accessed through the provided link above.

9. Overview of Survey Findings

9.1. Summary of Overall Performance

Based on our analysis of the survey results, it appears that the automated "See Also" sections generated by our system were well-received overall. It is worth mentioning that a significant majority of 58.36% of the responses showed a preference for the lists generated by our automated process (List A). This indicates that our system has the potential to be a practical alternative to the current manual method of creating these sections. The majority preference indicates that our NLP-driven approach aligns with user interests and relevancy criteria, suggesting a successful outcome.

Although it is promising that most people favored our automated system, the fact that just over half (58.36%) chose List A highlights the need for improvement. The ultimate objective is to have a larger majority of people support the automated system, thus confirming its efficiency and accuracy compared to the manual method.

9.2. Article Performance Overview

Upon closer examination of individual articles, we noticed a wide range of responses across various topics, revealing valuable insights into the intricacies of the system's performance. Articles such as "Gaza Strip," "United States," and "Science" received very positive feedback, with the majority clearly expressing a preference for the automated suggestions. Our system was able to effectively identify the topics of these articles.

In contrast, certain articles presented less positive results. Some examples that stand out are "Human," "Mercedes-Benz," and "Google Analytics," where a significant number of participants preferred the current manual "See Also" sections (List B). The difference in system performance suggests that the automated method may face challenges in certain content areas.

9.3. Impact of Article Popularity

Upon further analysis, it was found that there is a clear correlation between the popularity of an article and the success rate of the system. Our system has shown a strong performance with high-engagement content, as evidenced by the improved performance of articles related to popular topics in January 2024. This observation is important and indicates that although the system is effective in dealing with common topics, there is room for improvement in its ability to handle less popular or more obscure articles. The system may have been influenced by the emphasis on the top 60,000 articles that are most commonly discussed, potentially leading to a bias towards more frequently mentioned topics. Including a wider range of articles in the dataset could enhance the system's versatility and accuracy across various topics.

9.4. Analysis of Users’ Preferences

Examining the details of user choices from the survey data, a noticeable split emerged in the rationale behind preferences for List A and List B. In List A, which consisted of automated suggestions, individuals expressed their preferences based on different factors. These factors included relevance, popularity, and a combination of both relevance and popularity. Based on the data, it appears that the automated system was effective in identifying articles that were both relevant and popular among users, leading to its successful performance.

On the other hand, when it came to List B (manual suggestions), the selections were often justified by a combination of reasons. Specifically, 31.55% of the preferences included relevance, diversity, and popularity as the chosen factors, while 17.56% solely focused on relevance. It seems that the participants preferred a comprehensive approach in their manual selections, which may have allowed for a greater range of user interests to be included through diverse content.

10. Conclusion

As we wrap up this project, we look back on our accomplishments, the obstacles we encountered, and the valuable feedback we received from the Wikipedia community.

During the project, our team created and put into action a system that showed significant success. This was evident from the fact that 58.36% of survey participants preferred the automated "See Also" sections compared to the traditional ones. The approval rating is positive, but there is room for improvement since many users still prefer the manual-created ones.

One important finding from this project was the discovery of differences in performance among different articles according to their popularity. Based on our findings, we discovered that by incorporating a variety of data points (articles), we can significantly improve the system and increase user satisfaction. Input from Wikipedia editors and administrators has played a crucial role in enhancing our understanding of how the system can be enhanced to better cater to the diverse needs of Wikipedia users.

In addition, the project faced some significant challenges. We approached each challenge with a strategic mindset, combining innovative problem-solving with thorough evaluation. This allowed us to tackle the vast and varied to meet our specific needs. The process of creating and distributing numerous surveys, and then analyzing this extensive dataset, was quite overwhelming. However, these efforts played a vital role in giving us the valuable feedback required to improve our approach.

Overall, this project has been a valuable learning opportunity for the entire team, offering deep insights into the intricate workings of content recommendation systems. We successfully approached the research question, and we are also pleased to report that ongoing improvements and robust collaboration from the Wikipedia and free Knowlodge community have left us feeling optimistic about our project.

11. Recommendations and Future Directions

As we progress from the successful deployment of our automated "See Also" system, we are thrilled to explore several important recommendations and plans for the future. This section highlights our ongoing efforts to enhance the functionality of our system, both within the Wikipedia community and beyond.

11.1. Enhancing the "See Also" Sections

Based on our analysis of the 60,000 articles, it was found that more than 29,000 of them lack a "See Also" section at present. Based on discussions with administrators at Wikipedia, it has been suggested that our system is capable of addressing this issue effectively. For the sake of transparency and to ensure ongoing quality control, we suggest including a note in the edit summary when making edits using our system. This note should indicate that the "See Also" section was generated automatically. This will notify other editors about the section, indicating that it may require additional review and potential corrections.

11.2. Adopting the System by House of Wisdom 2.0 Project

Our system has already shown its worth by being integrated into projects, like the House of Wisdom 2 initiative by the Ideas Beyond Borders Foundation. The team is currently utilizing our system to streamline the process of identifying articles on specific topics. Their main objective is to translate English Wikipedia articles into Arabic. Our system has greatly simplified their workflow by cutting down on the time they spend on manual searches. It now offers them precise article suggestions tailored to their specific requirements. In addition, our system has been adopted by translation volunteers on Wikipedia.

[redacted]

[redacted]

[redacted]

11.4. Enhancements

Nowadays, we are actively working to increase the size of our project's database by incorporating all Wikipedia articles. This expansion will enhance the coverage and utility of the "See Also" system, making it even more valuable. In addition, we are investigating cutting-edge cloud computing solutions to streamline and consistently update the system. Regular updates are necessary to ensure that the "See Also" suggestions remain relevant and up-to-date. This involves incorporating new articles and refreshing existing article views on a monthly basis.

11.5. In summary

We have received extremely positive feedback and have been delighted to see various groups within the Wikipedia community proactively adopting our system. Our ongoing efforts involve improving the tool's precision, broadening its scope, and guaranteeing its effectiveness in serving the Wikipedia community.

[redacted]

[redacted]

[redacted]

[redacted]

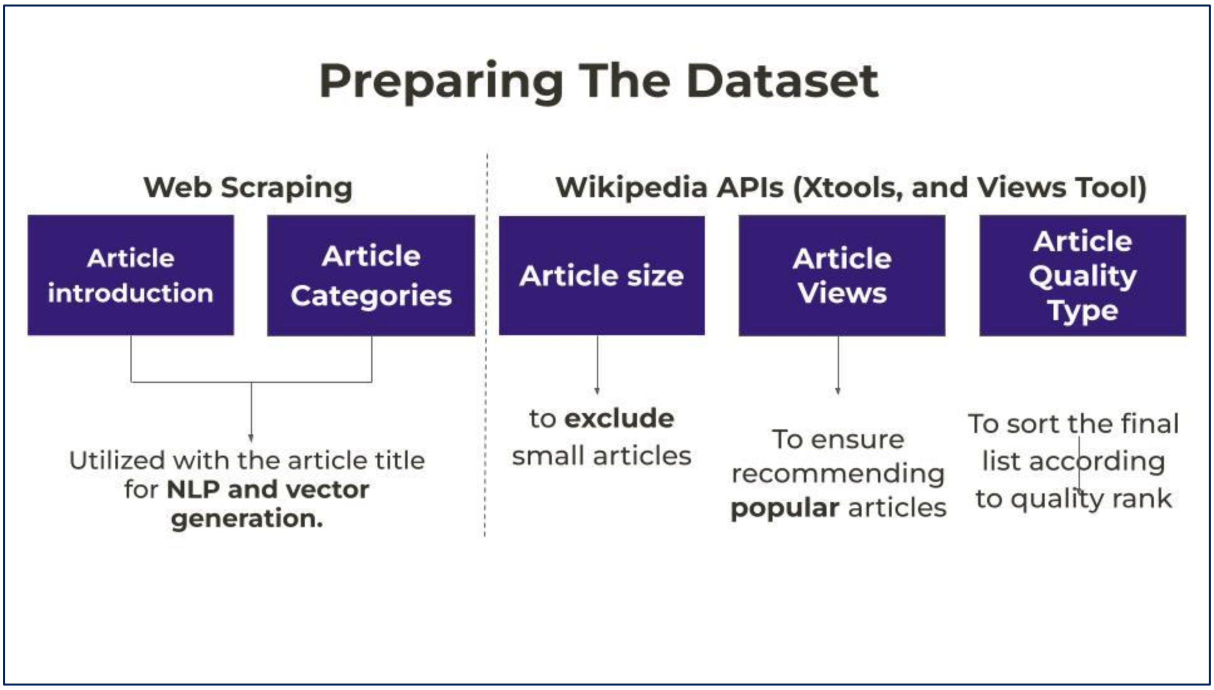
[redacted]

[redacted]

[redacted]

13. Appendices

13.1 Appendix A (List of Figures)



Preparing The Dataset

Web Scraping

Article introduction

Article Categories

Utilized with the article title for **NLP and vector generation.**

Wikipedia APIs (Xtools, and Views Tool)

Article size

to **exclude** small articles

Article Views

To ensure recommending **popular** articles

Article Quality Type

To sort the final list according to quality rank

Figure 3. Preparing The Dataset Overview

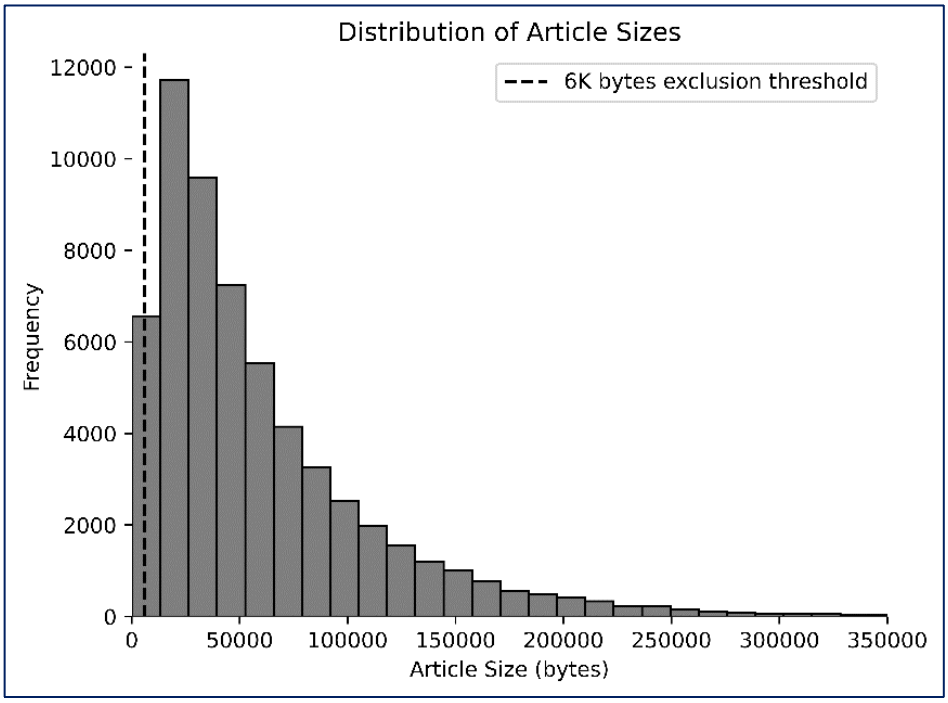


Figure 4. Distribution of Article Sizes

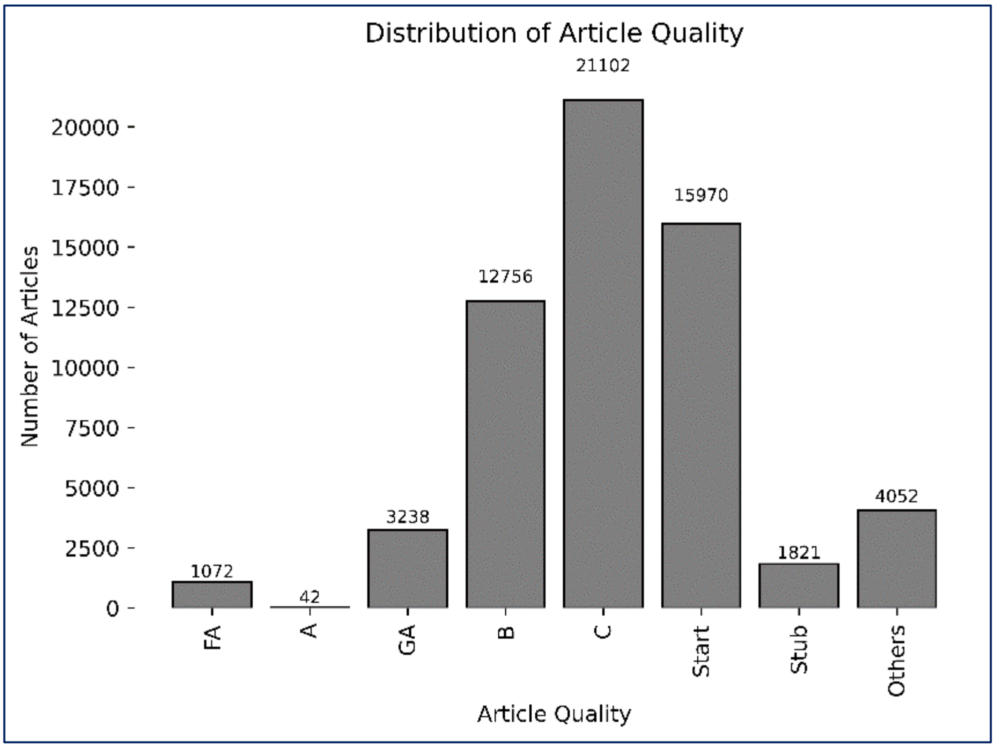


Figure 5. Articles' Quality Distribution

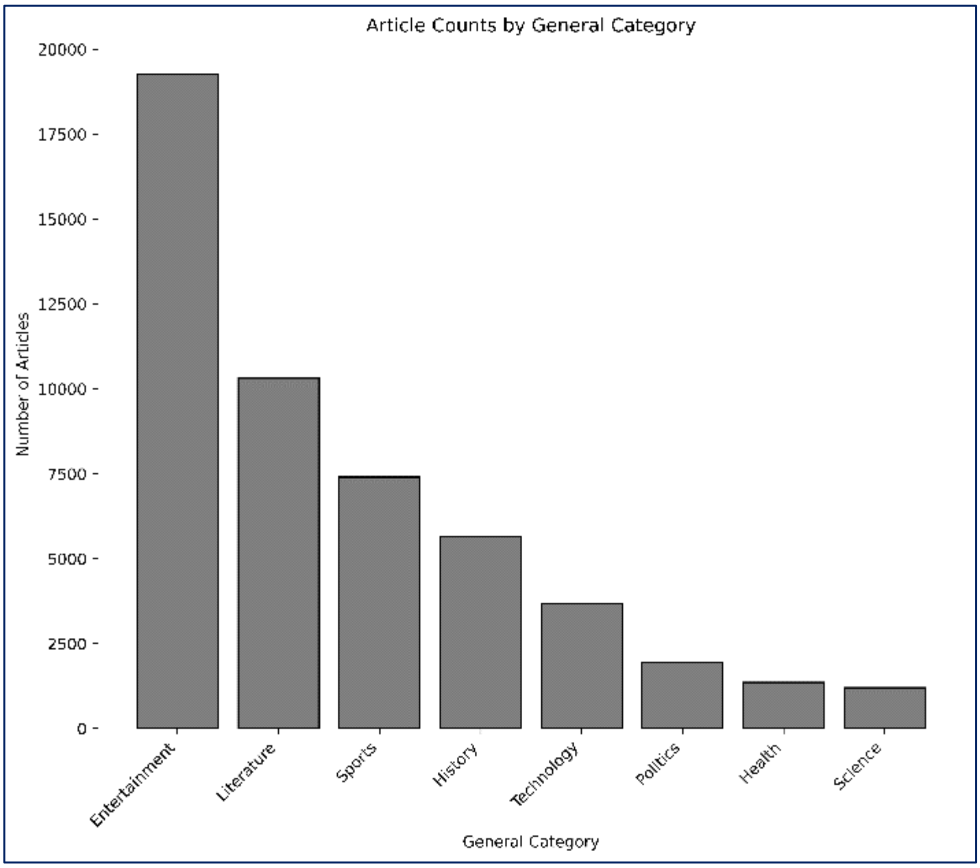


Figure 6. Article Counts by General Category

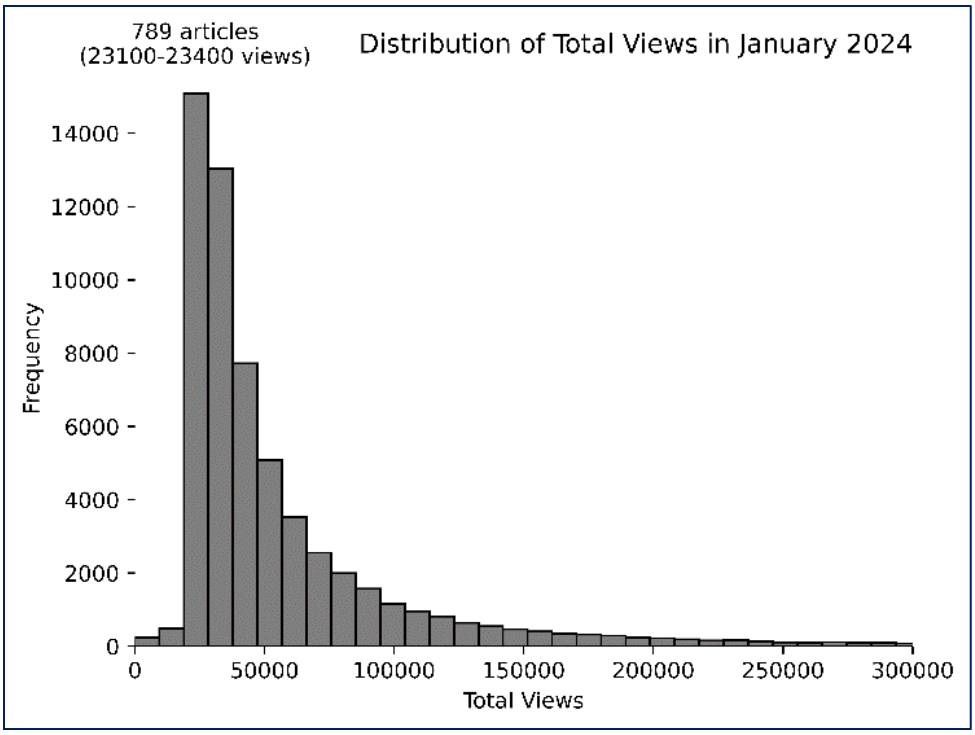


Figure 7. Distribution of Total Views in January 2024

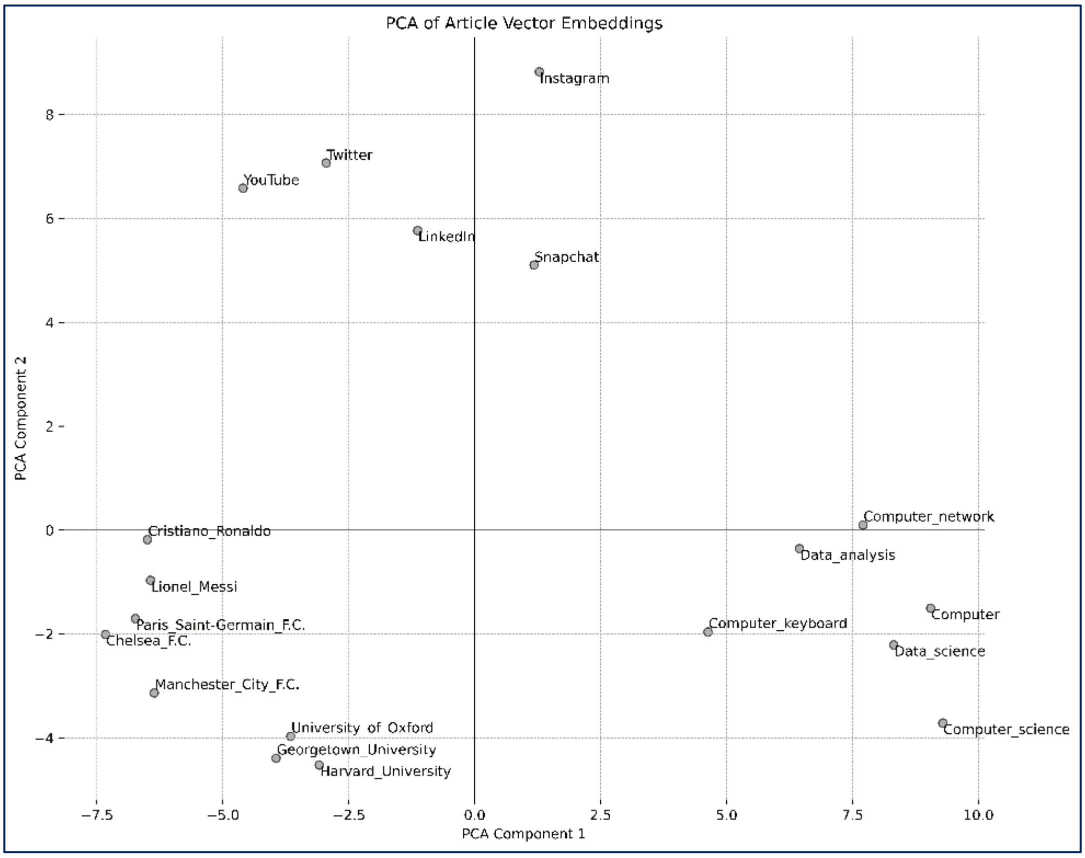
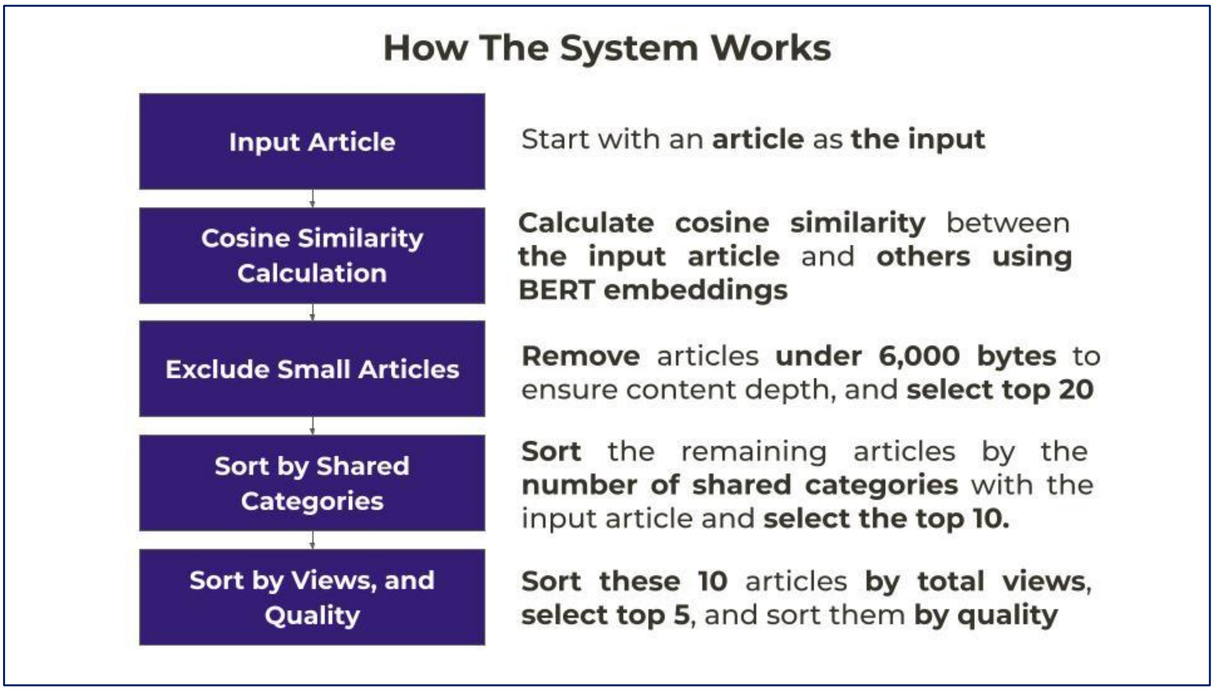


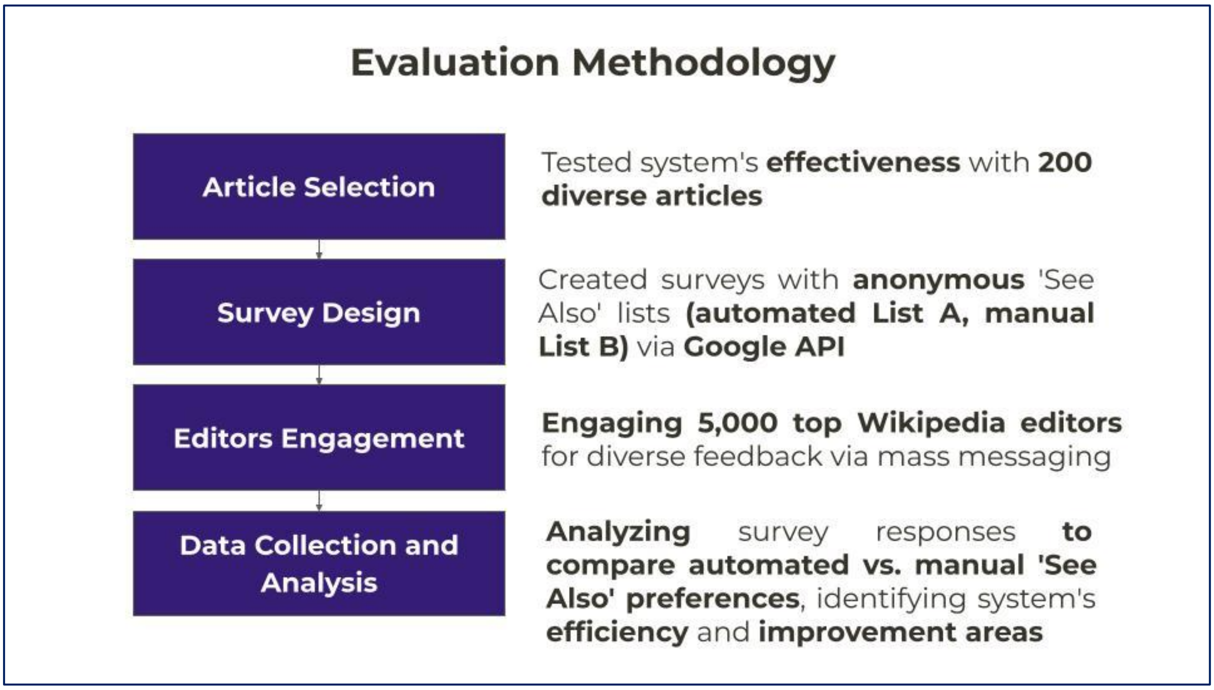
Figure 8. PCA of Article Vector Embeddings



How The System Works

Input Article	Start with an **article** as **the input**
Cosine Similarity Calculation	**Calculate cosine similarity** between **the input article** and **others using BERT embeddings**
Exclude Small Articles	**Remove** articles **under 6,000 bytes** to ensure content depth, and **select top 20**
Sort by Shared Categories	**Sort** the remaining articles by the **number of shared categories** with the input article and **select the top 10.**
Sort by Views, and Quality	**Sort these 10** articles **by total views, select top 5**, and sort them **by quality**

Figure 9. Overview of the System Workflow



Evaluation Methodology

Article Selection

Tested system's **effectiveness** with **200 diverse articles**

Survey Design

Created surveys with **anonymous** 'See Also' lists **(automated List A, manual List B)** via **Google API**

Editors Engagement

Engaging 5,000 top Wikipedia editors for diverse feedback via mass messaging

Data Collection and Analysis

Analyzing survey responses **to compare automated vs. manual 'See Also' preferences**, identifying system's **efficiency** and **improvement areas**

Figure 10. Evaluation Methodology

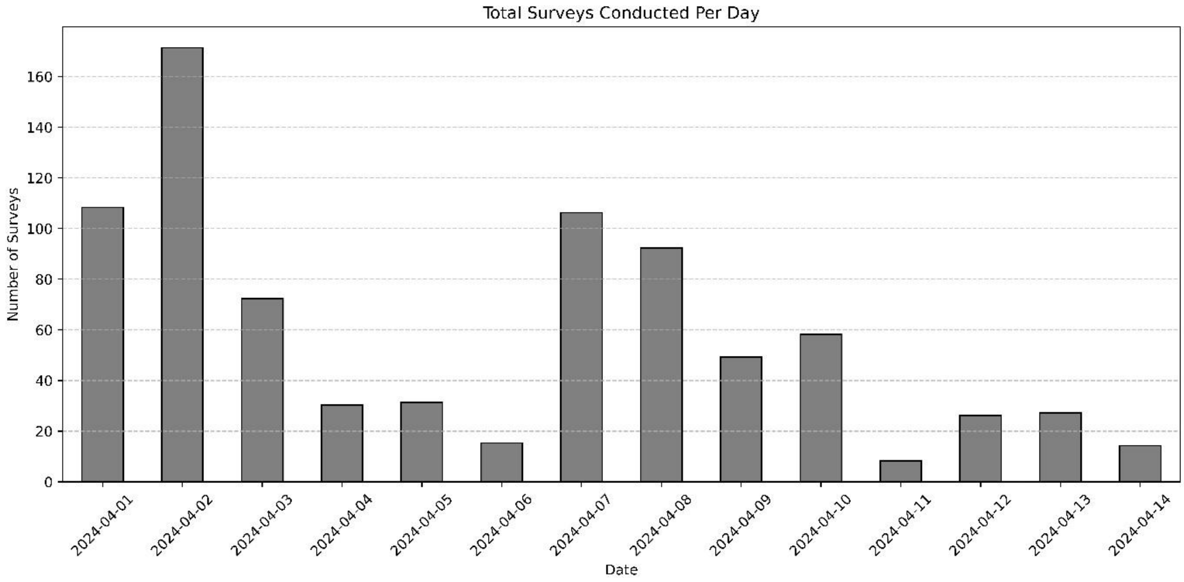


Figure 11. Total Surveys Conducted Per Day

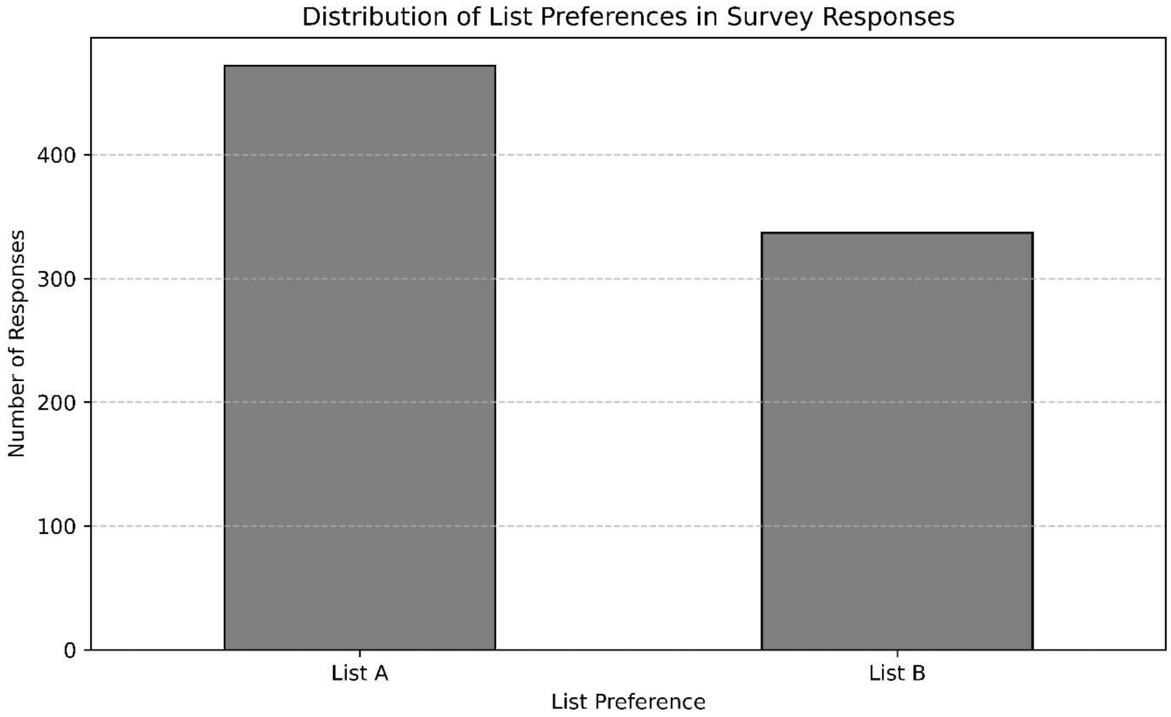


Figure 12. Distribution of List Preferences in Survey Responses

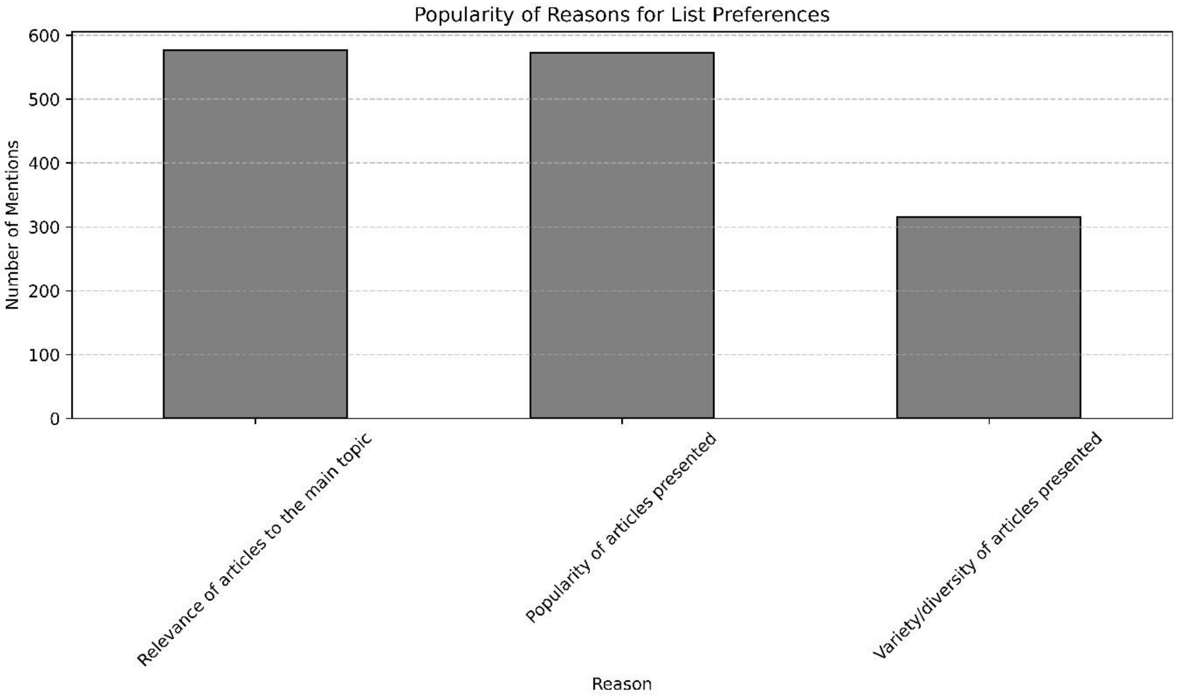


Figure 13. Popularity of Reasons for List Preferences

13.2 Appendix B (Surveys Details)

Useful Links

- A Sheet for All Surveys Responses, Compiled: https://docs.google.com/spreadsheets/d/13HLztr4U3wDcILunVCWBvJCO0UTo5Xp9pi1og5Hvan0/edit?usp=sharing
- Surveys Page in Wikipedia: https://en.wikipedia.org/wiki/User:Mohammad_Hijjawi/Survey_Automating_Wikipedia_%27See_Also%27_Sections_Project_Surveys

Detailed Surveys Responses

Article Name/Survey	Category	#Responses	A	B
COVID-19_pandemic	Health and Medicine	27	24	3
TikTok	Entertainment, Arts, and Media	22	19	3
American_Revolution	History and Geography	19	16	3
PlayStation_2	Technology and Internet	18	16	2
ChatGPT	Technology and Internet	16	2	14
Gaza_Strip	History and Geography	16	16	0
Tesla,_Inc.	Business and Companies	16	7	9
Great_Britain	History and Geography	16	10	6
Xbox_360	Technology and Internet	16	16	0
Climate_change	Science and Education	14	14	0
World_War_II	History and Geography	13	12	1
IPhone	Technology and Internet	13	7	6
Spider-Man	Entertainment, Arts, and Media	12	2	10
COVID-19	Health and Medicine	12	6	6
Toy_Story	Entertainment, Arts, and Media	12	2	10
United_States	History and Geography	11	10	1

UEFA_Champions_League	Sports	11	9	2
One_Direction	Entertainment, Arts, and Media	11	10	1
History_of_pizza	Science and Education	11	9	2
Chess	Entertainment, Arts, and Media	10	4	6
Hamas	Politics, Law, and Government	9	8	1
2024_United_States_Senate_elections	Politics, Law, and Government	9	2	7
Video_game	Entertainment, Arts, and Media	9	6	3
The_Pursuit_of_Happyness	Entertainment, Arts, and Media	9	1	8
William_Shakespeare	Entertainment, Arts, and Media	8	2	6
Attack_on_Pearl_Harbor	History and Geography	8	5	3
Science	Science and Education	8	8	0
Independence_Day_(India)	Politics, Law, and Government	8	1	7
Religion_in_India	Religion	8	4	4
Texas_Revolution	History and Geography	8	6	2
Machine_learning	Science and Education	7	0	7
Mercedes-Benz	Business and Companies	7	0	7
Indian_Rebellion_of_1857	History and Geography	7	2	5
Newton's_laws_of_motion	Science and Education	7	2	5
Windows_10	Technology and Internet	7	4	3
Binance	Business and Companies	7	6	1

The_Guardian	Business and Companies	7	7	0
Harry_Potter_and_the_Prisoner_of_Azkaban_(film)	Entertainment, Arts, and Media	7	6	1
Mobile_app	Technology and Internet	7	4	3
Brain_tumor	Health and Medicine	7	2	5
Johnny_Depp	Entertainment, Arts, and Media	6	5	1
English_football_league_system	Sports	6	3	3
BMW	Business and Companies	6	4	2
Red_Bull	Business and Companies	6	4	2
13_Reasons_Why	Entertainment, Arts, and Media	6	5	1
Biryani	Miscellaneous	6	2	4
Bachelor_of_Science	Science and Education	6	3	3
Umrah	Religion	6	6	0
Google_Analytics	Technology and Internet	6	0	6
Dollar	Politics, Law, and Government	6	1	5
German_resistance_to_Nazism	History and Geography	6	2	4
Taylor_Swift	Entertainment, Arts, and Media	5	3	2
Wales	History and Geography	5	3	2
The_Lord_of_the_Rings:_The_Return_of_the_King	Entertainment, Arts, and Media	5	1	4
Theory_of_relativity	Science and Education	5	4	1
HP_Inc.	Technology and Internet	5	1	4
History_of_the_Roman_Empire	History and Geography	5	1	4
Real_Madrid_CF	Sports	4	3	1

Pep_Guardiola	Sports	4	2	2
Alphabet_Inc.	Business and Companies	4	3	1
Microsoft_Excel	Technology and Internet	4	0	4
Abbasid_Caliphate	History and Geography	4	4	0
Chinese_Communist_Party	Politics, Law, and Government	4	2	2
Midjourney	Entertainment, Arts, and Media	4	4	0
AirDrop	Technology and Internet	4	0	4
Camera	Entertainment, Arts, and Media	4	3	1
Super_Mario	Entertainment, Arts, and Media	4	2	2
Eye_color	Entertainment, Arts, and Media	4	3	1
Computer_science	Technology and Internet	4	1	3
Vodka	Miscellaneous	4	0	4
Secularism	Politics, Law, and Government	4	4	0
Federalist_Party	Politics, Law, and Government	4	1	3
Stone_Age	History and Geography	4	4	0
Discrete_mathematics	Science and Education	4	4	0
New_York_Post	Technology and Internet	4	2	2
WinRAR	Technology and Internet	4	0	4
Cell_division	Science and Education	4	2	2
Google_Scholar	Technology and Internet	3	0	3
California	History and Geography	3	2	1

Discord	Technology and Internet	3	1	2
Google_Drive	Technology and Internet	3	0	3
World_War_II_casualties	History and Geography	3	0	3
UFC_Rankings	Sports	3	2	1
Kashmir	History and Geography	3	0	3
Allah	Religion	3	1	2
2018_FIFA_World_Cup	Sports	3	1	2
Balfour_Declaration	History and Geography	3	3	0
Adobe_Inc.	Business and Companies	3	0	3
Nickelodeon	Entertainment, Arts, and Media	3	3	0
C++	Technology and Internet	3	2	1
Microsoft_365	Technology and Internet	3	0	3
MSN	Technology and Internet	3	0	3
Samsung_Galaxy_S_series	Technology and Internet	3	0	3
Goldman_Sachs	Business and Companies	3	2	1
Suicide_by_hanging	Miscellaneous	3	3	0
IPad	Technology and Internet	3	0	3
FIFA_Club_World_Cup	Sports	3	2	1
Information_technology	Science and Education	3	1	2
UEFA_Super_Cup	Sports	3	2	1
Isra'_and_Mi'raj	Religion	3	0	3
DNA_and_RNA_codon_tables	Science and Education	3	3	0
Google_DeepMind	Technology and Internet	3	2	1
History_of_Egypt	History and Geography	3	0	3

FIBA_Basketball_World_Cup	Sports	3	2	1
PyTorch	Technology and Internet	3	2	1
HTTP_404	Technology and Internet	2	1	1
Human	Science and Education	2	0	2
Yahoo!	Technology and Internet	2	0	2
Super_Bowl_XLVIII	Sports	2	1	1
Android_version_history	Technology and Internet	2	2	0
Amazon_Web_Services	Business and Companies	2	0	2
Yandex	Technology and Internet	2	2	0
United_States_Navy_SEALs	Politics, Law, and Government	2	1	1
FIFA_Men's_World_Ranking	Sports	2	2	0
2024_in_public_domain	Miscellaneous	2	2	0
Volleyball	Sports	2	1	1
Go_(programming_language)	Technology and Internet	2	2	0
Adobe_Photoshop	Entertainment, Arts, and Media	2	0	2
Shawarma	Miscellaneous	2	1	1
Adidas	Business and Companies	2	0	2
Heathrow_Airport	Miscellaneous	2	0	2
Doctor_of_Philosophy	Science and Education	2	1	1
Mario	Entertainment, Arts, and Media	2	1	1
Bashar_al-Assad	Politics, Law, and Government	2	1	1
Egyptian_pyramids	History and Geography	2	2	0
SoundCloud	Technology and Internet	2	1	1

Recep_Tayyip_ErdoÄŸan	Politics, Law, and Government	2	2	0
HDMI	Technology and Internet	2	1	1
Software	Technology and Internet	2	1	1
Dior	Business and Companies	2	1	1
Serial_killer	Science and Education	2	2	0
Taxonomic_rank	Science and Education	2	2	0
Community	Miscellaneous	2	2	0
Imperial_College_London	Science and Education	2	0	2
Tourism_in_the_Maldives	History and Geography	2	1	1
Geometric_distribution	Science and Education	2	2	0
Atmosphere	Science and Education	2	1	1
Saudi_Vision_2030	History and Geography	2	2	0
Hair_loss	Health and Medicine	2	2	0
Romantic_music	Entertainment, Arts, and Media	2	1	1
Canon_Inc.	Business and Companies	2	1	1
Bayesian_statistics	Science and Education	2	2	0
United_Arab_Emirates	Politics, Law, and Government	1	0	1
Apollo_11	History and Geography	1	1	0
World_Trade_Center_(1973â€“2001)	History and Geography	1	1	0
Accenture	Business and Companies	1	1	0
Pregnancy	Health and Medicine	1	1	0
Microsoft_Bing	Technology and Internet	1	0	1
Kingdom_of_Kush	History and Geography	1	1	0

Aircraft_in_fiction	Entertainment, Arts, and Media	1	1	0
Isabella_of_France	History and Geography	1	1	0
Manufacturing	Technology and Internet	1	1	0
Arabian_Sea	History and Geography	1	1	0
Hypersonic_weapon	Science and Education	1	1	0
Microscope	Science and Education	1	0	1
Newspaper	Miscellaneous	1	0	1
Father's_Day	Entertainment, Arts, and Media	1	0	1
Palestinian_National_Council	Politics, Law, and Government	1	0	1
Homo_floresiensis	Science and Education	1	0	1
Sea_of_Japan	History and Geography	1	0	1
Cell_nucleus	Science and Education	1	1	0
Turkish_coffee	Miscellaneous	1	1	0
Mayor_of_London	Politics, Law, and Government	1	0	1
Egyptian_pyramid_construction_techniques	History and Geography	1	1	0
Ottoman_Caliphate	History and Geography	1	1	0
2024_Indian_general_election	Politics, Law, and Government	0	0	0
BBC_World_Service	Entertainment, Arts, and Media	0	0	0
Order_of_the_British_Empire	Politics, Law, and Government	0	0	0
Outlook.com	Technology and Internet	0	0	0
Pinterest	Technology and Internet	0	0	0

Mars	History and Geography	0	0	0
Turkmenistan	History and Geography	0	0	0
Web_browser	Technology and Internet	0	0	0
Space_Shuttle_Columbia_disaster	Science and Education	0	0	0
Twilight_(2008_film)	Entertainment, Arts, and Media	0	0	0
Etihad_Airways	Business and Companies	0	0	0
Qatar_national_football_team	Sports	0	0	0
Corruption_Perceptions_Index	Politics, Law, and Government	0	0	0
Calvin_Klein	Business and Companies	0	0	0
Nintendo_Entertainment_System	Technology and Internet	0	0	0
South_Korean_won	Politics, Law, and Government	0	0	0
Baba_ghanoush	Miscellaneous	0	0	0
Infinity_symbol	Science and Education	0	0	0
The_Family_International	Religion	0	0	0
Xerox	Business and Companies	0	0	0
Social_issue	Miscellaneous	0	0	0
Dropbox	Technology and Internet	0	0	0
Korean_People's_Army_Air_Force	History and Geography	0	0	0
Software_versioning	Technology and Internet	0	0	0
Middle_age	History and Geography	0	0	0
Electron_transport_chain	Science and Education	0	0	0
Children's_Day	Miscellaneous	0	0	0
Psychologist	Science and Education	0	0	0

Wastewater_treatment	Science and Education	0	0	0
Labor_Day	Miscellaneous	0	0	0
Space_elevator	Science and Education	0	0	0
SPSS	Technology and Internet	0	0	0
Economy_of_California	Politics, Law, and Government	0	0	0
Linear_interpolation	Science and Education	0	0	0

13.3 Appendix C (Datasets Links)

- Datasets Folder: https://drive.google.com/drive/folders/11vv0jnNOGG1C6C-Rbf-ZY7MK7Bo4FH0p?usp=sharing
- Full dataset without vectors: https://drive.google.com/file/d/11x0z_5MME6Ei0IFE44uC5L4PZgykmAJ1/view?usp=sharing
- Full dataset with vectors: https://drive.google.com/file/d/12-zNp7Eabdu1-mThnNQzUEloacNhaXkG/view?usp=sharing
- Full dataset with see also section articles: https://drive.google.com/file/d/12-oF44ltoteN2yysYs8d3DD7Z-6_x0P-/view?usp=sharing

References

1. Wikimedia Foundation. (2024). *Statistics*. Available at: https://stats.wikimedia.org/#/all-projects
2. Labhishetty, S., Siddiqa, A., Nagipogu, R. and Chakraborti, S., 2017. WikiSeeAlso: Suggesting tangentially related concepts (See also links) for Wikipedia articles. In: A. Ghosh, R. Pal and R. Prasath, eds. Mining Intelligence and Knowledge Exploration. MIKE 2017. Lecture Notes in Computer Science, vol. 10682. Cham: Springer. Available at: https://doi.org/10.1007/978-3-319-71928-3_27
3. Kaffee, L.-A., Vougiouklis, P. and Simperl, E., 2022. Using natural language generation to bootstrap missing Wikipedia articles: A human-centric perspective. *Journal Name*, [e-journal] pp.163-194.
4. Bhuiyan, H., Oh, K.-J., Hong, M.-D. and Jo, G.-S., 2015. An unsupervised approach for identifying the infobox template of Wikipedia article. In: *2015 IEEE 18th International Conference on Computational Science and Engineering*, Porto, Portugal. IEEE, pp.334-338. Available at: https://doi.org/10.1109/CSE.2015.47

5. Ghosh, S., Vinyals, O., Strope, B., Roy, S., Dean, T. and Heck, L., 2016. Contextual LSTM (CLSTM) models for large scale NLP tasks. *arXiv:1602.06291v2 [cs.CL]*. Available at: https://doi.org/10.48550/arXiv.1602.06291
6. Wikipedia contributors, no date, *Wikipedia: Content assessment*, Wikipedia: The Free Encyclopedia. Available at: https://en.wikipedia.org/wiki/Wikipedia:Content_assessment
7. Devlin, J., Chang, M., Lee, K. and Toutanova, K., 2018. *BERT: Pre-training of deep bidirectional transformers for language understanding*. Available at: https://arxiv.org/abs/1810.04805
8. Zhang, L., Wang, M., Chen, L., & Zhang, W. (2022). Probing GPT-3's Linguistic Knowledge on Semantic Tasks. In J. Bastings, Y. Belinkov, Y. Elazar, D. Hupkes, N. Saphra, & S. Wiegreffe (Eds.), Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP (pp. 297–304). Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics. https://aclanthology.org/2022.blackboxnlp-1.24 (doi:10.18653/v1/2022.blackboxnlp-1.24)
9. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs.CL]. https://doi.org/10.48550/arXiv.1907.11692
10. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2020). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108 [cs.CL]. https://doi.org/10.48550/arXiv.1910.01108
11. Li, H., Choi, J., Lee, S., & Ahn, J. H. (2020). Comparing BERT and XLNet from the Perspective of Computational Characteristics. In 2020 International Conference on Electronics, Information, and Communication (ICEIC) (pp. 1-4). Barcelona, Spain. doi: 10.1109/ICEIC49074.2020.9051081.
12. Sun, Y., Wang, S., Li, Y., Feng, S., Chen, X., Zhang, H., Tian, X., Zhu, D., Tian, H., & Wu, H. (2019). ERNIE: Enhanced Representation through Knowledge Integration. arXiv:1904.09223 [cs.CL]. https://doi.org/10.48550/arXiv.1904.09223
13. Zadeh, R. B., & Goel, A. (2012). Dimension Independent Similarity Computation. arXiv:1206.2082 [cs.DS]. https://doi.org/10.48550/arXiv.1206.2082