Enhancing Wikipedia Article “See Also” Section through Article Features and NLP-Generated Semantic Vectors

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?

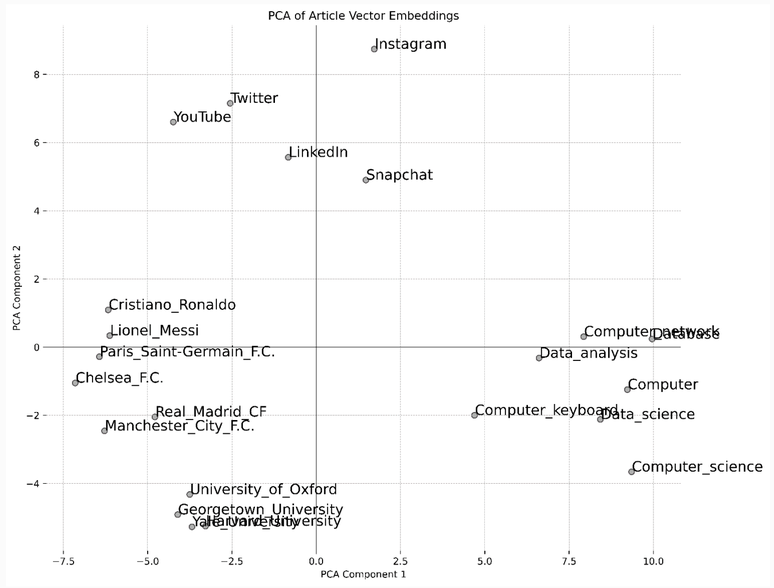
By:

Supervised by:

Abstract and Motivation

In today's rapidly expanding information age, Wikipedia stands out as a primary source of free knowledge with over 25 billion monthly visitors*. Understanding the need to keep up with this growth and ensure information is both relevant and high quality, we've started working on creating a system that automates the updating of the "See Also" sections for Wikipedia's top 60,000 articles. Our project employs machine learning and natural language processing techniques to simplify the discovery and linkage of related articles, aiming to enhance navigability and enrich the reader experience on the site. The evaluation phase of our system is set to begin on 10 March and will last for 3 weeks. Early feedback from Wikipedia's admins has been encouraging, showing that our system could effectively save the site’s editors’ efforts in manually updating these sections.

*Source: Wikimedia Statistics



BERT Embeddings Visualization: Employed PCA to visualize BERT embeddings, ensuring our system accurately groups and recommends related articles, enhancing the 'See Also' section's relevance.

Dataset Preparation Overview

We meticulously compiled a dataset to fuel our automated system, focusing on features that blend popularity with quality. The steps were:

Identified Top 60,000 Articles Titles

Identified Top 60,000 Articles Titles –only-: Based on users clicks dataset to select the most popular titles.

Fetched Article Features (Via Wikipedia APIs’)

We used **Wikipedia APIs** to collect data on **article size**, **views in January 2024**, **Quality Rank**, article’s **categories**, and **introduction**.

Article Categories | Article Introduction | Article Size | Article Views | Article Quality

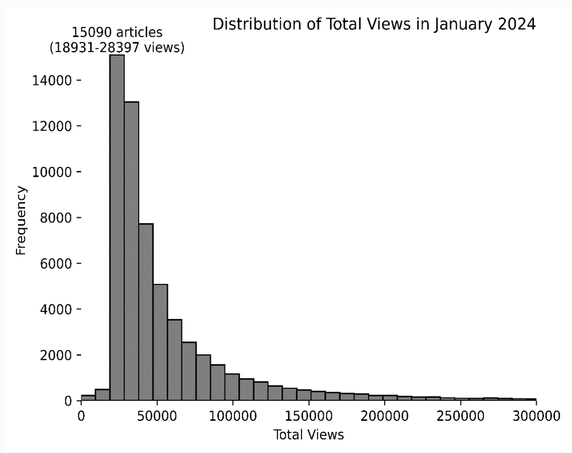
Generate Article Embeddings (Using NLI BERT Model by Google)

We generated the **article embeddings** by utilizing the article’s **title**, **categories**, and **introductory** paragraph by employing the **NLI BERT** model.

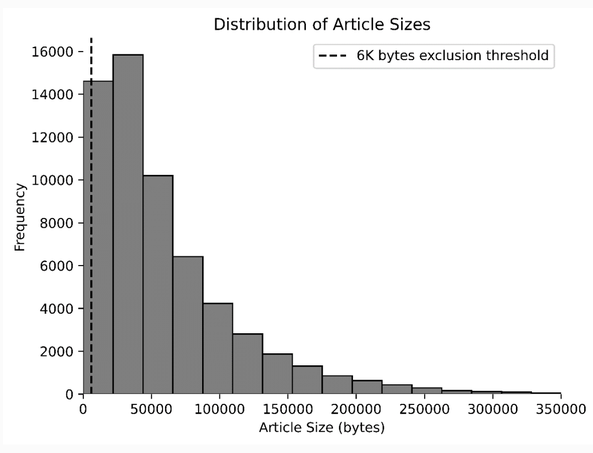
	title	size	total_views	first_paragraph	article_quality	article_categories	bert_0	bert_1	bert_2	bert_3	.
0	Hyphen-minus	11286	6486	The hyphen-minus symbol - is the form of hyphe...	C	['Punctuation', 'Typographical symbols']	-0.628009	0.816350	-0.294891	0.324033	.
1	Saltburn_(film)	69335	7450496	Saltburn is a 2023 black comedy psychological ...	B	['2020s American films', '2020s British films'...	-0.447147	0.665674	0.293427	-0.021366	.

Exploratory Data Analysis

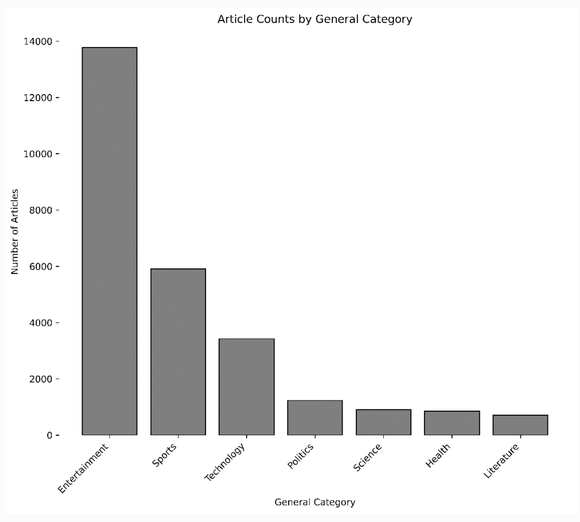
In our Exploratory Data Analysis, we carefully examined the dataset to inform our methodology for automating the 'See Also' section in Wikipedia articles.



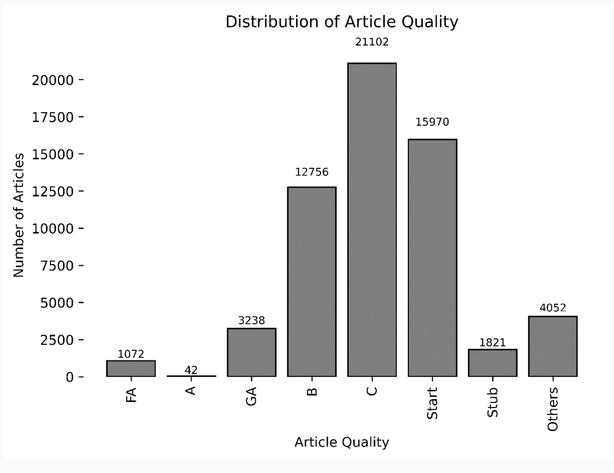
Views Distribution: Assessed to gauge the popularity of articles, confirming our selections resonate with reader interests.



Article Size Distribution: Analyzed to later exclude articles under 6KB, ensuring the recommendations are of substantial content



Categories Distribution: Explored to guarantee a diverse selection across various subjects, enhancing the general applicability of our automated system.



Quality Distribution: Evaluated with a focus on ascending quality, prioritizing 'Featured Article' (FA) as the highest quality, to ensure recommendations are high quality.

Recommendation System Methodology

Outlined below is our system methodology for automating the Wikipedia’s articles 'See Also' sections, leveraging NLP techniques and other features.

Input Article

Start with an **article** as the **input**.

Cosine Similarity Calculation

Calculate **cosine similarity** between the input article and others using BERT embeddings and **select top 20**.

Exclude Small Articles

Remove articles **under 6,000 bytes** to ensure content depth.

Rank by Shared Categories

Rank 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** to identify the most popular and **select the top 5** and **sort them according to their quality score**, and this will be the output

Evaluation, and Results

To assess the effectiveness of our automated "See Also" section generation system and its alignment with Wikipedia's high standards, we have outlined a detailed methodology:

Article Selection

200 articles from **various categories** are chosen to test our system’s **effectiveness** and **accuracy**.

Survey Design

To avoid bias, surveys with **anonymous "See Also" lists —our automated List A** and **manual List B**— are created **using Google API**

Survey Questions

Participants select **their preferred list** based on relevance and popularity, **assessing our system's effectiveness.**

Editors Engagement
(Starting 10 March)

We'll engage **5,000 top Wikipedia editors** via a **mass messaging** campaign for diverse feedback.

Data Collection and Analysis

Survey responses will be analyzed to compare preferences for automated versus manual "See Also" sections, **identifying efficiency**, and **improvement areas** for our system.

Here are some results from our system's performance compared to current manual lists:

Our System	Current Manual List
Branches of science	Open Data Science Conference
Data analysis	Scientific Data
Bayesian inference	Women in Data
Computer science	Python (programming language)
Deep learning	R (programming language)

Data Science Article

Our System	Current Manual List
Niels Bohr	Earth science
Albert Einstein	Neurophysics
Isaac Newton	Psychophysics
James Clerk Maxwell	Relationship between mathematics and
Theory of relativity	Science tourism

Physics Article