Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Fachbereich Informatik
Department of Computer Science

Bachelor Thesis

Computer Science

Utilizing the Power of Deep Learning to evaluate Wikipedia content

by

Mohammad Saknini
9040060

Supervisor	Philipp Baaden
Examiner 1	Prof. Dr. Jörn Hees
Examiner 2	Dr. Milos Jovanovic
Draft as of	2023-11-09
To be submitted on	2023-09-30

Declaration

I hereby assure in lieu of oath that I have independently written this thesis.

All passages taken verbatim or in meaning from published or unpublished works of others have been marked as such. All sources and aids that I have used for the work are indicated.

This thesis has neither been submitted with the same content nor in essential parts to any other examination authority.

Bonn, 2023-09-30

Mohammad Saknini

Abstract

In this work, an automated system for assessing the quality of Wikipedia content is developed using deep learning techniques. The main goal is to partially replicate the findings presented in the (Wang & Li, 2019) paper, while incorporating additional deep learning methods for assessing the quality of Wikipedia articles. To ensure a fair replication, the feature sets were mostly kept in their original form, but were also used to train a gradient boosting framework instead of exclusively neural networks. By employing different deep learning models, further optimizing them, and retaining similar features, the replication efforts achieved superior results to the original research. This research provided valuable insights into the potential for performance improvement. It highlights key features and techniques that, when refined, can improve the classification results of content quality assessment on Wikipedia.

Contents

List of Figures

List of Tables

Acronyms

CNN Convolutional Neural Network

DART Dropouts meets Multiple Addetive Regression Trees

DNN Deep Neural Network

FA Featured Article

GA Good Article

GBDT Gradient-boosted Decision Trees

GOSS Gradient-based One-Side Sampling

LGBM Light Gradient-Boosting Machine

LSTM Long Short-Term Memory

ORES Objective Revision Evaluation Service

RNN Recurrent Neural Network

SHAP SHapley Additive exPlanations

1. Introduction

Wikipedia has become a valuable source of information for millions of users across the world, with a total of 284 billion page views and a growth of 6.7% from the year 2022 to 2023 Wikipedia statistics[1]. However, the reliability and quality of its content have been a subject of debate throughout the years (Blikstad-Balas, 2016; Meishar-Tal, 2015). While efforts have been made to assess and improve Wikipedia's content, the sheer volume and dynamic nature of the platform pose significant challenges.

Recent approaches to assessing Wikipedia content often require a significant amount of manual work and review, which can be slow, subjective, and prone to error. Some of the most commonly examined features include citations (Kousha & Thelwall, 2017), link structure and editing behavior (Ruprechter et al., 2020). Deep learning, on the other hand, can automatically learn patterns from data without extensive manual effort, enabling more efficient and objective assessments.

Deep learning has emerged as a powerful tool for analyzing and understanding complex patterns and datasets, which has revolutionized various fields. This bachelor thesis will explore the deep learning approach and help evaluate a portion of the world's largest collaborative online encyclopedia, Wikipedia[2].

The aim of this Bachelor's thesis is to use the modern power of deep learning and recreate an automated system for assessing the quality of Wikipedia's content, in order to better understand it and provide the Wikipedia community with yet another tool to assess the quality of an article. The foundation of this study is the paper authored by Ping Wang and Xiadon Li, which was published in 2019 (Wang & Li, 2019). However, due to the absence of certain critical information necessary for a complete replication, this research will focus on a partial reproduction of the original paper.

The goal is to first reproduce the results of the original paper using a similar approach, and then further optimize it to achieve better results. To achieve this objective the thesis will be structured as follows.

first step will involve examining existing research on the reliability of Wikipedia and the impact of different features on the quality of its content. Then, a closer look will be taken at various studies and works that have employed deep learning approaches to assess the quality of Wikipedia articles. This will further enhance the understanding of the topic and provide a strong foundation for this research.

Furthermore, the assessment system of Wikipedia ("Content assessment", 10/07/2023) will be examined , which is the core of this study because it enables the usage of supervised deep learning approaches. It is also crucial to have at least a brief understanding of the general structure of Wikipedia content pages because they are the source of the features

[1] https://stats.wikimedia.org/#/all-projects
[2] https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia

that will be used in the deep learning models. The utilized data sources such as Mediawiki API[3] and Wikipedia Dumps[4] will also be discussed. The methodology section will outline the approach to feature engineering, with a brief introduction to the feature sets and their related algorithm. Additionally, the selection and extraction of relevant features from Wikipedia articles, as well as the design and training of deep learning models tailored to the evaluation objectives, will be described.

Finally, the results are presented, which include a review of the importance of the different features and the influence of each feature set in conjunction with the deep learning models. The performance of the optimized models will also be outlined, accompanied by a brief comparative analysis of the feature sets, which highlights the reasons behind the performance differences. Within the evaluation section, a brief overview of the study replication is provided, summarizing the findings and limitations of this research.

2. Literature Review

In this section, the current state of research on Wikipedia will be reviewed, with a focus on research concerning the reliability of content, the significance and impact of specific features on article quality, and the various deep learning approaches that have been employed for content evaluation on Wikipedia.

Wikipedia has been the subject of numerous studies since its release. It has consistently emerged as the most frequently visited source of information over the years (Ball, 2023; Dzendzik et al., 2021). Despite its popularity, some research suggests that Wikipedia is still not globally seen as a reliable source of information (Rector, 2008). Studies indicate that the amount of misinformation and errors resulting from its dynamic nature is not too far from those found even in highly specialized subjects peer-reviewed source (Jemielniak, 2019).

However, research in a variety of fields has shown that while Wikipedia may have occasional errors, its large user base and dynamic nature make consistent progress in correcting inaccuracies in order to maintain the overall quality of the information presented. This collaborative effort leads to continuous improvement and accuracy of the content on the platform, making it a reliable and valuable source of information (Brown, 2011; Fallis, 2008; Nielsen, 2007).

2.1. Studies on the Reliability of Wikipedia

The correlation between the quality and credibility of Wikipedia is widely recognized. In fact, credibility has always been an essential concern for Wikipedia users and has

[3] https://www.mediawiki.org/wiki/API:Main_page
[4] https://dumps.wikimedia.org/

been studied in detail making it a controversial topic within the Wikipedia community. Assessing the credibility of Wikipedia content could be extremely subjective, and a series of studies have shown that using cognitive measures to assess credibility is statistically less reliable (Mesgari et al., 2015, p. 7).

In 2006, the Research (Chesney, 2006) have done an empirical study regarding the reliability of Wikipedia by collecting data from 55 participants by completing a survey to evaluate the reliability of Wikipedia's content by comparing the opinions of experts and non-experts. Participants were divided into two groups, one group read Wikipedia articles related to their field of expertise, while the other group read random articles. Credibility was assessed based on judgments of authors, articles, and the website itself, as well as participants' distrust of online information. The results showed no significant difference in author or website credibility between the two groups. However, experts rated the articles as more reliable than non-experts, supporting the original hypothesis. The study also highlight, the need for caution when using Wikipedia, although its information is generally accurate information, as some articles may still contain errors despite supporting the hypothesis.

A less subjective approach was employed by (Del Garcia Valle et al., 2018) to compare features and evaluate the quality of medical information on Wikipedia. The researchers focused on 10 commonly encountered medical conditions and examined the content of related Wikipedia articles in comparison to peer-reviewed literature. The findings suggested that, overall, the medical information provided in Wikipedia articles was reasonably accurate. Nevertheless, the study also identified certain limitations, such as inadequate references and instances of outdated or inconsistent information.

In summary, the reliability of Wikipedia has been the focus of numerous studies. Assessing its credibility is usually a subjective task, and even though certain measures have shown reliability in certain contexts, caution should be taken when using Wikipedia as a source of information due to the possibility of errors. Expert opinion tends to consider Wikipedia articles reliable, but it is important to remain cautious. While Wikipedia generally provides reasonably accurate information, it has certain limitations, such as insufficient citations and occasionally outdated or inconsistent data. Therefore, it is advisable to verify critical information with alternative reputable sources to ensure reliability.

2.2. Studies on the Features of Wikipedia and their Importance

One of the most common features when it comes to written content is its length. In the case of Wikipedia, this would be the word count of each article. In 2008, a quantitative method was developed by Joshua (Blumenstock, 2008), which used only word count as a feature to measure the quality of content on Wikipedia. To evaluate his method, he used two sets of data: random articles and featured articles, whereby featured articles are content that has been thoroughly examined and peer-reviewed. In order to test his method, he used the data provided by Wikipedia dumps, which had a total of 11,067

English articles after preprocessing. Finally, the results have shown that by using the simple word count feature alone, he was able to outperform some other more complex methods, indicating that word count could be a great indicator of content quality. In the conclusion of the paper, it was mentioned that his assumption may not be as accurate as it seems.

Wikipedia, being an encyclopedia that is maintained voluntarily by its community, the quality of the contributions is one of the main factors that directly impact the quality of the content. In Suzuki's proposal (Suzuki, 2015), he introduced an h-index based on the assumption that "if an editor's texts are left untouched by other editors and approved, then the editor is considered a good editor". The core of the method used was to evaluate the editors based on their previous ratings and calculate the h-index and p-ratio. Then, the article quality is determined using the weighted sum of the calculated values. The paper conducted two experiments that were evaluated against Remain, a method proposed in the work of (Adler & de Alfaro, 2007). The results have shown that their proposed technique using the h-index has yielded better results than their baseline.

References tend to be the backbone of any scientific work as they support and provide evidence that the conducted research could have some truth to it by offering additional evidence to continue on. In addition to the usual citations in scientific works, Wikipedia has its own Wikilink structure that allows articles to reference other articles within Wikipedia itself. In (Ruprechter et al., 2020, p. 8), the Wikilinks structure was utilized to build a graph network, which provided additional features to help assess the quality of the content on Wikipedia, such as the in-degree and out-degree of each node.

Wikipedia pages typically begin with basic information and gradually expand over time, potentially becoming reliable sources of information. Wikipedia stores this progression through a system called revisions, which include a significant portion of the details tracking the changes made over time. These details consist of a timestamp, user information, and the size of the revisions. In certain studies, like the one conducted in (Zhang et al., 2018), this feature was utilized to assess the quality of Wikipedia content. By examining the revisions, the researcher was able to analyze the development of crucial elements such as the number of citations and sections. Finally, this information was fed into a Long Short-Term Memory (LSTM) model to learn how to evaluate the articles effectively.

(Q. V. Dang & Ignat, 2016) used a method that requires little to no manual feature engineering. The paper presented a novel approach for classifying Wikipedia articles based on their content rather than using a predefined feature set. The approach utilized natural language processing and deep learning techniques, specifically Doc2Vec and deep neural networks. The results of this experiment were later evaluated, achieving an accuracy of 55%. These results were compared with models implemented by Objective Revision Evaluation Service (ORES), an artificial service built by Wikimedia's Machine Learning Team.

2.3. Studies on Deep Learning Approaches to Assess the Quality of Wikipedia

In recent years, the popularity of deep learning approaches for problem-solving has gained noticeable attention, inspiring researchers to explore this path and push the boundaries of its application to achieve state-of-the-art results. In the case of Wikipedia, several studies have been conducted using a variety of deep learning models in combination with different features.

(Q. V. Dang & Ignat, 2016) used a Deep Neural Network (DNN) for a supervised classification task. To accomplish this, they utilized vectors derived from Wikipedia data and transformed them into vectors using the Doc2Vec algorithm. These vectors were subsequently fed into the DNN for processing. The network had a simple architecture comprising four hidden layers, each containing 2000, 1000, 500, and 200 neurons, respectively. This work resulted in 55% accuracy.

Later, the work was improved in (Q.-V. Dang & Ignat, 2017) by using a Recurrent Neural Network (RNN) with a Long Short-Term Memory (LSTM) cell to create a bidirectional LSTM model instead of a standard DNN. This improvement allowed them to enhance their previous results.

The main advantage of using an RNN is that they were able to utilize the raw data provided by Wikipedia, rather than having to transform it using Doc2Vec. This approach potentially enabled them to retain more important information from the page. It is worth noting that the computational time needed to train this model was around four days for training and six hours for testing. As a result of this work, they achieved 68% accuracy.

Later on, XGBoost was employed. XGBoost is a powerful machine learning algorithm for its excellent performance and flexibility in classifying data. It is highly scalable, allowing for classifications to be performed using fewer computing resources compared to deep learning approaches. The model was initially introduced by (Chen & Guestrin, 2016) and later on used by (Schmidt & Zangerle, 2019) to assess the quality of Wikipedia's content. The research also highlighted the importance of the features used, particularly emphasizing the significance of the document text that was fed into the model using Doc2Vec, in comparison to the other features. As a result, an accuracy rate of 73% was achieved by the model.

Last but not least, The research conducted by (Wang & Li, 2019) is the core of this thesis. In their study, they used a comprehensive framework comprising six unique feature sets. These feature sets were later used as input for several deep learning classification models, namely Convolutional Neural Network (CNN), DNN, and LSTM. Among these models, the LSTM architecture underwent extensive experimentation, with five different variations explored: Basic LSTM, LSTM with dropout, Stacked LSTM, Bidirectional LSTMs, and CNN-LSTM, which combines both models into a larger one. Compared to the LSTM model used in the research conducted by (Q.-V. Dang & Ignat, 2017), noticeable improvements in training time were achieved. The training times for the basic, stacked, and bidirectional LSTMs were reduced to only 10.4, 17.6, and 47.9 seconds, respectively. The combination of the features and the deep learning models has outperformed the previously mentioned works, achieving an impressive accuracy of 79.8% using the stacked LSTM model as the best result.

3. Wikipedia

Wikipedia is a massive online encyclopedia that serves as a collaborative platform for users to create and edit articles on a wide range of topics. It was launched in January 2001[1] and has since grown to become one of the most popular and comprehensive sources of information on the internet. In this section, the importance of the systems and the way articles are structured on Wikipedia will be explained.

3.1. Content Assessment System

In order to train a supervised machine learning model, labeled data is required. Fortunately, Wikipedia has an existing system called Content Assessment ("Content assessment", 10/07/2023) that assesses the quality of articles based on the editors' collective opinion. However, it is important to note that these evaluations are subjective and may sometimes be incorrect. Therefore, in the event of significant disputes, the WikiProject[2] team will be involved to reach a consensus on the most accurate rating. The article quality grading scheme as shown in Figure 1 comprises six distinct grades, each with its own set of criteria that must be satisfied to receive the corresponding grade.

Although there are distinct levels for evaluating articles, they fail to provide a substantial differentiating factor between the classes. As previously mentioned, the community primarily evaluates articles based on a predefined set of criteria that must be fulfilled to meet the requirements at each classification level.

Stub

A stub article is very short and lacks substantial information on the topic. It typically provides only basic details or a brief overview and requires significant expansion.

Start

Start-class articles are longer than stubs and provide additional information about the subject. However, they often lack depth and may require further development. To meet the minimum criteria, an article should possess at least one of the following attributes: an accompanying image, helpful links, or well-structured subsections.

[1] https://en.wikipedia.org/wiki/History_of_Wikipedia

[2] https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Council/Guide

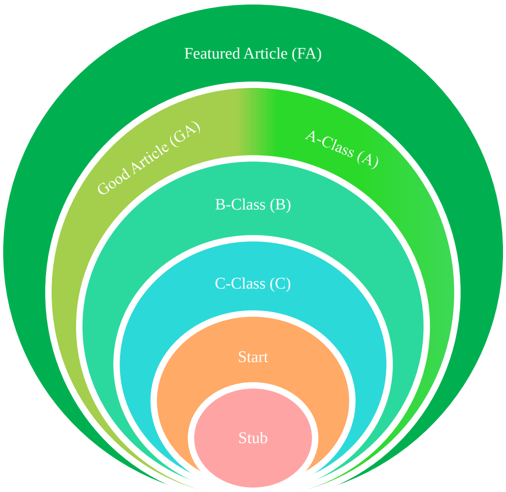


Figure 1: Article quality rating scheme based on ("Content assessment", 10/07/2023), ranging from the lowest-rated "Stub" to the highest-rated "Featured Article", where "Good Article" and "A-Class" are equally good.

C-Class (C)

C-Class articles provide a satisfactory level of coverage of the topic and present a substantial amount of content. However, there may still be occasional gaps or deficiencies in their organizational structure. In order to achieve a C-Class or higher classification, these articles must contain citations from multiple reliable sources and demonstrate improved style, structure, and overall quality compared to Start Class articles. However, despite these improvements, the article does not meet one or more of the criteria necessary to be classified as a B-Class article.

B-Class (B)

From this point onward, the criteria for evaluation become clearer and more detailed. To achieve a B-class grading, an article should demonstrate a good level of quality and completeness, without any major gaps or issues. It should meet the following requirements:

- Criteria in C-Class must be met.
- The article should be cited by reliable sources and include inline citations.
- It should be free from major inaccuracies.
- Has a consistent structure.
- The content must be presented in a manner that is easily understandable.
- The article should include suitable supporting materials.

Good Article (GA)

The assessment system undergoes changes and adopts a more stringent approach at this grading level. In order for an article to achieve the status of Good Article (GA), it must first be nominated by users who have made significant contributions to the project and possess a strong understanding of the subject matter, as well as cited sources. The nomination is then reviewed by users with in-depth knowledge of Wikipedia's content policies. This comprehensive process typically spans around 7 days and consists of two distinct stages: nomination and review.

During the nomination[3] process, an article is submitted after ensuring it meets all the necessary criteria. Subsequently, in the reviewing phase, the article must be adjusted based on the feedback received from the reviewers to advance in the assessment process. To be eligible for reviewing an article. The reviewing users must be registered and not be the nominators themselves, nor have made significant contributions to the article in question.

In addition to meeting the criteria for B-Class, an article must satisfy six additional criteria[4] to attain GA status, which is:

- Well-written and follows the styling guide outlined in the Manual of Style[5].
- References must be verifiable and should not include any original research.
- Stability is essential, meaning there should be no content disputes or ongoing edit wars.
- The article should maintain a neutral standpoint, presenting content without bias.
- Should provide comprehensive coverage of all the key aspects pertaining to the topic.
- Should remain focused on the topic at hand, avoiding unnecessary details.

Besides the criteria that must be met to achieve a GA rating, there are certain immediate failures that can stop the review process altogether. These failures include:

- Significantly deviating from meeting any of the GA criteria.
- Violation of copyright laws by including unauthorized content.
- Displaying clean-up banners, such as "citation needed," indicating a lack of proper referencing.
- The article is involved in an ongoing edit war.
- Failing to adequately address feedback from a previous reviewer, as noticed by a different reviewer.

[3] https://en.wikipedia.org/wiki/Wikipedia:Good_article_nominations/Instructions
[4] https://en.wikipedia.org/wiki/Wikipedia:Good_article_criteria
[5] https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style

A-Class (A)

In order for an article to receive an A rating, it must meet criteria similar to those required for a GA article. To ensure its excellence, the article must be extensively supported by verifiable sources, provide comprehensive coverage, maintain accuracy and neutrality, and be mainly focused on the topic. In addition, the article should have a consistent and coherent organization with appropriate sections and clear headings. The article must also comply with the established style guidelines to ensure consistency and clarity throughout and it is also critical to avoid potential copyright violations. While an A-Class[6] article is nearly on par with an FA in terms of quality, it may fall short due to minor styling issues.

It is important to note that while the criteria[7] for achieving Class A status are very similar to those for Class GA status, prior classification as a Class GA article is not a prerequisite for achieving Class A status ("Content assessment", 10/07/2023).

The nomination process for articles differs from the GA process. To nominate an article, a discussion must be initiated on the article's talk page. It should receive support from two uninvolved editors and no significant objections. Alternatively, a more formal review can take place, involving a project coordinator who assesses and approves the nomination.

A project coordinator[8] serves as a single point of contact for project-related issues. They are responsible for maintaining and managing the tasks necessary to ensure the running of the project and its internal processes. These tasks include making announcements, updating task lists, and overseeing the review process. Coordinators are elected through a simple approval vote by the members, typically held every twelve months.

Featured Article (FA)

A featured article, which represents the pinnacle of Wikipedia content, is the result of a careful evaluation process that combines measures of both GA and A-Class Articles. To achieve FA status, an article must meet all of the above criteria mentioned under (GA and A-Class) and undergo a thorough review by multiple editors, resulting in a consensus on its compliance with standards. Coordinators then evaluate the validity of the consensus and ultimately determine whether the article merits the prestigious FA status.

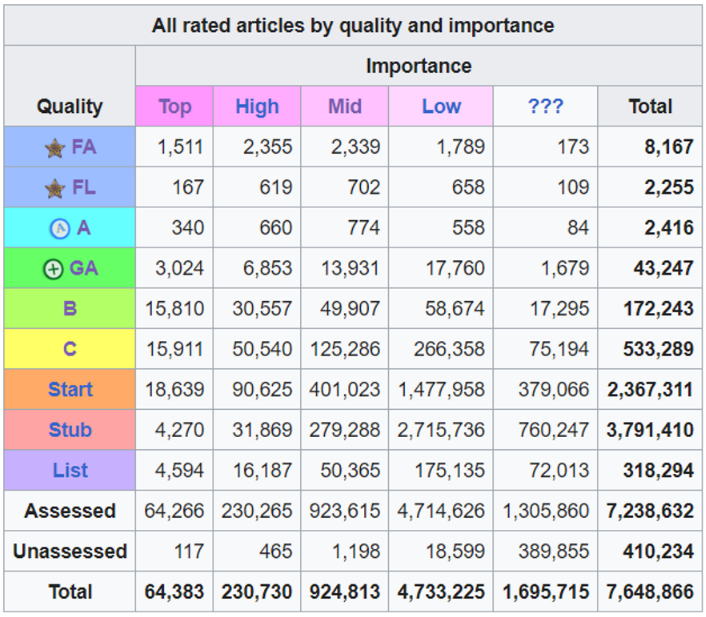
Although the content assessment system was initially introduced in 2005 ("Content assessment", 10/07/2023), only a relatively small fraction of the total Wikipedia articles have undergone assessment. Figure 2 provides a summary of the assessed articles. In this specific case study, the importance of the articles is deemed irrelevant.

In 2011, researchers (Yaari et al., 2011) conducted a user study to investigate how users who are not involved with Wikipedia perceive the quality of its articles. The primary goal of the study was to classify articles as either high or low quality. By analyzing

[6] https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Military_history/Assessment/A-Class

[7] https://en.wikipedia.org/wiki/Wikipedia:Content_assessment/A-Class_criteria

[8] https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Military_history/Coordinators



All rated articles by quality and importance						
	Importance					
Quality	Top	High	Mid	Low	???	Total
★ FA	1,511	2,355	2,339	1,789	173	**8,167**
★ FL	167	619	702	658	109	**2,255**
A	340	660	774	558	84	**2,416**
⊕ GA	3,024	6,853	13,931	17,760	1,679	**43,247**
B	15,810	30,557	49,907	58,674	17,295	**172,243**
C	15,911	50,540	125,286	266,358	75,194	**533,289**
Start	18,639	90,625	401,023	1,477,958	379,066	**2,367,311**
Stub	4,270	31,869	279,288	2,715,736	760,247	**3,791,410**
List	4,594	16,187	50,365	175,135	72,013	**318,294**
Assessed	64,266	230,265	923,615	4,714,626	1,305,860	**7,238,632**
Unassessed	117	465	1,198	18,599	389,855	**410,234**
Total	**64,383**	**230,730**	**924,813**	**4,733,225**	**1,695,715**	**7,648,866**

Figure 2: Summary table showing the number of graded articles per respective class and their corresponding importance, source: ("Content assessment", 10/07/2023).

participants' arguments for selecting the best and worst quality articles from a set of five articles they viewed, the study identified two main categories of criteria: measurable and non-measurable criteria for evaluating content. These categories are shown in Figure 3 and Figure 4.

Interestingly, during the experiment, users have noticed some new features that were not originally mentioned in the criteria list of the Wikipedia content assessment guide. These features include the number of unique editors, the count of anonymous editors, and even the names of registered users who have made contributions to the article.

In addition, the study (Yaari et al., 2011) examined not only the categorization of criteria, but also the distribution patterns of users who used specific criteria to rate articles, distinguishing between low and high quality. The goal of this in-depth study was to identify the factors that influence users' ratings. Figure 5 and Figure 19 illustrate these findings.

Overall, the assessment system falls short of being a flawless system that delivers perfect evaluations for articles. Its purpose is to provide a general estimation of an article's quality and its relative position compared to highly rated pages. According to the information provided by Wikipedia regarding the system's functioning ("Content assessment", 10/07/2023), it becomes evident that the initial three grades primarily depend on factors related to features such as text length and pictures. On the other hand, the higher grades incorporate these same criteria while also considering subjective opinions from the community. This inclusion of subjective opinion makes it difficult to assess the accuracy and reasoning behind each grade, justifying a statistical approach to compare the the importance features used.

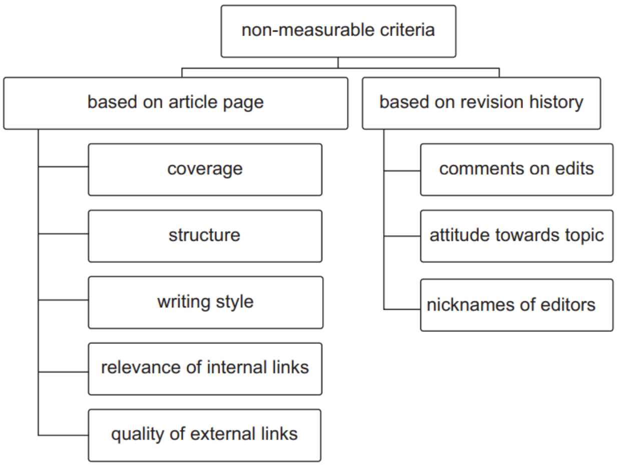


Figure 3: Categories of non-measurable criteria cited by participants, as documented in the study conducted by (Yaari et al., 2011, p. 492)

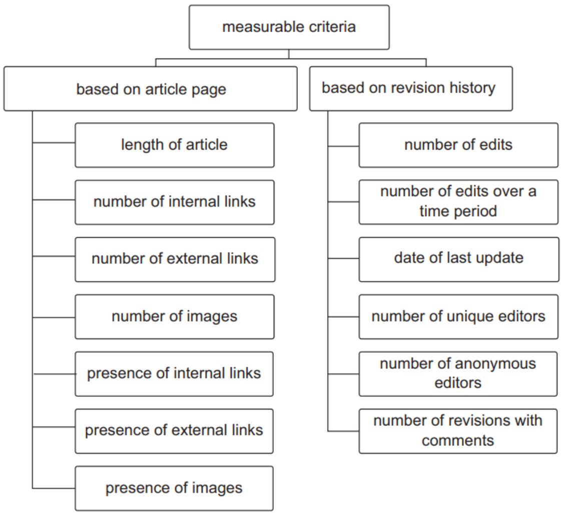
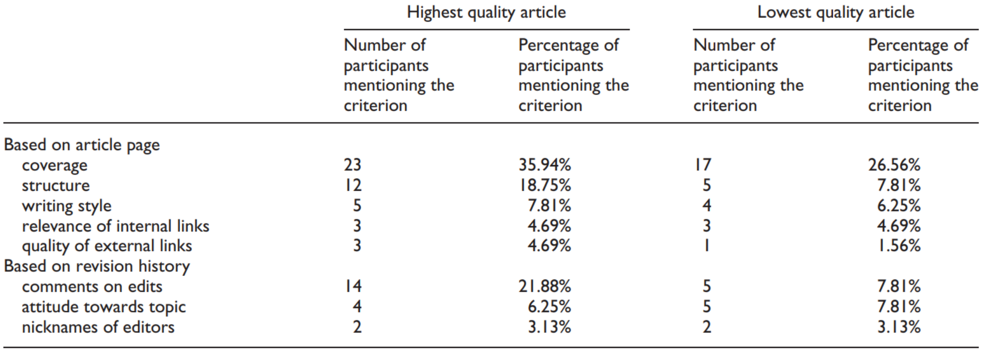


Figure 4: Categories of measurable criteria cited by participants, as documented in the study conducted by (Yaari et al., 2011, p. 492)

3.2. Content Structure

A Wikipedia article is not just raw text; it consists of a structured[9] set of information that follows a general format used by Wikipedia. The purpose of providing a general structure to an article is to enhance readability, and completeness, and ensure a consistent layout throughout Wikipedia. This common layout makes it easier for readers to navigate through the content. Generally, an article's structure should obey the following principles:

[9] https://en.wikipedia.org/wiki/Wikipedia:How_to_structure_the_content



	Highest quality article		Lowest quality article	
	Number of participants mentioning the criterion	Percentage of participants mentioning the criterion	Number of participants mentioning the criterion	Percentage of participants mentioning the criterion
Based on article page				
coverage	23	35.94%	17	26.56%
structure	12	18.75%	5	7.81%
writing style	5	7.81%	4	6.25%
relevance of internal links	3	4.69%	3	4.69%
quality of external links	3	4.69%	1	1.56%
Based on revision history				
comments on edits	14	21.88%	5	7.81%
attitude towards topic	4	6.25%	5	7.81%
nicknames of editors	2	3.13%	2	3.13%

Figure 5: Distribution of the usage of non-measurable criteria (Yaari et al., 2011, p. 493)

Start with the core topic, providing a concise introduction. Then, briefly develop all the main aspects of the topic, followed by an optional in-depth exploration of specific aspects. This approach provides 2 or 3 successive pictures of the topic, each focusing on it but with increasing content. This allows readers to stop reading when they have gathered sufficient information. When possible, follow a chronological order of events, as people tend to remember information better when it is connected in this way. For each individual event, explain "who did what when why," and include possible explanations of special challenges, techniques, resources, or consequences. If chronological order is not appropriate, use a logical order and ensure that all the steps in the logic are stated in the correct sequence. In articles with complex or varied topics, it may be helpful to include a paragraph or section after the introduction that explains the different ways the topic can be described, compared, or considered.

The opening section of an article, referred to as the lead section, serves as both an introduction and a concise summary of the article's key points. It occupies the beginning of the article. In some cases, articles on Wikipedia tend to expand rapidly, and in order to prevent internal issues, Wikipedia has imposed limitations on their length. This necessitates the practice of article splitting[10], which involves dividing the content into appropriately sized sections and creating separate pages for larger subsections. A general guideline suggests that articles exceeding 100 kilobytes in size should be divided. This approach helps maintain focus on the topic and enhances the overall readability and organization of the article.

A well-developed article often incorporates several essential elements[11], including:

- Infoboxes: These panels are typically positioned at the top right of the article and provide a structured layout to present key features or information about the subject.

- Sidebar: This element contains a cohesive collection of relevant links, pointing to multiple related articles.

[10] https://en.wikipedia.org/wiki/Wikipedia:Splitting

[11] https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Lead_section#Citations

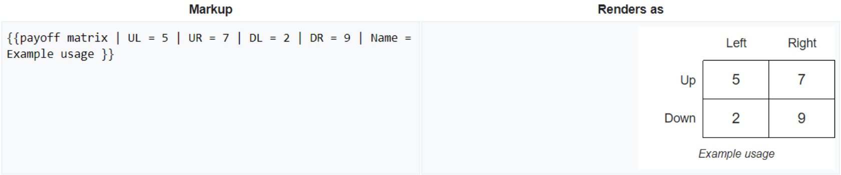
- Table of Contents (ToC): Automatically generated table of contents that appears on pages with a minimum of four headings.

One of the basic components of a Wikipedia page is called a template. A template[12] is essentially a page within Wikipedia that is designed to be included in other pages. Its purpose is usually to display repetitive content that needs to appear on multiple pages. Templates can take various forms, ranging from boilerplate messages such as warnings or notices to more structured elements such as infoboxes. Much like calling a function in a programming language, templates can take parameters, including text to be displayed along with the template itself.

To invoke a template, the following syntax is used “template name”, while to pass arguments to a template, use

```
{{template name|parameter1=value1|parameter2=value2|...}}
```

Figure 6 shows a simple example of a template used to generate a 2-by-2 matrix.



Markup

```
{{payoff matrix | UL = 5 | UR = 7 | DL = 2 | DR = 9 | Name =
Example usage }}
```

Renders as

	Left	Right
Up	5	7
Down	2	9

Example usage

Figure 6: Generating a 2x2 matrix using a Template, Source: (“Wikipedia Template”, 23/06/2023)

In the research conducted by (Yaari et al., 2011), they investigated the significance of templates in assessing page quality. The study highlighted the crucial role templates play in evaluating the quality of an article. Specifically, it demonstrated that when templates are absent from the structure of a page, users tend to perceive the article as having low quality. This finding underlines the importance of incorporating templates in enhancing the overall evaluation of page quality.

In addition, in the study of (Lamprecht et al., 2017), they investigate how individuals navigate the vast information network of Wikipedia. The researchers examined two types of navigation: free-form navigation, which involves analyzing all clicks within the English Wikipedia over the course of a month, and goal-directed navigation, which involves examining Wikigames in which users aim to find specific articles by following links. The results show that users click more often on links located near the top of an article. This pattern can be attributed to the structure of Wikipedia articles, which strategically places links to more general concepts at the top. This organization facilitates navigation by allowing users to quickly access articles that are more closely related. These findings highlight the importance of article structure and link position in Wikipedia navigation and suggest that improving the organization of information can significantly improve the user experience.

[12] https://en.wikipedia.org/wiki/Help:Template

4. Data Sources

To train the supervised learning models in this work, data must first be extracted from Wikipedia, processed, and then fed to the model. Allowing the model to learn the patterns in order to assess the quality of the provided article. The data from Wikipedia is available to the general public for access and use, as it is a free encyclopedia. In this section, an overview will be provided of the available methods for extracting the necessary data.

4.1. Mediawiki API

MediaWiki[1] is an open-source software platform for creating and managing wikis. It is known as the driving force behind Wikipedia, this software allows users to manipulate and interact with the rich data stored in the Wikipedia database through its Wikipedia API. With the MediaWiki Wikipedia API, developers gain access to plenty of features that allow them to retrieve information, perform searches, and even make edits to articles.

One of the core features of the MediaWiki Wikipedia API[2][3] is article content extraction, which allows for the extraction of specific Wikipedia articles in various formats such as HTML, Wikitext, plain text, or JSON. Everyone can access not only the textual content, but also the files and templates associated with an article, including sections, headings, images, and links. In addition, valuable page metadata can be retrieved, providing insight into important details such as the article's title, revision history, categories, and other information, all without the necessity of fetching the entire content. The API also enforces rate limits on the number of requests that can be made in a given period of time. These limits are determined by factors such as request type, user authentication, and the specific API client being used.

In addition, the MediaWiki Wikipedia API offers extensive extensibility, allowing developers to build custom applications that take advantage of its capabilities. It provides a comprehensive set of parameters and options that can be used to tailor API requests and retrieve precise data according to specific requirements. Which will be utilized in this thesis by using tools such as pywikibot[4] providing a handful of customizable classes and functions that allow direct access to the API using Python[5].

[1] https://www.mediawiki.org/wiki/Manual:What_is_MediaWiki

[2] https://www.mediawiki.org/w/api.php

[3] https://www.mediawiki.org/wiki/API:Main_page

[4] https://github.com/wikimedia/pywikibot

[5] https://www.python.org/

4.2. Wikipedia Dumps

Wikipedia dumps[6] refer to large files that contain the complete text and metadata of the articles on Wikipedia. These dumps are generated at regular intervals, presenting a snapshot of the complete Wikipedia database at a specific point in time. They are openly available for download to the general public.

Wikipedia dumps are generally available in XML format, which means that they are structured records encoded in a standardized markup language. This XML structure makes it easy to parse and extract data from these dumps. For example, within the article dump, the content of the article from the most recent revision can be found, along with metadata, templates, categories, and other associated details. The size of Wikipedia dumps can vary considerably, from a few gigabytes to several hundred gigabytes. This variability depends on factors such as the specific type of dump and the language edition of Wikipedia being examined.

There are several advantages to using Wikipedia dumps instead of the MediaWiki API:

- Offline Access: Wikipedia dumps provide a complete snapshot of the entire Wikipedia content at a specific point in time. This allows for offline access, meaning you can access and analyze Wikipedia data without relying on an internet connection or making API requests.

- Data Integrity: Wikipedia dumps are generated as static files, ensuring data consistency. This means that the retrieved data from the dumps will be reliable and consistent.

- Data Processing Efficiency: XML-formatted Wikipedia dumps can be processed efficiently using various tools and programming languages. For example, the mwxml[7] parser is specifically designed for parsing dumps, and mwparserfromhell[8] provides tools to parse the content and templates of an article. These tools make it easier to work with the data in the dumps.

- Speed: Fetching data from the MediaWiki API can be slow, especially when dealing with a large amount of information or complex queries. In contrast, using Wikipedia dumps allows for faster access to the data, as it eliminates the need for making API requests.

Overall, opting for Wikipedia dumps instead of the MediaWiki API offers the benefits of offline access, data integrity, efficient data processing, and improved speed in data retrieval.

[6] https://dumps.wikimedia.org/

[7] https://github.com/mediawiki-utilities/python-mwxml

[8] https://github.com/earwig/mwparserfromhell

5. Methodology

In this section, the methods used to partially replicate the study of (Wang & Li, 2019) will be explained in detail, starting with data extraction. The process of retrieving the necessary data and identifying appropriate sources for it will be covered, the technical limitations and data selection criteria will be outlined as well. Then analyze how the data was manipulated and transformed to create meaningful features that can be effectively used by the deep learning models. Finally, the implementation of the deep learning models is examined, including details such as fine-tuning, architecture, and a basic description of each model used to assess the quality of an article from Wikipedia.

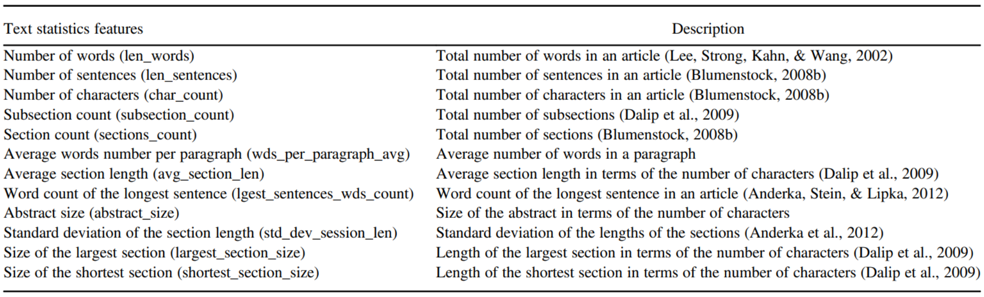
5.1. Feature Engineering

The process of transforming raw data into meaningful features that can be used as input for training a deep learning model is known as feature engineering. In this process, the appropriate data is selected and key statistics and findings are extracted using domain-specific knowledge and statistical approaches. Training a model relies heavily on feature engineering as it determines the patterns that will be made available to the deep learning algorithm for learning and improvement.

In this thesis, the models will be trained using three of the feature sets mentioned in (Wang & Li, 2019) will be used to train the models, namely:

Text Statistics Features

The majority of information on Wikipedia is in the form of text. This prompts further study of these features and their incorporation into deep learning models to uncover patterns within the text. By understanding these patterns, the quality of articles can be improved. The study by (Blumenstock, 2008) highlighted the importance of basic text-related features, such as page length. Even simple text statistics can have a significant impact on the analysis of content quality. The features used in the Text Statistics Features set are shown in Figure 7.



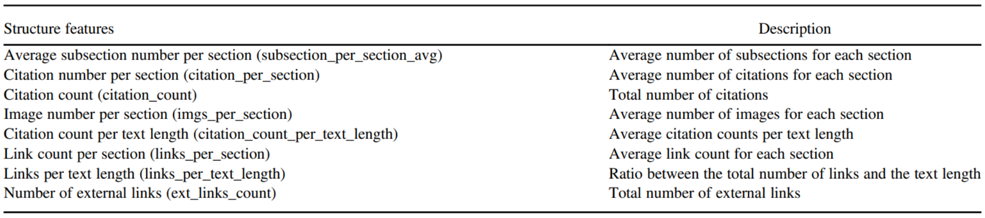
Text statistics features	Description
Number of words (len_words)	Total number of words in an article (Lee, Strong, Kahn, & Wang, 2002)
Number of sentences (len_sentences)	Total number of sentences in an article (Blumenstock, 2008b)
Number of characters (char_count)	Total number of characters in an article (Blumenstock, 2008b)
Subsection count (subsection_count)	Total number of subsections (Dalip et al., 2009)
Section count (sections_count)	Total number of sections (Blumenstock, 2008b)
Average words number per paragraph (wds_per_paragraph_avg)	Average number of words in a paragraph
Average section length (avg_section_len)	Average section length in terms of the number of characters (Dalip et al., 2009)
Word count of the longest sentence (lgest_sentences_wds_count)	Word count of the longest sentence in an article (Anderka, Stein, & Lipka, 2012)
Abstract size (abstract_size)	Size of the abstract in terms of the number of characters
Standard deviation of the section length (std_dev_session_len)	Standard deviation of the lengths of the sections (Anderka et al., 2012)
Size of the largest section (largest_section_size)	Length of the largest section in terms of the number of characters (Dalip et al., 2009)
Size of the shortest section (shortest_section_size)	Length of the shortest section in terms of the number of characters (Dalip et al., 2009)

Figure 7: Text Statistics Features, Source: (Wang & Li, 2019, p. 4)

The text statistics are collected by utilizing mwparserfromhell, spacy[1], and basic statistics. First, the data of the corresponding article will be extracted from the dump. Then, the raw text will be parsed and certain features such as sections and subsections will be extracted using mwparserfromhell. Meanwhile, spacy will be used to extract sentences from the text, making use of its built-in pre trained pipeline for English, "en_core_web_sm"[2]. Finally, all the data features will be stored as a comma-separated file, ready to be further utilized.

Structure Features

In addition to textual content, Wikipedia articles contain a number of other features, as discussed in Section 3.2. This implies that the structure and non-textual elements of an article are relevant to users and have the potential to improve the overall quality of the article. To address these aspects, the paper (Wang & Li, 2019) uses a set of previously proposed features that specifically target certain basic structural attributes, as shown in Figure 8. By considering these features, there is an opportunity to further optimize performance, as these elements have been proven to enhance the overall user experience and quality of Wikipedia articles.



Structure features	Description
Average subsection number per section (subsection_per_section_avg)	Average number of subsections for each section
Citation number per section (citation_per_section)	Average number of citations for each section
Citation count (citation_count)	Total number of citations
Image number per section (imgs_per_section)	Average number of images for each section
Citation count per text length (citation_count_per_text_length)	Average citation counts per text length
Link count per section (links_per_section)	Average link count for each section
Links per text length (links_per_text_length)	Ratio between the total number of links and the text length
Number of external links (ext_links_count)	Total number of external links

Figure 8: Structure Features, Source: (Wang & Li, 2019, p. 4)

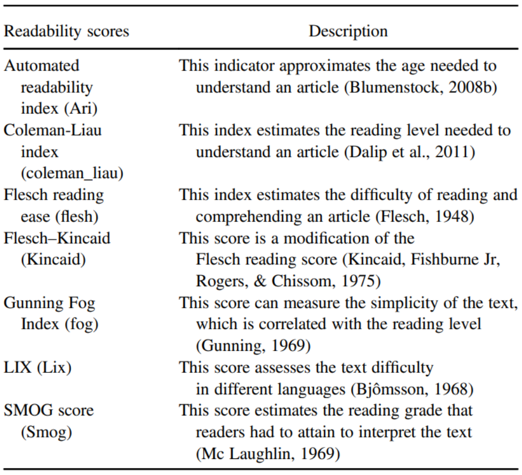
The collection of Structure Features will be done similarly to Text statistics Feature. The total number of citations will be gathered by checking every instance of <ref> and <cite> tags, and verifying if they are not empty. If they are empty, they will be considered as invalid references. Additionally, the extraction of images will involve iterating through all the files and checking if they are of type png, jpg, gif, or jpeg.

The impact of Wikipedia's structure on user behavior and page navigation has been highlighted in a study conducted by (Lamprecht et al., 2017). The study examined how the arrangement of content and hyperlinks within Wikipedia pages influences user actions. Interestingly, the results showed that users tend to click on hyperlinks positioned at the top of an article. This suggests that studying elements such as infoboxes and sidebars, which provide additional information and are positioned early in the article, could potentially yield better results in understanding user behavior. Therefore, further investigation into the influence of these template elements within Wikipedia could result in new valuable features.

[1] https://spacy.io/
[2] https://spacy.io/models/en

Readability scores

The features presented in Figure 9 are valuable for evaluating the level of readability associated with the text within the article. These scores take into account factors like education and grade level, giving us insights into the knowledge and skills required to understand the content. By examining a handful of scores, the model should be able to evaluate the text's difficulty and determine if it affects the article's quality.



Readability scores	Description
Automated readability index (Ari)	This indicator approximates the age needed to understand an article (Blumenstock, 2008b)
Coleman-Liau index (coleman_liau)	This index estimates the reading level needed to understand an article (Dalip et al., 2011)
Flesch reading ease (flesh)	This index estimates the difficulty of reading and comprehending an article (Flesch, 1948)
Flesch–Kincaid (Kincaid)	This score is a modification of the Flesch reading score (Kincaid, Fishburne Jr, Rogers, & Chissom, 1975)
Gunning Fog Index (fog)	This score can measure the simplicity of the text, which is correlated with the reading level (Gunning, 1969)
LIX (Lix)	This score assesses the text difficulty in different languages (Bjômsson, 1968)
SMOG score (Smog)	This score estimates the reading grade that readers had to attain to interpret the text (Mc Laughlin, 1969)

Figure 9: Readability scores feature set, Source: (Wang & Li, 2019, p. 5)

Various features are relied upon by the readability scores used in the feature set. For instance, the Automated Readability Index (ARI) formula is utilized, which is calculated as follows: $(4.71 \cdot \frac{\text{characters}}{\text{words}} + 0.5 \cdot \frac{\text{words}}{\text{sentences}} - 21.43)$ (Smith & Senter, 1967, p. 14). The results of this formula fall between 0 and 100, with a higher score indicating greater ease of readability.

In contrast, the SMOG score (Laughlin, 1969) employs a different approach. Unlike ARI, which utilizes basic text statistics, SMOG assesses the actual words and determines their difficulty by counting syllables. The formula is calculated as follows: $(1.043 \cdot \sqrt{(\text{polysyllabic words} \cdot \frac{30}{\text{word count}})} + 3.1291)$. The higher the output, the greater the estimated number of years of education required to comprehend the text.

The readability scores are calculated using the readcalc[3] library, which offer easy to use functions that automatically calculate all the required readability scores and even more.

It is worth noting that when later on tested separately in (Wang & Li, 2019, p. 10), the structured feature set achieved the highest performance, with a precision of 85.05%. In comparison, text statistics and readability scores achieved 80.80% and 62.37% respectively.

[3] https://pypi.org/project/ReadabilityCalculator/

5.2. Data Extraction

The data extraction step is critical because it lays the foundation for subsequent analysis and modeling. During this process, relevant data records that meet certain criteria are identified and collected. As mentioned in Chapter 4 the main sources for retrieving data from Wikipedia are the real-time data from the MediaWiki API and the Wikipedia dumps, which provide a snapshot of the data to be processed locally.

To ensure the extraction of both the content of the articles and the grading of the content grading, both the API and the dump are used as sources. This approach was used because the template for the content rating is only found in the dumps for the GA and FA ratings. Meanwhile, the rating of almost any given article can be obtained through the MediaWiki API, regardless of whether it is FA, GA, or any other grading.

The dump used for this experiment is en-dump-20230620[4] which contains only English articles. The mwxml library is used to process the dump, and the Pageassements[5] query from the MediaWiki API is used to retrieve the content grading. To ensure the accuracy of the information obtained, the revision ID was used when querying the API instead of the title. This is necessary because the dump was generated as a snapshot, and articles tend to change over time. Therefore, the revision ID is critical to ensure that the content rating accurately matches the article content stored in the dump.

Experiment

A preliminary test experiment is performed to determine the estimated duration of the process and to examine the distribution of the content when iterating the dump alphabetically using mwxml. The setup involved going through the first 200,000 pages of the dump and retrieving the assessment for each article using MediaWiki's API. It is important to note that the experiment was run on a single thread and without any parallelism. In addition, the requests were made in chunks of 50 pages per request, following the guidelines and rate limits provided by MediaWiki under Etiquette & usage guidelines[6], which also helped speed up the process.

The experiment's findings show that out of the total articles, 51,751 (25.88%) have been assessed, while 148,249 (74.12%) articles remain unassessed, as shown in Figure 10.

The distribution of ratings was also examined to get a better idea of the approximate amount of data that needs to be iterated over to ensure a sufficiently large and evenly distributed dataset of assessed articles. In addition, the dataset was filtered to exclude irrelevant gradings such as disambiguation and lists, as they are not relevant to this study. A total of 21,638 relevantly assessed articles were kept in the dataset.

The distribution of the assessed articles in the experiment, as shown in Figure 11, is significantly uneven, with A-Class, FA, GA, Stub, B-Class, C-Class, and Start assessment grading being 105 (0.08%), 1,643 (1.23%), 4,172 (3.12%), 16,085 (12.03%), 18,203 (13.62%), and 93,474 (69.92%) respectively.

[4] https://dumps.wikimedia.org/enwiki/20230620/
[5] https://en.wikipedia.org/w/api.php?action=help&modules=query%2Bpageassessments
[6] https://www.mediawiki.org/wiki/API:Etiquette

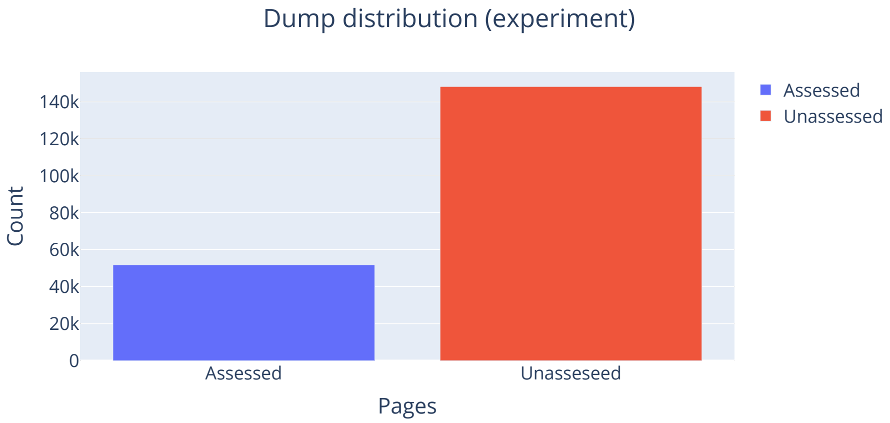


Figure 10: Assessment distribution within the first 200,000 pages of the dump. Approximately 25% of the total articles are assessed, highlighting the need to retrieve more articles.

The need to fetch a large number of pages to obtain a evenly distributed dataset containing at least 3,000 articles is indicated by the results. Finally, the fetching of the data took roughly 16 minutes.

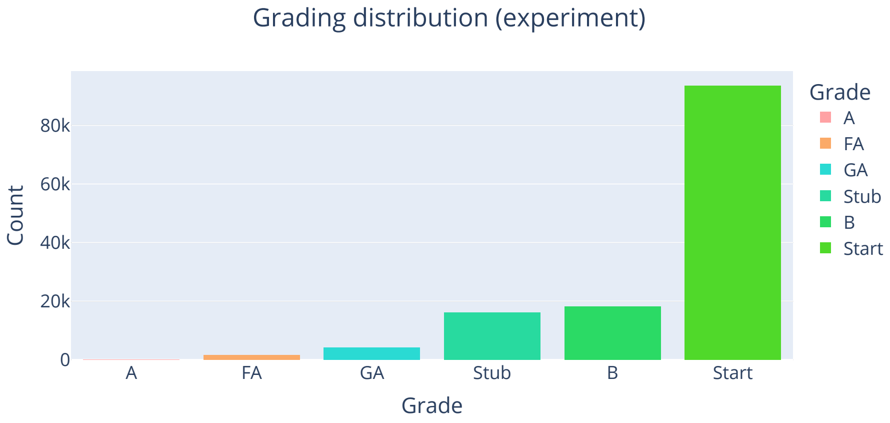


Figure 11: Grading distribution of the assessed pages in the first 200,000 pages of the dump. The distribution of grades demonstrates a clear skew towards lower-rated articles.

Dataset

After analyzing the findings of the Experiment, it was determined that 9 million elements in the dump would be extracted. A larger dataset than needed would be produced by this extraction, but it would also be uploaded to the public for potential further research if necessary. The goal was to extract the assessment grading for each of these articles, which would provide a satisfactory number of articles representing all grading levels. As a result, the distribution of evaluated elements, as shown in Figure 12, revealed that there were 2,704,895 (30.05%) articles classified as assessed, while the remaining 6,295,105 (69.95%) articles were classified as unassessed.

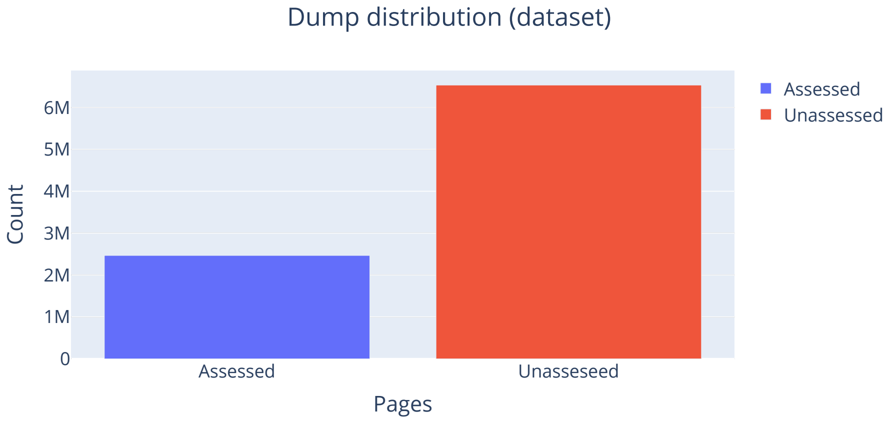


Figure 12: Assessment distribution of the dataset. Roughly 2.7 million of the 9 million total articles are assessed, providing a sufficient number of articles to create a dataset with 500 articles from each grade.

Similar to the experiment, the analysis also includes the distribution of the rated articles, as shown in Figure 13. It is clear that the distribution remains highly skewed in favor of lower-graded articles. The distribution includes 1,348,458 articles (54.68%) graded as Stub, 1,016,835 articles (41.23%) graded as Start, 71,311 articles (2.89%) graded as B-Class, 23,858 articles (0.97%) graded as GA, 4,999 articles (0.20%) graded as FA, and 693 (0.03%) as A-Class articles. Notably, there has been a significant increase in the distribution of stub articles, while the number of class A articles has decreased. Finally, the fetching of the data took roughly 9 hours and a half.

To maintain an accurate comparison with the original paper (Wang & Li, 2019), a similar dataset size, which is 3,000, and distribution will be used. Specifically, we will use only six levels of content evaluation grades provided by Wikipedia, which are grouped into two categories: High-quality articles (including the FA, GA, and A ratings) and low-quality articles (including B, Start, and Stub). Notably, the C class is excluded from the dataset used for training and evaluating the models. In addition, the k-fold cross-validation approach was implemented, with k set to 5 as stated in the original paper.

The final dataset used contains 3,150 articles, with exactly 525 articles per grade. The additional 150 articles, which accounts for an additional 5% compared to the articles used in the original experiment (Wang & Li, 2019, p. 7), were allocated for validation purposes. These validation articles were never exposed to the deep learning models during either the training or testing phases. This approach ensures a fair evaluation of the model's accuracy.

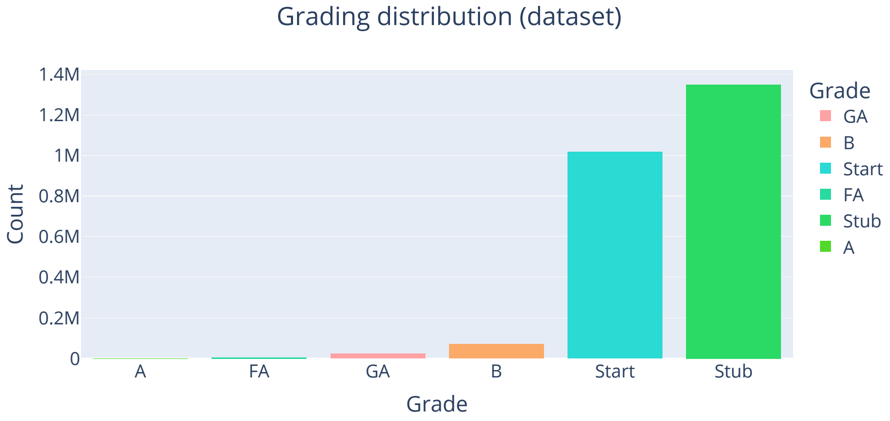


Figure 13: Grading distribution of the assessed pages in the dataset. Although the dataset remains skewed towards lower-graded articles, it still offers a sufficient quantity of articles to construct a dataset comprising 500 articles from each grade.

5.3. Deep Learning Models

In this section, the dataset previously proposed in Section 30 will be used to train deep learning models for classification task, specifically classifying the assessment of a Wikipedia article based on the proposed feature sets. Two machine learning models will be utilized: a gradient boosting model named LightGBM, deviating from the approach outlined in the original paper (Wang & Li, 2019) where only Neural Networks were used, and a simple deep neural network as proposed in the original study. The decision to avoid using LSTMs or CNNs is based on the absence of sequence data in the feature sets, which raises questions about the original paper's decision to initially employ these models.

5.3.1. Light Gradient-Boosting Machine (LGBM)

LightGBM, which was introduced by (Ke et al., 2017), is a gradient-boosting framework known for its speed, efficiency, and memory-friendly design. It shares similarities with XGBoost (Chen & Guestrin, 2016), but incorporates several key differences. One notable technique introduced in LightGBM is called Gradient-based One-Side Sampling (GOSS). This technique helps identify the most informative data points during the training process. Additionally, LightGBM uses an additional feature called Gradient-based Leaf-wise Tree Growth. Unlike the level-wise approach used in XGBoost, this technique grows trees leaf by leaf, resulting in more efficient tree construction.

To implement an LGBM model, the LightGBM library[7], an open-source software provided by Microsoft, is utilized. For hyperparameters tuning, the Optuna library[8] is used.

[7] https://github.com/microsoft/LightGBM
[8] https://github.com/optuna/optuna

Optuna enables the automatic optimization of machine learning models by utilizing efficient algorithms for hyperparameters sampling and effective pruning of unpromising trials. This approach reduces unnecessary computational time. Moreover, Optuna supports running the optimization process on multiple threads, further reducing the optimization time.

In order to train the LGBM provided by the LightGBM library, a custom class was created with various options to facilitate the replication of the model if necessary. The class incorporates several main parameters. First, it includes the input dataset used for training. Secondly, it allows the selection of the feature set to be used for training, such as the Text statistics Feature, the Structure Features, or a combination of both.

To ensure accurate experimentation, the class also provides the "random_state" parameter, which allows for the setting of the seed for number generators within Optuna and LightGBM. Additionally, the number of folds for cross-validation is also a parameter that can be adjusted, with the default value set to 5.

To optimize the model, a total of four Optuna studies were undertaken, with two studies dedicated to each boosting type. Each study involved 100 trials, and the optimization process followed Optuna's recommendations by combining the TPESampler and HyperbandPruner. Within the studies, two boosting types, namely Gradient-boosted Decision Trees (GBDT) and Dropouts meet Dropouts meets Multiple Addetive Regression Trees (DART), were employed. The results were impressive, with the DART model achieving accuracies of 84.41%, 81.16%, and 64.28% for Text Statistics Features, Structure Features, and Readability Features, respectively. Similarly, the GBDT model achieved accuracies of 86.36%, 82.46%, and 63.63% for the same features, as shown in Table 1. The hyperparameters utilized by the best models for each feature set are presented in Table 11, Table 12, and Table 13.

	Text Statistics Features	*Structure Features*	*Readability Features*
GBDT	**86.36%**	**82.46%**	63.63%
DART	84.41%	81.16%	**64.28%**

Table 1: LightGBM Classification accuracy

In terms of training time, approximately one hour was required for each feature set in both studies combined on the DART model. On the other hand, the GBDT model took an average of 1 minute and 42 seconds, as shown in details in Table 2. The noticeable

	Text Statistics Features	*Structure Features*	*Readability Features*
GBDT	**00:01:48**	**00:01:42**	**00:01:35**
DART	01:00:54	01:03:51	01:02:21

Table 2: Computational time needed to train and optimize the LightGBM models. In general, both boosting types perform well, with GBDT excelling when used along side Text Statistics and Structure Features, while DART outperforms in conjunction with Readability Scores.

difference in computation time can be partially attributed to not using the early stopping mechanism in combination with the DART boosting type. The utilization of early stopping, which terminates the optimization process if the accuracy fails to improve for a specified number of iterations (10 iterations in this case), is prevented due to this instability. Despite this limitation, the DART model consistently performed well throughout the trials, achieving an average accuracy of 83.09% across both studies. In comparison, the GBDT model averaged 78.34% across both studies, as shown in Figure 14. It is also worth noting that a significant number of identical trials are generated when TPESampler is combined with GBDT. This phenomenon can be observed throughout the results of the study. The issue of generating redundant trials has been a longstanding problem that has been extensively discussed[9] within the Optuna community.

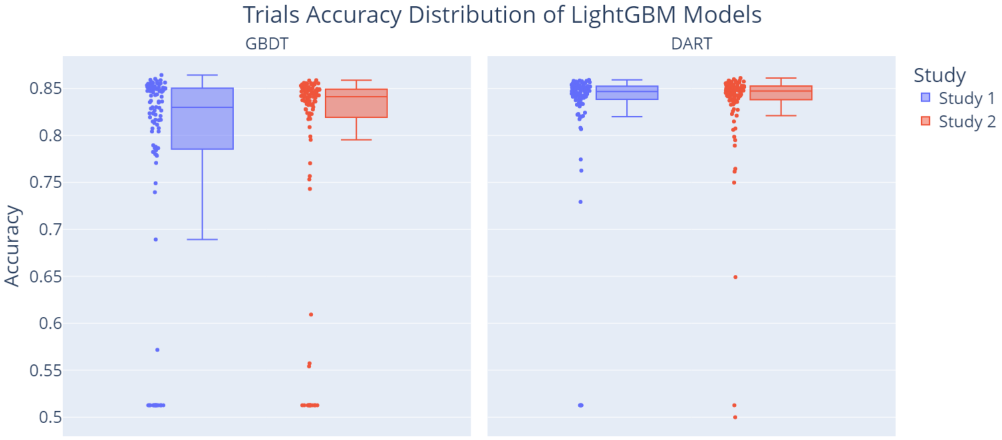


Figure 14: Comparison of trials accuracy distributions between GBDT and DART in Optuna studies using the Text Statistics Feature Set.

In terms of performance, overall GBDT demonstrated better results than DART. Specifically, when using Text Statistics Features, GBDT achieved a significantly higher accuracy of 1.95% compared to DART. Similarly, when using Structure Features, GBDT outperformed DART by 1.3% in accuracy. However, when using Readability Features, DART showed a slight advantage of 0.65% over GBDT.

5.3.2. Deep Neural Network (DNN)

A basic deep neural network (DNN) is an interconnected network of artificial neurons. It consists of an input layer, one or more hidden layers, and an output layer. The hidden layers learn complex patterns and relationships from the input data. During training, the network adjusts the connection weights to minimize the difference between predicted and desired outputs. DNNs have achieved significant success in various domains, such as image recognition and natural language processing, due to their ability to automatically

[9] https://github.com/optuna/optuna/issues/2021

extract features from data. Their depth enables them to capture increasingly abstract representations of the input.

Similarly to the original paper (Wang & Li, 2019), a basic deep neural network will be employed and tested. However, in this case, all the parameters utilized for model creation are generated during the optimization process. This is done due to the absence of information concerning the structure of the neural network utilized in the original research. The creation of the DNN will be done using the suggested libraries, Keras[10] and TensorFlow[11]. Additionally, for the optimization process, Keras_tuners[12] will be employed instead of Optuna because it performs better when used with its parent library Keras.

To implement the neural network, a custom class was created, similar to the one described in section 5.3.1, "random_state" parameter is also added to ensure consistency. This class provides several methods to efficiently build new models with different feature sets, using cross-validation with 5 folds. The optimization process was slightly modified. The number of trials per model was increased to 200, since KerasTuners does not support studies, only trials, thus doubling the number of trials.

Text Statistics Features	*Structure Features*	*Readability Features*
84.41%	82.46%	66.88%

Table 3: Deep Neural Network Classification accuracy

The DNN achieved an accuracy of 84.41% using Text Statistics Features, 82.46% using Structure Features, and 66.88% using Readability Features, as shown in table Table 3. The computation time for the respective feature sets was 00:27:12, 00:29:51 and 00:23:28, as shown in the table Table 4. The training and optimization of the models were done using the CPU, unlike LightGBM, which used the GPU. This decision was made due to the relatively small dataset, as transferring data from the CPU to the GPU takes a significant amount of computational time when using a deep neural network. Another difference in the training process is that the optimization of the DNN was run on a single thread, as TensorFlow does not support early stopping with multiple threads, resulting in a noticeable slowdown of the process.

The deep neural network's architecture varied depending on the feature set used, resulting in three distinct models with different complexities. When the text statistics feature set was employed, the optimized model had 1,401 parameters and included only one hidden layer, as shown in Figure 20. On the other hand, using the readability feature set led to a model with 2,711 parameters and two hidden layers, as shown in Figure 21. Finally, the largest model of the three was built using Structure Features, comprising 8,576 parameters and featuring three hidden layers, as shown in Figure 22.

[10] https://github.com/keras-team/keras

[11] https://github.com/tensorflow/tensorflow

[12] https://keras.io/keras_tuner/

Text Statistics Features	*Structure Features*	*Readability Features*
00:27:12	00:29:51	00:23:28

Table 4: Computational time needed to train and optimize the Deep Neural Networks. Overall, GBDT shows significantly less computational times compared to DART, saving an average of approximately one hour.

6. Evaluation

In this section, a comparison will be made between the outcomes of this replication and those of the original paper (Wang & Li, 2019), highlighting various performance differences and mentioning the possible reasons behind such discrepancies. Additionally, the importance of certain features within a feature set will be examined using feature importance detection libraries such as SHAP[1], along with other statistical approaches. Lastly, the introduced feature sets will be further examined as a whole, potentially showing gaps that could be improved in order to achieve better results.

6.1. Features Importance

Feature importance refers to the process of determining the significance of each input variable in a predictive model's performance. It helps identify which features have the most influence on the model's predictions and allows it to prioritize those that contribute most to the model's accuracy. Feature importance can be computed using various techniques such as permutation importance (Altmann et al., 2010), Gini impurity, or Entropy, as well as the information gained in decision trees. These techniques can be applied to different machine learning algorithms (Rajbahadur et al., 2022). Understanding feature importance aids in feature selection, and model optimization, and provides valuable insights into the underlying relationships between the features and the target variable, thereby enhancing the interpretability and trustworthiness of the predictive model.

To calculate the importance of the features within each feature set in each respective model, SHapley Additive exPlanations (SHAP) (Lundberg & Lee, 2017) will be used. SHAP is a popular method for explaining machine learning model predictions. It is based on cooperative game theory and the concept of Shapley values, which attribute contribution scores to individual features for a given prediction. SHAP works by computing the marginal contributions of features in all possible feature subsets and then averaging them to obtain the Shapley values. These values represent the influence of each feature compared to a baseline prediction, allowing users to understand how each feature contributes to a specific model prediction. SHAP's interpretability and fairness make it a widely used approach for explaining predictions and building trust in machine learning models. Visualizations like summary plots and waterfall plots help users grasp the importance of each feature in the model's decision-making process.

[1] https://github.com/shap/shap

To extract feature importance, the analysis was limited to the evaluation dataset (the additional 150 articles). This approach aims to examine the decision of the models using unseen data, which will be processed through a SHAP explainer to facilitate visualization. Three different plots will be focused on:

- Global bar plot, ranking the features according to their average absolute SHAP values across all the provided samples.
- Local bar plot, which is similar to the bar plot but is performed over only one input sample instead of the entire dataset, offering a closer look into why a certain prediction was made.
- Beeswarm plot, which combines the advantages of both local and global bar plots, providing a better overview of how the feature importance changes based on their current value.

The different global feature importance between the DNN and the LGBM was observed as a result of using the text statistics feature set. The top three features considered most important for prediction by DNN were the Number of Words, the Number of Characters, and the Size of The Largest Section. In contrast, for LightGBM, the most significant features were identified as an average Number of Words per Paragraph, Abstract Size, and Number of Words, as shown in Figure 15. This indicates that the two approaches have minimal overlap in terms of the global importance of the features and the reasoning behind their predictions.

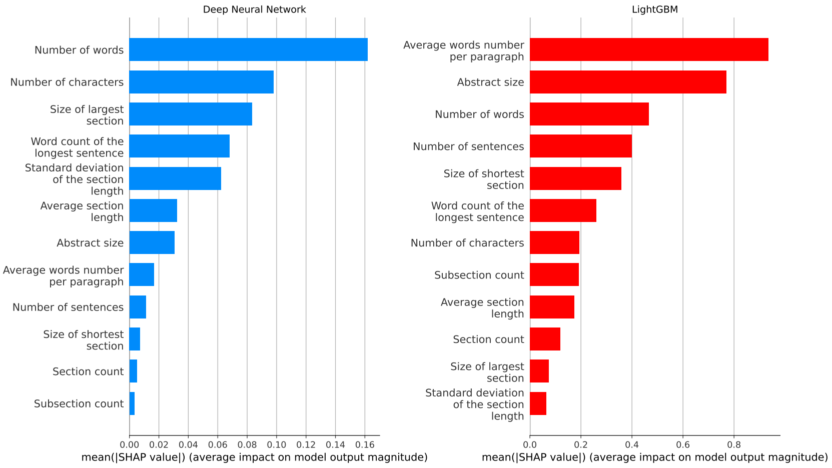


Figure 15: Comparison of global feature importance between the Deep Neural Network and LightGBM models using text statistics features. This figure emphasizes the difference in feature importance between the two models. While both models assign significant importance to the number of words, they diverge in their opinions of other features.

An interesting observation is made when examining the local feature importance. The data in Table 14 corresponds to a false-positive prediction, which occurs when an item that was initially rated as low quality is predicted to be high quality. This data sample is used as input for the local bar plot, which provides key insights into the factors contributing to preventing the false prediction. Notably, the randomly selected article stands out as an outlier, highlighting the limitations of the text statistics feature set and disproving the idea that all massive articles are of high quality.

Analyzing the importance of the features shown in Figure 16, it can be seen that for both DNN and LGBM, the word count of the longest sentence plays the most dominant feature, which in this outlier case has a significant positive impact on the evaluation of the article as low quality. This observation emphasizes the importance of refining this particular feature to effectively mitigate such outliers in the future.

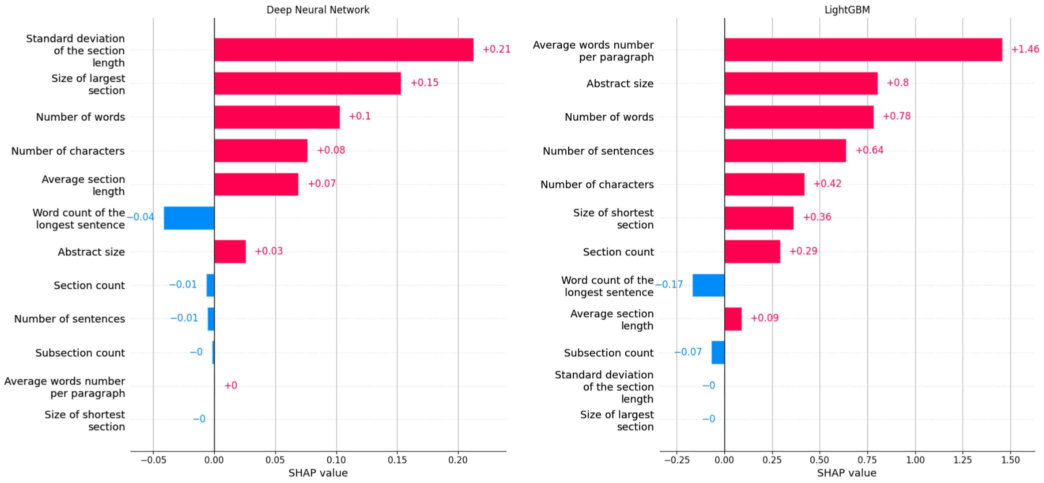


Figure 16: Local features importance using a false-positive data sample. Notably, the importance of each feature for a given prediction varies from the global feature importance. This variation is attributed to the influence of feature values, as demonstrated by the figure that most features suggest the selected article should be of high quality, except for the word count of the longest sentence, which indicates low quality for both models. This insight holds the potential to refine the models and reduce the likelihood of false positive predictions.

An overview of the features importance of the DNN is provided in Figure 17, showcasing how impactful the value of each respective feature could be regarding the model's prediction. For instance, it is evident that the subsection count rarely holds any meaningful importance, while the number of words almost always has a significant influence on the prediction, regardless of its current value. Interestingly, it can also be observed that higher numbers of characters, in contrast to the number of words, imply that the article is more likely to be of low quality. This trend is similarly applicable to the number of sentences. Overall, the DNN heavily relies on the top 8 features, with other features being considered impactful only in the case of an outlier.

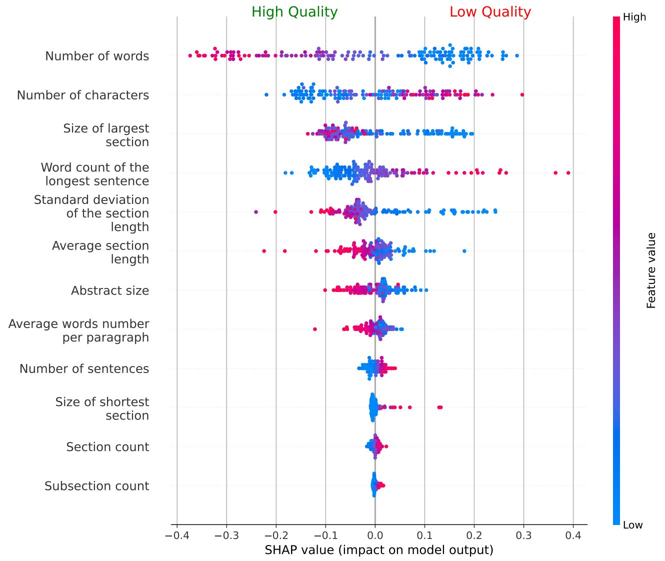


Figure 17: Overview of the features importance using the Deep Neural Network and Text Statistics Features. The figure reveals a strong correlation between predictions and the Number of Words, indicating that a higher word count corresponds to a higher likelihood of the article being of high quality. Additionally, an interesting observation emerges: the Number of Characters negatively correlates with article quality, despite the correlation between the number of words and number of characters. Overall we can see the DNN relies heavily on only a portion of the features.

On the other hand, LGBM, as shown in Figure 23, takes into account almost every feature, regardless of its current value, and clear correlations are observed between the number of words and the number of sentences.

6.2. Models Performance

In the original paper (Wang & Li, 2019, p. 10), the classification precision for all the feature sets was mentioned using their best model, which is the Stacked LSTM. Precision values of 0.80, 0.85, and 0.62 were achieved using the Text Statistics Features set, the Structure Features set, and the Readability Scores Features set, respectively. In this reproduction, the DNN was implemented, and precision values of 0.87, 0.86, and 0.80 were obtained for the same feature sets, as shown in Table 5. The results raised the question of whether the values of the Text Statistics feature set and Structure features set were mistakenly swapped in the original paper, which is also shown by figure 5 in (Wang & Li, 2019, p. 10).

Feature sets	**Text Statistics**	**Structure features**	**Readability Scores**
DNN	0.87	0.86	**0.80**
LGBM	**0.90**	**0.87**	0.71
Stacked LSTM	0.80	0.85	0.62

Table 5: Classification Precision using Text Statistics, Structure Features and Readability Scores

Table 6 compares the performance of two models, DNN (Deep Neural Network) and LGBM (LightGBM), using different feature sets and evaluation metrics. The feature sets used for training and evaluation are Text, Structure, and Readability. The evaluation metrics analyzed include Accuracy, Precision, Recall, TNR (True Negative Rate), and F-Score. The results indicate that both models perform well overall, with DNN achieving an accuracy of 0.83 and LGBM obtaining 0.87. For Precision, LGBM scores higher as well with 0.90 compared to DNN's 0.87. However, LGBM has a much better Recall score of 0.83, while DNN has 0.79. Regarding the True Negative Rate, DNN and LGBM achieve 0.88 and 0.91, respectively. Finally, the F-Score for DNN is 0.83, and for LGBM is 0.86. Overall, the results indicate that LGBM has better overall performance, while DNN has better precision. The choice between them would depend on the specific task requirements and priorities.

Model name	**DNN**			**LGBM**		
Feature set	Text	Structure	Readability	Text	Structure	Readability
Accuracy	0.83	0.81	0.72	**0.87**	0.83	0.69
Precision	0.87	0.86	0.80	**0.90**	0.87	0.71
Recall	0.79	0.73	0.58	**0.83**	0.79	0.66
TNR	0.88	0.88	0.58	**0.91**	0.88	0.73
F-Score	0.83	0.79	0.67	**0.86**	0.82	0.68

Table 6: [
Comparing Classification Performance of Text Statistics, Structure, and Readability Feature Sets]Performance metrics for all three of the Feature Sets with the optimized LGBM and DNN Models. The results show a clear advantage of LGBM across most metrics and feature sets, with the exception of precision and accuracy, where the DNN yields better performance when used with the Readabillity Scores.

6.3. Feature Sets Comparison

The values of the feature sets is examined to better understand the reasoning behind the drastic performance difference observed between them when used as fed used to train the models, as discussed in Section 6.2. In the case of the text statistics feature set, Figure 32 clearly shows significant differences in both the median and mean values, implying that higher values correspond to better articles. On the other hand, the readability features show similar values Figure 18, and for certain cases, such as the Coleman-Lia index, the low-quality articles show higher mean and median values compared to the high-quality articles. This observation clearly shows that the readability score does not positively correlate with the quality of the article, nor does it provide any distinguishing features to improve the performance of the model, which explains the low accuracy results.

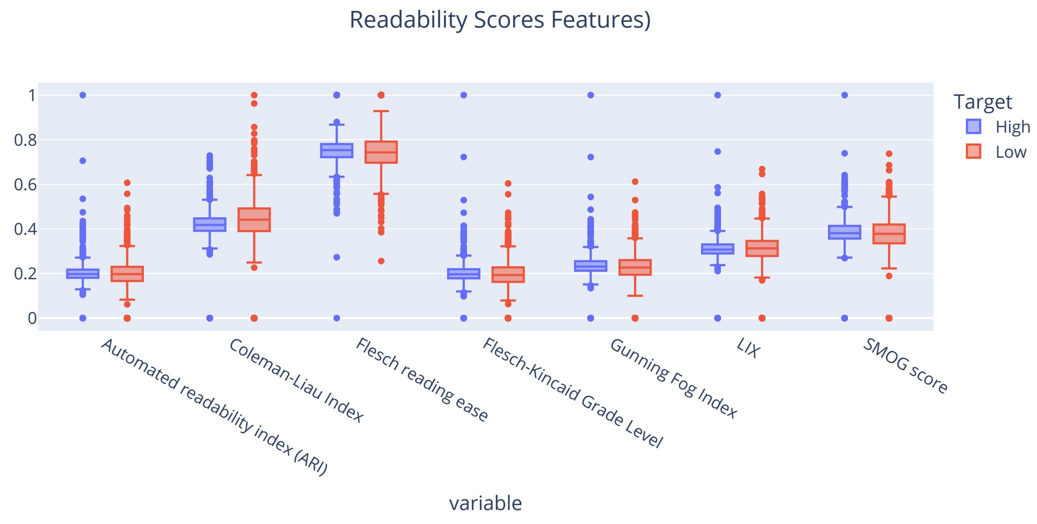


Figure 18: Values distribution of each individual features within the Readability Scores feature set

6.4. Further Testing

The initial goal was to test only three feature sets, namely Text Statistics, Structure, and Readability Scores. However, in this section, the remaining feature sets mentioned in (Wang & Li, 2019) will be covered, along with their performance in the results using the introduced LGBM and DNN.

The feature set that was exactly replicated is the Writing Styles feature set, which was done using spacy. Regarding the Edit History feature set, two slight adjustments have been made. Firstly, the Modified Lines Rate feature, representing the ratio between the number of modified lines and the total number of lines from a revision that is approximately 3 months old, has been replaced. It was determined that the ratio of the size difference in bytes makes more sense in this case. This is due to the fact that references, images, or even templates could be added, which do not contribute to the modified number of lines but could impact the quality of the article.

The feature related to the Quality of Review, which yields a value representing the quality of the reviewer, has been eliminated. This decision has been made due to the extensive data required for calculating this value. Access to such data can be obtained either through the mediawiki API or the revision history dump. However, utilizing the API poses a challenge, as retrieving the necessary data for a dataset comprising random articles might extend over several days depending on how old the article is. On the other hand, the history dump option contains approximately 1.3 billion revisions. Effectively parsing and iterating through such a vast number of revisions demands substantial computational time.

A similar feature is also found within the Network feature set, which is referred to as the Clustering Coefficient. Computing this value requires recursive retrieval of outgoing link data for an article. This means that a significant number of API requests, typically around 600-800, are required for an average sized article. The number of requests can potentially escalate exponentially depending on the size of the recursively linked articles.

It's worth noting that an effort was made to construct a graph including all Wikilinks present in the corresponding dump. This comprehensive graph contained about 1.5 billion edges and about 685 million nodes. However, due to computational limitations, this graph was partitioned into 20 smaller graph chunks, each ranging in size from 2.2 to 3.5 GB. Despite this approach, the entirety of these graph chunks couldn't be merged together as a single connected graph due to computational resource constraints. This particular approach was expected to yield more precise values, particularly for the Page Rank feature.

The results of the additional three feature sets were surprisingly found to be better than those in the original paper, which led to the question of whether there had been data leakage from the evaluation feature set. To investigate this hypothesis, a second evaluation dataset was introduced to test these models. To ensure the absence of leakage, the following approach was adopted: Firstly, all articles from the original training dataset were extracted. Subsequently, a sample consisting of 150 articles was generated, excluding the titles from the original training dataset. This implies that only new articles, unseen by the model during the training and testing, were included, regardless of any potential data leakage in the original scenario. The outcomes of this strategy are presented in Table 7. Despite the notably high precision and TNR achieved by the DNN model, it still demonstrates inferior performance in F1 when compared to LGBM.

	DNN			**LGBM**		
Feature set	Writing Style	Edit History	Network	Writing Style	Edit History	Network
Accuracy	0.85	0.85	0.80	0.83	**0.86**	0.83
Precision	0.92	**0.95**	0.84	0.90	0.88	0.89
Recall	0.77	0.75	0.75	0.75	**0.84**	0.75
F1	0.84	0.84	0.79	0.82	**0.86**	0.81
TNR	0.93	**0.96**	0.85	0.78	0.88	0.91

Table 7: Performance metrics for all three of the feature sets with the optimized LGBM and DNN Models. The results demonstrate that Light Gradient-Boosting Machine excels when applied with Network Features, while the Deep Neural Network performs better with Writing Style Features. For Edit History, Light Gradient-Boosting Machine maintains a slight advantage, except in precision and TNR, where the Deep Neural Network shows excellent performance.

After conducting separate evaluations on all individual feature sets, pair-wise combinations of these sets were also examined. Utilizing all six feature sets, 15 pair-wise combinations were created, in addition to testing all sets together. The performance overall demonstrated consistency, even when the readability score was combined with any other feature set. The model that is used for those evaluation is the base line LGBM, meaning that there was no optimization involved.

In terms of precision, the combination of all the feature sets yielded the best results achieving 93% which is almost as good as the Edit history precision when used along side the DNN, but along side better results from the other metrics.

In order to compare the results across all metrics with the best model presented in (Wang & Li, 2019, p. 7) in Table 8, the training of the model must make use of all feature sets. In this setup, optimization was performed on both the LGBM and DNN models to obtain the most optimal results.

Model Name	**LGBM**	**DNN**	**Stacked LSTM**
Accuracy	**0.90**	0.83	0.80
Precision	**0.93**	0.92	0.82
Recall	**0.87**	0.73	0.78
TNR	**0.93**	**0.93**	0.81
F1	**0.90**	0.81	0.80

Table 8: Comparing classification performance of the presented DNN and LGBM Models with the best model (Stacked LSTM) from (Wang & Li, 2019) , using all of the presented feature sets combined. It is clear that LGBM is superior to the other models when used fed all of the features together.

The final results are shown in Table 8, where LGBM is the superior model across all the metrics. Although the DNN provides remarkably high precision, its performance falls short in other aspects. This implies that the DNN's superiority over the stacked LSTM isn't assured, as the LSTM maintains its edge when considering other metrics.

	LGBM	**DNN**	**Stacked LSTM**
Top 1	Discussion Count	Modified size rate	Prob_review
Top 2	Avg. words per paragraph	Number of characters	Avg_edit_per_user
Top 3	Abstract size	Number of words	Translations count
Top 4	Long phrase rate	Avg. section length	IP count
Top 5	Std. edit number per user	Review Count	pronouns count

Table 9: Top five most important features comparison between the deep learning models

In terms of feature importance, each of the three models had a distinct top five. However, due to the lack of information on the feature importance methodology used in the original research, it is not possible to accurately determine the impact of each feature. As a consequence of this limitation, only the ranking of feature importance is presented in Table 9. Nevertheless, the actual SHAP scores, along with the top 12 features for each model presented in this study, are visually represented in Figure 33.

Overall, LGBM has great good results; however, there is potential for improvement. Therefore, an additional experiment will be conducted to attempt to further refine its performance. As described in Section 2.3, there are studies that have shown strong performance using only one feature, namely an embedding of the entire Wikipedia article. In this approach, the unprocessed content of the page is fed into an embedding model, resulting in a vector that encodes all the content of the page as a numerical value, including potential features that are not directly integrated into the existing feature sets.

Adding the additional embedding features have shown almost no improvement in the results, as shown in Table 10, by including embedding vector in conjunction with all of the previously mentioned feature sets. This aggregation results in a total of 370 features input into the model. It is worth noting that LGBM using only the embedding features yielded an accuracy of only 76%.

Accuracy	Precision	Recall	TNR	F1
0.91	0.92	0.88	0.92	0.90

Table 10: Classification performance of LGBM using the combined feature sets and Doc2Vec embedding all together. The results indicate minimal to no improvement when compared to using only the feature sets without the embeddings.

The extraction of the Doc2Vec embedding was performed using Spacy, using the large model which uses BERT, namely “en_core_web_lg”[2]. This decision was made because the smaller language model yielded 25 dimensions in the retrieved embedding, while the larger model yielded 300 dimensions, thus providing the models with a richer set of features to work with.

7. Results

This research mainly aimed to partially replicate the deep-learning-based approach proposed by Wang & Li in assessing the quality of Wikipedia articles, classifying them as either high or low-quality (Wang & Li, 2019). The study introduced two models, namely Deep Neural Network (DNN) and Light Gradient Boosting Machine (LGBM), with DNN being used in the original paper as well. However, instead of utilizing all six feature sets from the original work, this study mainly focused on three specific feature sets: Text Statistics, Structure, and Readability Scores. These same feature sets were used to evaluate the precision of the best model introduced in the original research.

The dataset used in this work consisted of precisely 3150 English Wikipedia articles, with an equal distribution of 525 articles for each grading level in the Wikipedia grading system. Out of these, 3000 articles were used for training and testing, which underwent K-fold cross-validation with k set as 5, while the remaining 150 articles were reserved for evaluation.

[2] https://spacy.io/models/en#en_core_web_lg

Both the DNN and LGBM models demonstrated improved precision when using their respective feature sets compared to the best model in the original paper, which was the Stacked LSTM, achieving precision rates of 89% and 86% using the Text Statistics Feature set, respectively. In comparison, the Stacked LSTM resulted in a precision of 80%.

The detailed comparison between the two models revealed their individual strengths and weaknesses. The DNN model showed superior precision, indicating its ability to correctly classify a high proportion of actual high-quality items. On the other hand, the LGBM model achieved better recall and true negative rates, indicating its effectiveness in identifying a higher number of true low-quality items while minimizing the instances of false negatives.

Ultimately, the choice of model should be based on the specific requirements and priorities of the classification task at hand. If precision is paramount, the DNN model would be preferred, whereas if recall and true negative rate are more critical, the LGBM model would be the better option. Understanding these trade-offs can help make informed decisions when applying these models to assess the quality of Wikipedia articles.

It was evident from the evaluation of the feature sets that the Text Statistics feature set performed the best among all three of the proposed feature sets, followed by the Structure feature set, and lastly, the Readability Score feature set. The drastic difference in performance between the Readability Score feature set and the other two sets is attributed to the contradiction of the meaning of the readability scores presented in the feature set, indicating that some of the scores do not correlate with one another.

The optimization and training time, the fastest model was the LGBM utilizing the GBDT boosting type, averaging 1 minute and 42 seconds. It was followed by the DNN, which took an average of 26 minutes and 50 seconds, while the slowest was the LGBM using the DART boosting type, resulting in an average computational time of one hour. The drastic difference in results can be attributed to factors such as GPU inefficiency when using the DNN and the absence of early stopping in the DART LGBM due to its monotony.

The additional testing, as described in Section 6.4, provided greater clarity and highlighted the superiority of the proposed LGBM and DNN models over those used in the original paper. This observation was particularly true when the models were optimized and used in conjunction with all six feature sets.

The final experiment using only LGBM, coupled with a novel feature not present in the original paper, namely doc2Vec word embedding, yielded no further performance advancements. The overall outcome resulted in an average improvement of around 11% across all metrics, indicating a significant performance improvement.

Finally, the experiments were conducted on machines equipped with a GTX 1080 8GB GPU, an AMD Ryzen 5 5600X 6-Core processor, and 32GB of 3600 MHZ DDR4 RAM. As for the multi-threaded experiments, only a total of 6 threads out of the available 12 threads were used. The source code, datasets and models that are used in this study all could be found under Wikipediatools[1] repository on Github.

[1] https://github.com/MohammadSakhnini/WikipediaTools

7.1. Findings and Discussion

The aim of this partial reproduction was to replicate and evaluate the performance of the models presented in the original paper (Wang & Li, 2019), which was found to be more challenging than expected due to several reasons, mainly the lack of proper explanation and reasoning behind the usage of certain models, such as LSTMs and CNNs, which are commonly employed in the case of a dataset containing sequence data that, which in this case, was not provided. Furthermore, crucial details for the training of the models were either missing or only briefly introduced, such as the hyperparameters used for the training. The number of epochs was set to 15 for the training, even though the figures displaying the accuracy progress of the models indicate that the models may have not yet converged, leading to questions about the rationale behind using this specific number of epochs.

An attempt was made to accurately reproduce the DNN presented in the original study, wherein the provided hyperparameters (epoch 15, dropout 0.2, and batch size 195) were exactly used as provided. However, this resulted in a performance that was found to be very underwhelming.

Furthermore, it is suggested by the classification precision in the original work that the Structure feature set outperforms the Text Statistics feature set, as opposed to what is demonstrated in this work. The finding from this work is also supported by Figure 5 (Wang & Li, 2019, p. 10), which shows that the accuracy of the Text Statistics features is, in fact, outperformed by the Structure feature set, raising the question of whether the values have been mistakenly swapped. It is worth noting that an attempt was made to contact one of the authors, Xiaodan Li, requesting more information regarding the models and whether the values were indeed misplaced, but no answer was received.

7.2. Limitations & Feature Work

The reproduction was based on a very small portion of assessed articles provided by Wikipedia and the collected dataset, suggesting the possibility of further improvement by increasing the dataset size.

In addition, the experiment only tested one new additional feature on top of the features originally used in the paper. However, Wikipedia content contains several other potential features that could be explored, such as templates and protection levels, thus offering many opportunities for further investigation and potential improvement.

In addition, the models were exclusively tested on English Wikipedia articles, leaving room for further testing to determine if the feature sets yield similar results across different languages.

The quality of the Wikipedia articles was not assessed based on their original gradings provided by Wikipedia; instead, they were classified as high or low-quality articles. Improvements can be made to enhance the model's ability to classify the actual gradings of the articles.

8. Conclusion

In this study, the deep learning-based approach proposed by (Wang & Li, 2019) to evaluate the quality of Wikipedia articles and classify them as high or low quality was partially replicated and extended. Two models, the Deep Neural Network (DNN) and the Light Gradient Boosting Machine (LGBM), were introduced, and three specific feature sets, namely text statistics, structure, and readability scores, were focused on. Valuable insights into the performance of the models, the importance of the features, and potential areas for improvement were gained through extensive evaluation and analysis.

The results showed that both the DNN and LGBM models outperformed the top model presented in the original paper. In fact, both models showed promising results across all feature sets. However, LGBM showed a clear advantage when used in conjunction with any of the feature sets, except for readability scores, where the DNN clearly outperformed.

Furthermore, the evaluation of the feature sets showed that the Text Statistics and Edit History sets had the best performance among the six proposed sets. They were followed by the Structure, Network, and Writing Style sets. In contrast, the Readability Scores feature set showed less promising results. However, the combination of all feature sets produced the best overall performance. This highlights the importance of understanding the strengths and weaknesses of each feature set, which can provide valuable guidance to researchers and practitioners in selecting the most relevant features tailored to their specific use cases.

Challenges were encountered during the replication process due to missing details in the original paper regarding model hyperparameters and explanations for certain model choices. Nevertheless, the replication efforts provided valuable insights into the performance and feature importance of the DNN and LGBM models.

In conclusion, this study contributes to the field of Wikipedia article quality assessment using machine learning models. The models presented here can be further refined and extended to improve the accuracy. By incorporating more diverse datasets and further fine-tuning the models, the reliability of article quality ratings can be increased, ultimately benefiting Wikipedia editors and readers alike. Continued research in this area will pave the way for more sophisticated and robust approaches to evaluating the quality of online content and supporting the dissemination of knowledge on the internet.

A. Appendix

Text Statistics Features Set		
Parameter	**GBDT**	**DART**
boosting_type	gbdt	dart
class_weight	null	null
colsample_bytree	0.533	0.803
importance_type	split	split
learning_rate	0.363	0.007
max_depth	15	12
min_child_samples	101	38
min_child_weight	0.001	0.001
min_split_gain	4.565e-6	2.5e-8
n_estimators	900	3150
n_jobs	6	6
num_leaves	31	56
objective	binary	binary
random_state	42	42
reg_alpha	5.421	0.299
reg_lambda	0.001	4.275e-5
silent	warn	warn
subsample	0.929	0.860
subsample_for_bin	200000	200000
subsample_freq	4	14
device	gpu	gpu
metric	0.846	auc
verbose	-1	-1
scale_pos_weight	0.846	0.938

Table 11: Best hyperparameters for GBDT and DART using Text Statistics Features set

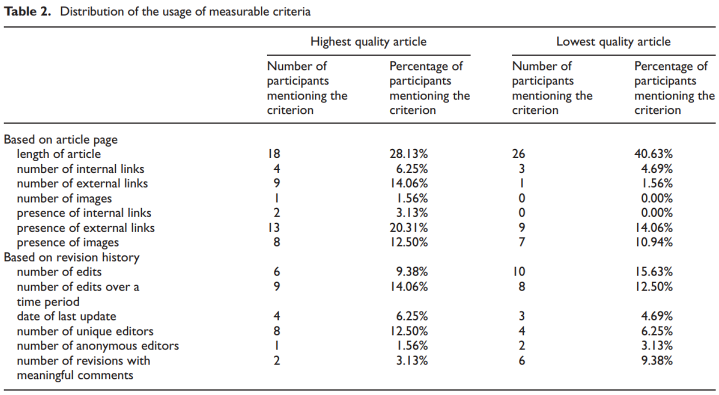


Table 2. Distribution of the usage of measurable criteria

	Highest quality article		Lowest quality article	
	Number of participants mentioning the criterion	Percentage of participants mentioning the criterion	Number of participants mentioning the criterion	Percentage of participants mentioning the criterion
Based on article page				
length of article	18	28.13%	26	40.63%
number of internal links	4	6.25%	3	4.69%
number of external links	9	14.06%	1	1.56%
number of images	1	1.56%	0	0.00%
presence of internal links	2	3.13%	0	0.00%
presence of external links	13	20.31%	9	14.06%
presence of images	8	12.50%	7	10.94%
Based on revision history				
number of edits	6	9.38%	10	15.63%
number of edits over a time period	9	14.06%	8	12.50%
date of last update	4	6.25%	3	4.69%
number of unique editors	8	12.50%	4	6.25%
number of anonymous editors	1	1.56%	2	3.13%
number of revisions with meaningful comments	2	3.13%	6	9.38%

Figure 19: Distribution of the usage of measurable criteria (Yaari et al., 2011, p. 493)

Structure Features Set		
Paramter	**GBDT**	**DART**
boosting_type	gbdt	dart
class_weight	null	null
colsample_bytree	0.742	0.604
importance_type	split	split
learning_rate	0.0244	0.002
max_depth	15	7
min_child_samples	226	39
min_child_weight	0.001	0.001
min_split_gain	5.771e-8	0.0252
n_estimators	3900	3100
n_jobs	6	6
num_leaves	108	131
objective	binary	binary
random_state	42	42
reg_alpha	4.354e-5	2.727e-7
reg_lambda	0.001	2.1586e-8
silent	warn	warn
subsample	0.475	0.949
subsample_for_bin	200000	200000
subsample_freq	4	32
device	gpu	gpu
metric	auc	auc
verbose	-1	-1
scale_pos_weight	0.856	0.976

Table 12: Best hyperparameters for GBDT and DART using Structure Features set

Readability Features Set		
Parameter	**GBDT**	**DART**
boosting_type	gbdt	dart
class_weight	null	null
colsample_bytree	0.927	0.768
importance_type	split	split
learning_rate	0.410	0.146
max_depth	11	7
min_child_samples	151	64
min_child_weight	0.001	0.001
min_split_gain	0.002	4.728e-6
n_estimators	500	500
n_jobs	6	6
num_leaves	341	66
objective	binary	binary
random_state	42	42
reg_alpha	1.903e-7	9.271e-4
reg_lambda	1.422e-8	42.736
silent	warn	warn
subsample	0.947	0.842
subsample_for_bin	200000	200000
subsample_freq	27	10
device	gpu	gpu
metric	auc	auc
verbose	-1	-1
scale_pos_weight	0.965	0.973

Table 13: Best hyperparameters for GBDT and DART using Readability Features set

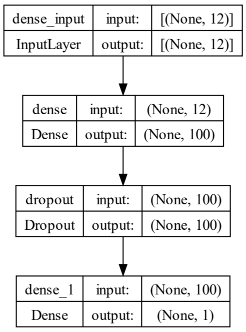


Figure 20: Architecture of the Deep Neural Network using Text Statistics Features

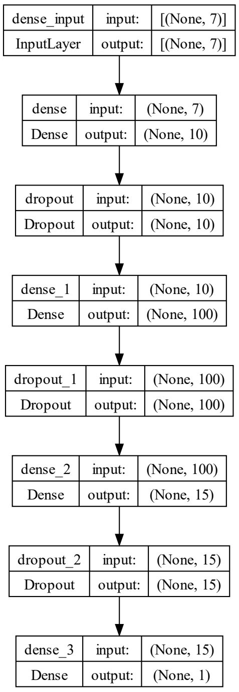


Figure 21: Architecture of the Deep Neural Network using Readability Features

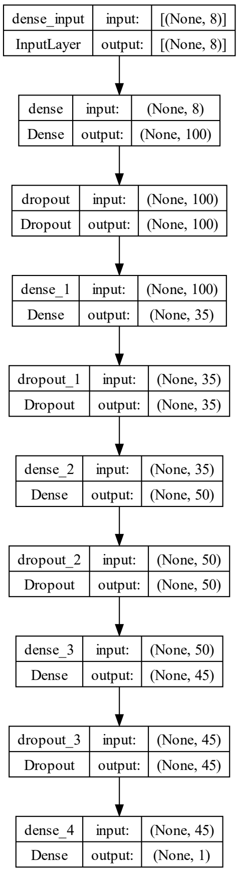

dense_input | input: | [(None, 8)]
InputLayer | output: | [(None, 8)]
dense | input: | (None, 8)
Dense | output: | (None, 100)
dropout | input: | (None, 100)
Dropout | output: | (None, 100)
dense_1 | input: | (None, 100)
Dense | output: | (None, 35)
dropout_1 | input: | (None, 35)
Dropout | output: | (None, 35)
dense_2 | input: | (None, 35)
Dense | output: | (None, 50)
dropout_2 | input: | (None, 50)
Dropout | output: | (None, 50)
dense_3 | input: | (None, 50)
Dense | output: | (None, 45)
dropout_3 | input: | (None, 45)
Dropout | output: | (None, 45)
dense_4 | input: | (None, 45)
Dense | output: | (None, 1)

Figure 22: Architecture of the Deep Neural Network using Structure Features

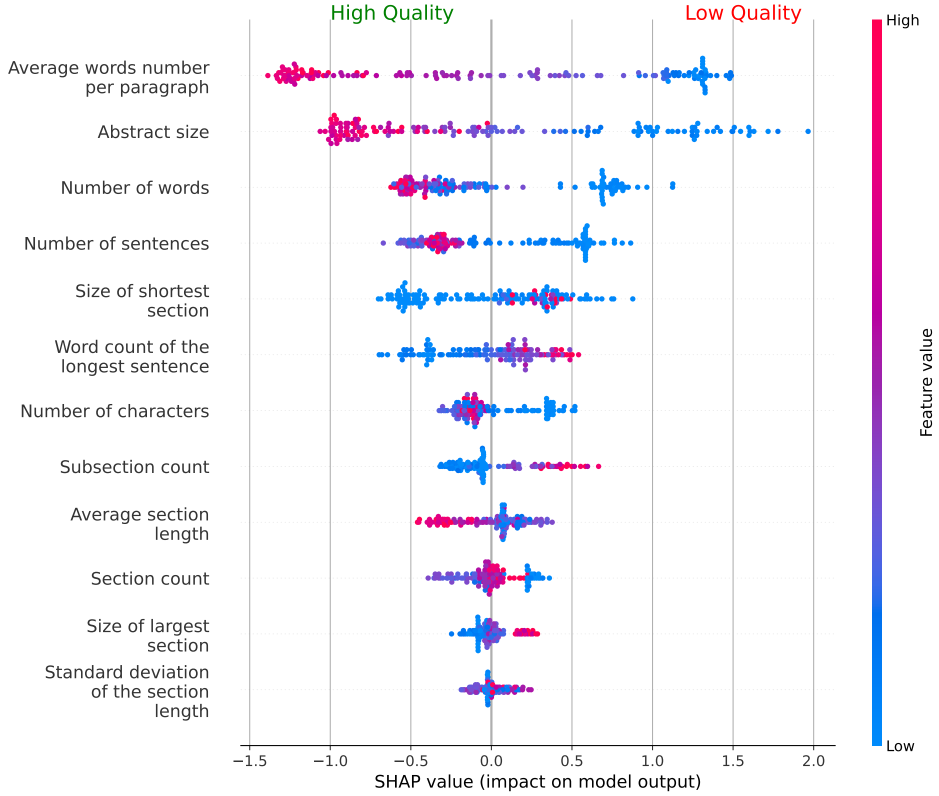


Figure 23: Overview of the features importance using LightGBM and Text Statistics Features. In contrast to the DNN, LGBM considers nearly all features when making predictions. Notably, the figure reveals that the Size of the Shortest Section and Word Count of the Longest Sentence show a clear negative correlation with article quality.

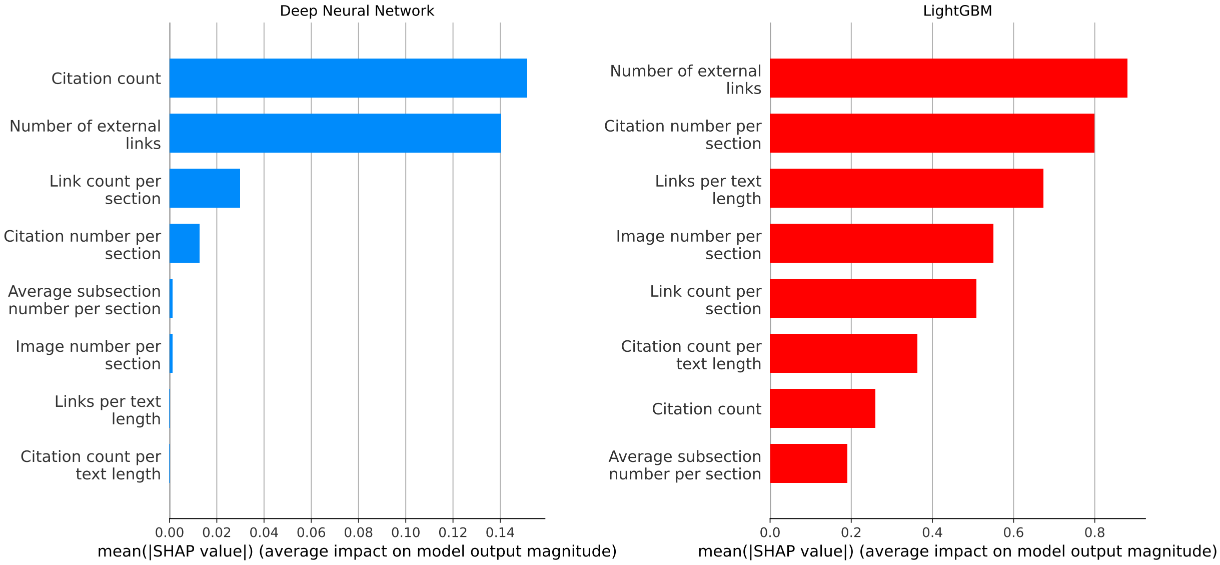


Figure 24: Global features importance of the Deep Neural Network and LightGBM using Structure Features. The Deep Neural Network predominantly relies on two features, namely, Citations Count and Number of External Links, for its predictions. In contrast, LGBM uses the entirety of the feature set to make predictions.

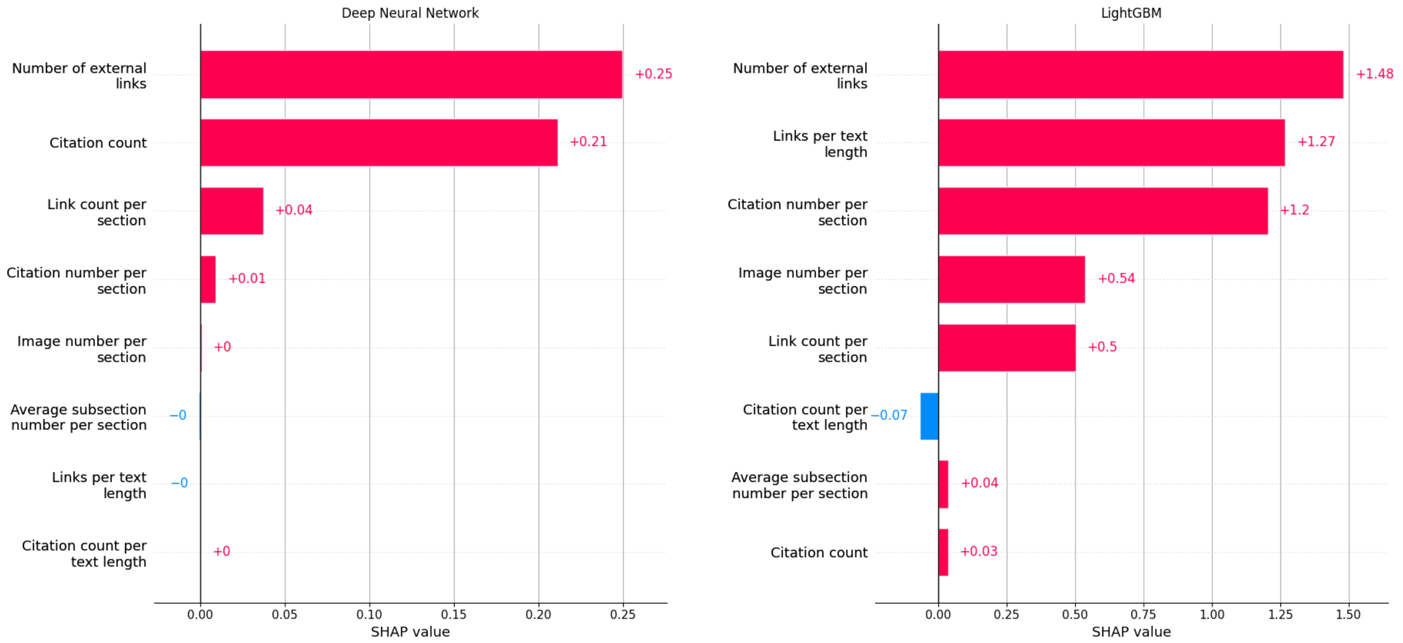


Figure 25: Local features importance of the Deep Neural Network and LightGBM using Structure Features

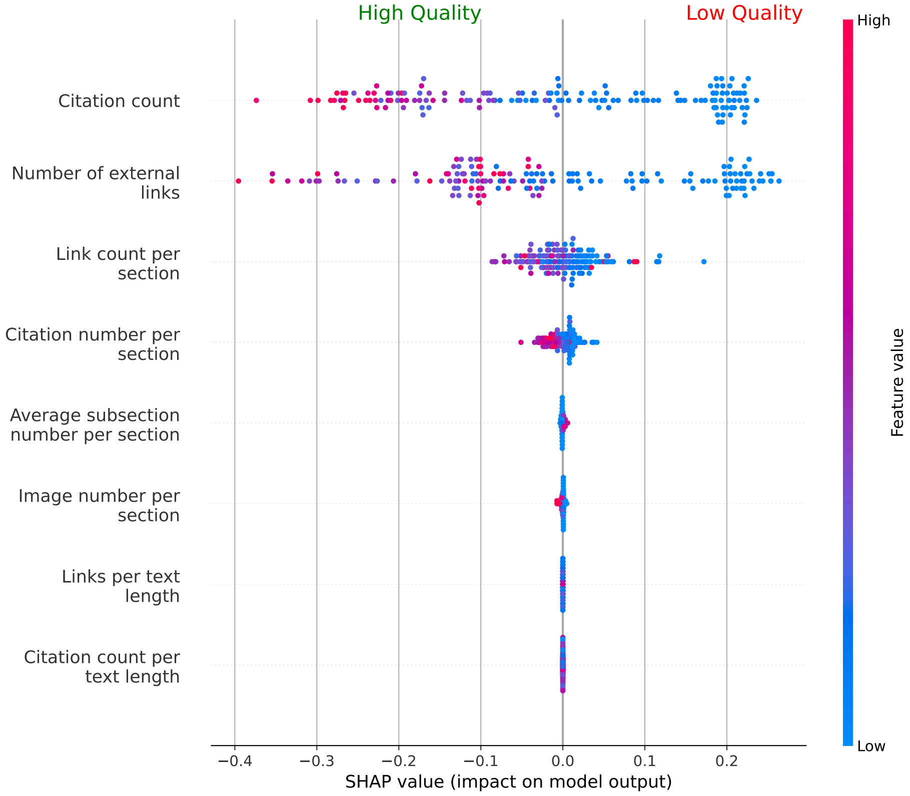


Figure 26: Overview of the features importance using the Deep Neural Network and Structure Features. As shown, the DNN predominantly relies on two features, namely, Citations Count and Number of External Links, for its predictions regardless of the value of the feature is.

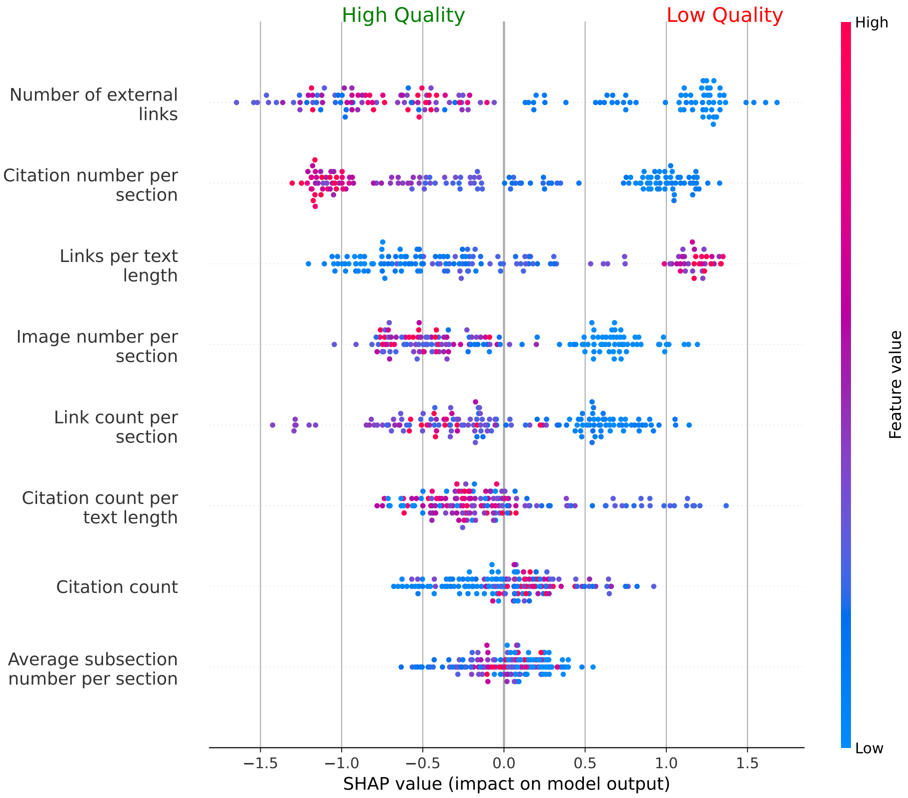


Figure 27: Overview of the features importance using the LightGBM and Structure Features. An interesting observation can be seen; the number of citations alone does not directly indicate that an article is of high quality. Instead, it is the number of citations per section that seems to play a more crucial role. This suggests that an article may not be considered high quality if not all of its sections have an appropriate number of citations, even if the overall citation count is high.

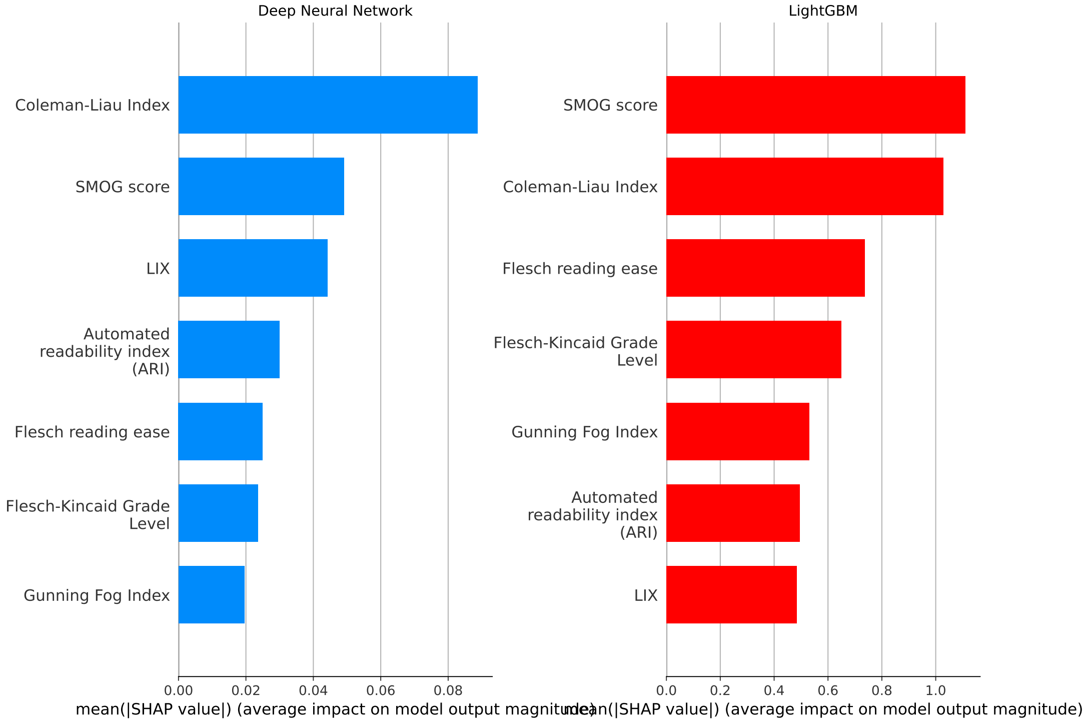


Figure 28: Global features importance of the Deep Neural Network and LightGBM using Readability Features

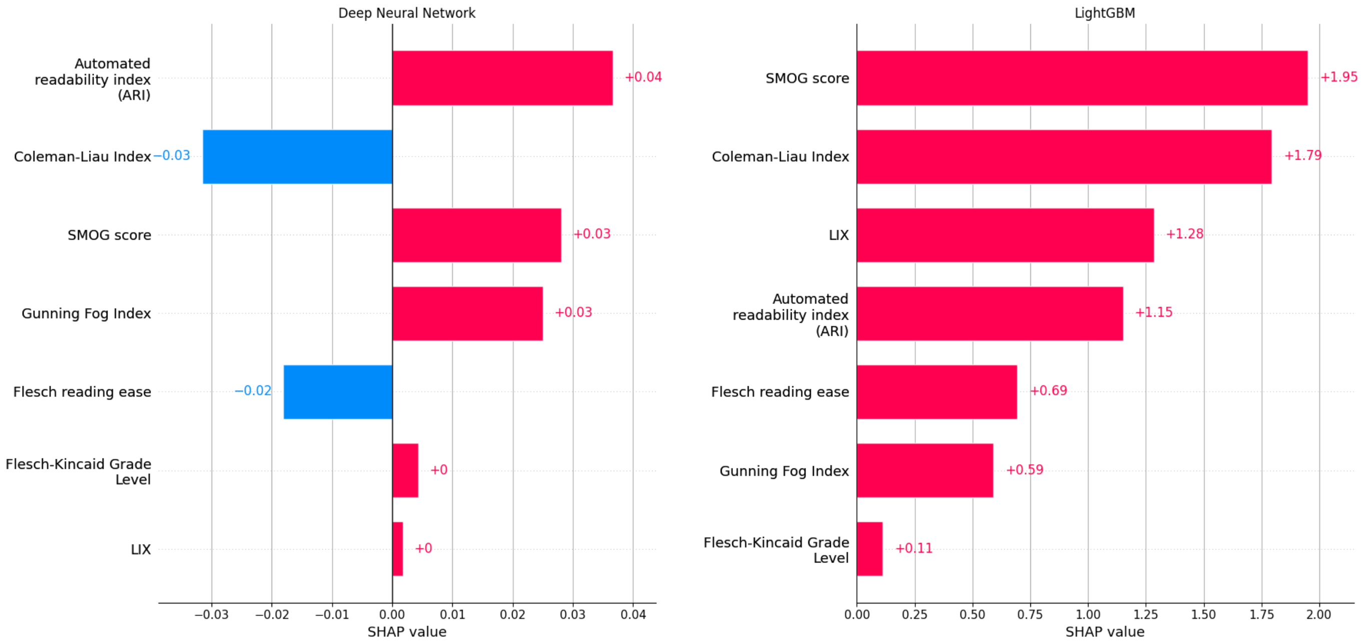


Figure 29: Local features importance of the Deep Neural Network and LightGBM using Readability Features

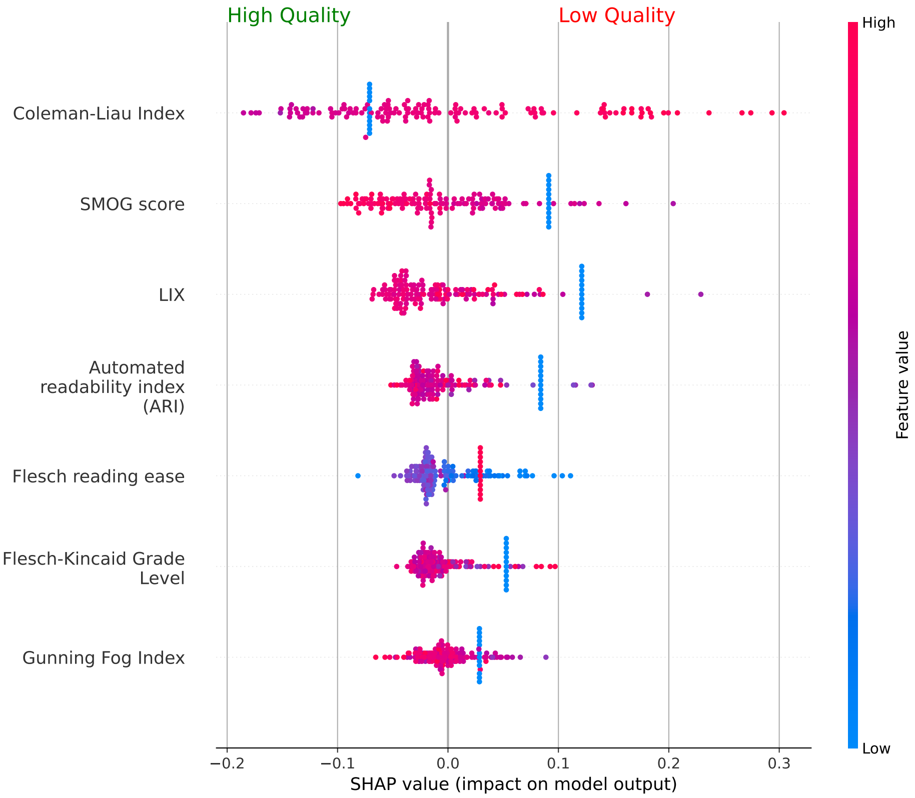


Figure 30: Overview of the features importance using the Deep Neural Network and Readability Features

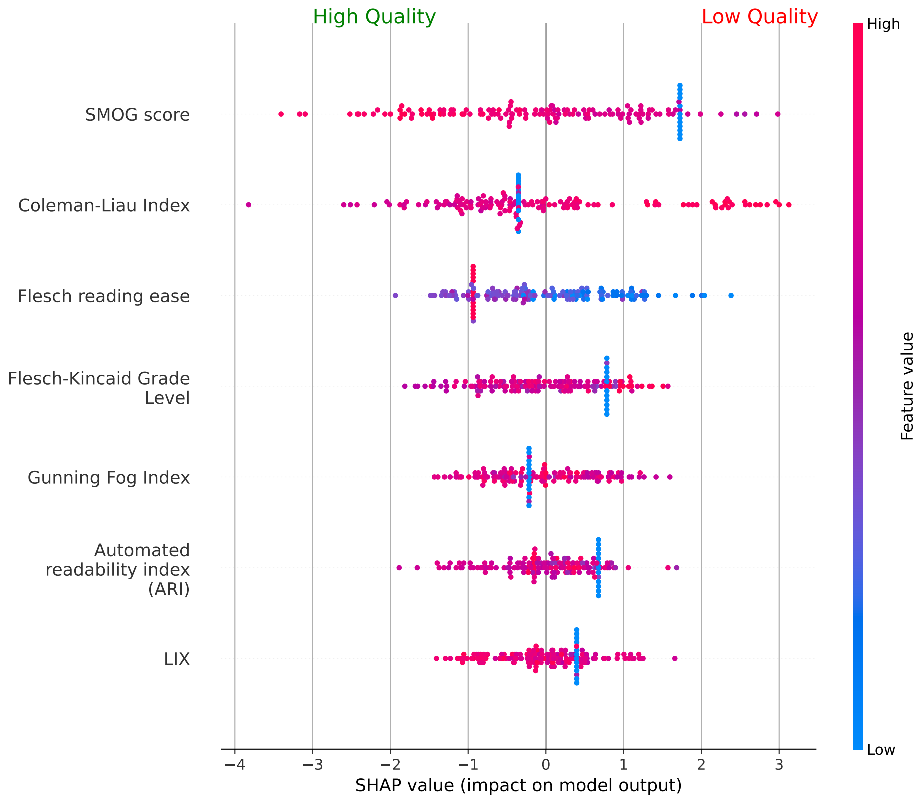


Figure 31: Overview of the features importance using the LightGBM and Readability Features

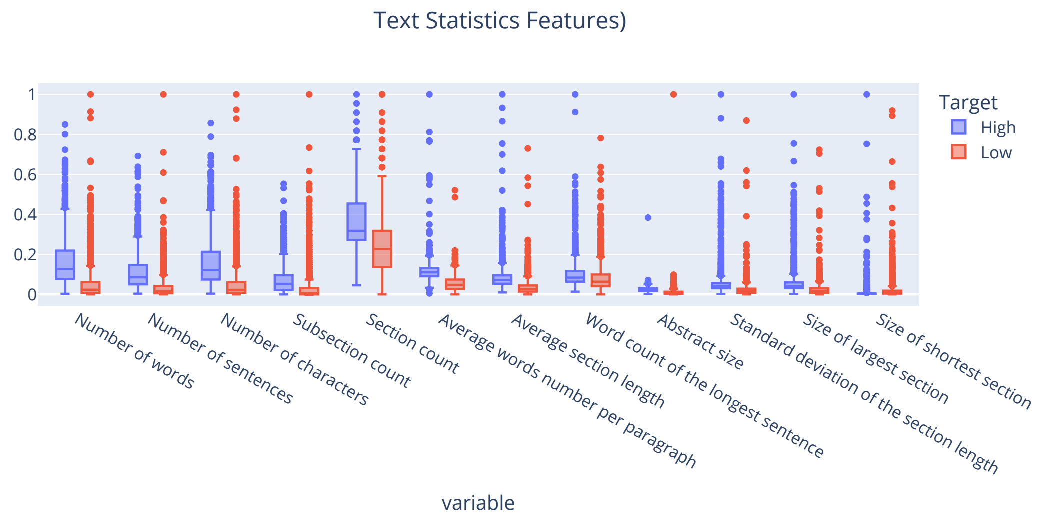


Figure 32: Values distribution of each individual feature within the Text Statistics feature set

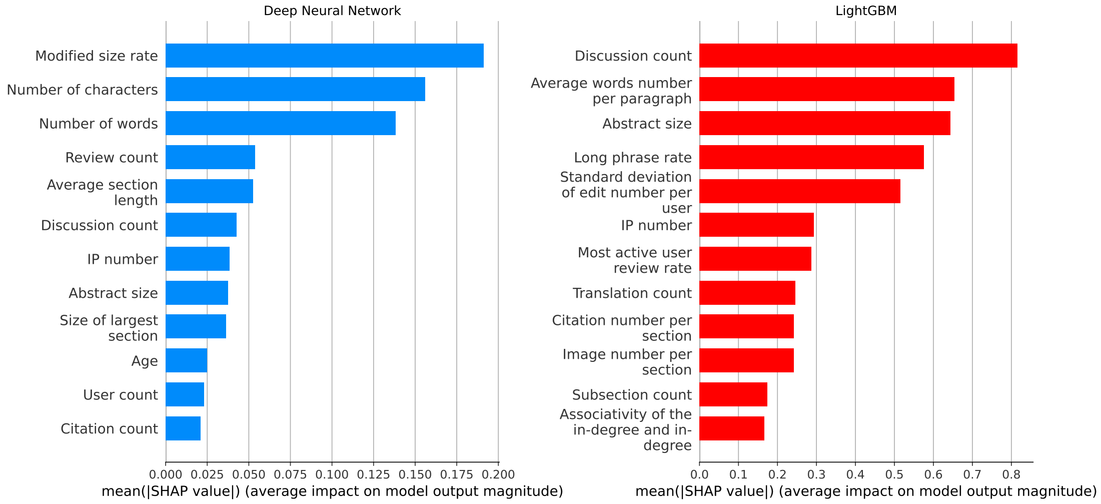


Figure 33: Top 12 global feature importance of the Deep Neural Network and LightGBM Models using all of the feature sets combined.

Feature	**Feature Value**	**Average (Low Quality)**
Number of words	14855	1567
Number of sentences	758	73
Number of characters	93389	9999
Subsection count	18	3.29
Section count	11	5.59
Avg. words per paragraph	49	19
Avg. section length	3111	810
Word count of the longest sentence	325	76
Abstract size	3838	818
Std. of the section length	2379	846
Size of largest section	10210	2769
Size of shortest section	5	45

Table 14: The false positive data sample used to create to local bar plot in Figure 16. We can see that the values of this Low Quality article are exceptionally high when compared with the average indicating that this datasample is an extreme outlier.

Features	**Accuracy**	**Precision**	**Recall**	**F1**	**TNR**
Text Statistics & Structure	0.87	0.88	0.87	0.87	**0.87**
Text Statistics & Readability Scores	0.83	0.87	0.79	0.83	0.80
Text Statistics & Writing Style	0.87	0.88	0.87	0.87	**0.87**
Text Statistics & Edit History	0.88	0.91	0.84	0.87	0.85
Text Statistics & Network	0.83	0.85	0.81	0.83	0.82
Structure & Readability Scores	0.85	0.87	0.81	0.84	0.82
Structure & Writing Style	**0.89**	0.92	**0.87**	**0.89**	**0.87**
Structure & Edit History	**0.89**	0.90	**0.87**	0.88	**0.87**
Structure & Network	0.83	0.87	0.79	0.83	0.80
Readability Scores & Writing Style	0.86	0.89	0.83	0.86	0.84
Readability Scores & Edit History	0.87	0.91	0.81	0.86	0.83
Readability Scores & Network	0.82	0.86	0.76	0.81	0.79
Writing Style & Edit History	**0.89**	0.93	0.85	0.89	0.86
Writing Style & Network	0.85	0.88	0.80	0.84	0.82
Edit History & Network	0.88	0.91	0.84	0.87	0.85
All of the feature sets	**0.89**	**0.94**	0.84	**0.89**	0.86

Table 15: Evaluating Performance of Feature Set Pairs and All the Feature Sets Combined using LightGBM without Optimization.

Bibliography

Adler, B. T., & de Alfaro, L. (2007). A content-driven reputation system for the wikipedia. In C. Williamson, M. E. Zurko, P. Patel-Schneider, & P. Shenoy (Eds.), *Proceedings of the 16th international conference on world wide web* (pp. 261–270). ACM. https://doi.org/10.1145/1242572.1242608. (Cit. on p. 5)

Altmann, A., Toloşi, L., Sander, O., & Lengauer, T. (2010). Permutation importance: a corrected feature importance measure. *Bioinformatics*, *26*(10), 1340–1347. https://doi.org/10.1093/bioinformatics/btq134 (cit. on p. 27)

Ball, C. (2023). Defying easy categorization: Wikipedia as primary, secondary and tertiary resource. *Insights the UKSG journal*, *36*. https://doi.org/10.1629/uksg.604 (cit. on p. 3)

Blikstad-Balas, M. (2016). "you get what you need" : A study of students' attitudes towards using wikipedia when doing school assignments. *Scandinavian Journal of Educational Research*, *60*(6), 594–608. https://doi.org/10.1080/00313831.2015.1066428 (cit. on p. 2)

Blumenstock, J. E. (2008). Size matters. In J. Huai, R. Chen, H.-W. Hon, Y. Liu, W.-Y. Ma, A. Tomkins, & X. Zhang (Eds.), *Proceedings of the 17th international conference on world wide web* (pp. 1095–1096). ACM. https://doi.org/10.1145/1367497.1367673. (Cit. on pp. 4, 17)

Brown, A. R. (2011). Wikipedia as a data source for political scientists: Accuracy and completeness of coverage. *PS: Political Science & Politics*, *44*(2), 339–343. https://doi.org/10.1017/S1049096511000199 (cit. on p. 3)

Chen, T., & Guestrin, C. (2016). Xgboost. In B. Krishnapuram, M. Shah, A. Smola, C. Aggarwal, D. Shen, & R. Rastogi (Eds.), *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794). ACM. https://doi.org/10.1145/2939672.2939785. (Cit. on pp. 6, 23)

Chesney, T. (2006). An empirical examination of wikipedia's credibility. *First Monday*. https://doi.org/10.5210/FM.V11I11.1413 (cit. on p. 4)

Content assessment. (10/07/2023). https://en.wikipedia.org/wiki/Wikipedia:Content_assessment. (Cit. on pp. 2, 7, 8, 10, 11)

Dang, Q. V., & Ignat, C.-L. (2016). Quality assessment of wikipedia articles: A deep learning approach by quang vinh dang and claudia-lavinia ignat with martin vesely as coordinator. *ACM SIGWEB Newsletter*, *2016*(Autumn), 1–6. https://doi.org/10.1145/2996442.2996447 (cit. on pp. 5, 6)

Dang, Q.-V., & Ignat, C.-L. (2017). An end-to-end learning solution for assessing the quality of wikipedia articles. In L. Morgan (Ed.), *Proceedings of the 13th international symposium on open collaboration* (pp. 1–10). ACM. https://doi.org/10.1145/3125433.3125448. (Cit. on p. 6)

Del Garcia Valle, E. P., Lagunes Garcia, G., Prieto Santamaria, L., Zanin, M., Menasalvas Ruiz, E., & Rodriguez Gonzalez, A. (2018). Evaluating wikipedia as a source of information for disease understanding. *2018 IEEE 31st International Symposium* *on Computer-Based Medical Systems (CBMS)*, 399–404. https://doi.org/10.1109/CBMS.2018.00076 (cit. on p. 4)

Dzendzik, D., Foster, J., & Vogel, C. (2021). English machine reading comprehension datasets: A survey. In M.-F. Moens, X. Huang, L. Specia, & S. W.-t. Yih (Eds.), *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 8784–8804). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.693. (Cit. on p. 3)

Fallis, D. (2008). Toward an epistemology of wikipedia. *Journal of the American Society for Information Science and Technology*, *59*(10), 1662–1674. https://doi.org/10.1002/asi.20870 (cit. on p. 3)

Jemielniak, D. (2019). Wikipedia: Why is the common knowledge resource still neglected by academics? *GigaScience*, *8*(12). https://doi.org/10.1093/gigascience/giz139 (cit. on p. 3)

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, *30*, 3146–3154 (cit. on p. 23).

Kousha, K., & Thelwall, M. (2017). Are wikipedia citations important evidence of the impact of scholarly articles and books? *Journal of the Association for Information Science and Technology*, *68*(3), 762–779. https://doi.org/10.1002/asi.23694 (cit. on p. 2)

Lamprecht, D., Lerman, K., Helic, D., & Strohmaier, M. (2017). How the structure of wikipedia articles influences user navigation. *The new review of hypermedia and multimedia*, *23*(1), 29–50. https://doi.org/10.1080/13614568.2016.1179798 (cit. on pp. 14, 18)

Laughlin, G. H. M. (1969). Smog grading-a new readability formula. *Journal of Reading*, *12*(8), 639–646. Retrieved July 3, 2023, from http://www.jstor.org/stable/40011226 (cit. on p. 19)

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems 30* (pp. 4765–4774). Curran Associates, Inc. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf. (Cit. on p. 27)

Meishar-Tal, H. (2015). Teachers' use of wikipedia with their students. *Australian Journal of Teacher Education*, *40*(12). https://doi.org/10.14221/ajte.2015v40n12.9 (cit. on p. 2)

Mesgari, M., Okoli, C., Mehdi, M., Nielsen, F. Å., & Lanamäki, A. (2015). "the sum of all human knowledge": A systematic review of scholarly research on the content of wikipedia. *Journal of the Association for Information Science and Technology*, *66*(2), 219–245. https://doi.org/10.1002/asi.23172 (cit. on p. 4)

Nielsen, F. A. (2007). Scientific citations in wikipedia. *First Monday*. https://doi.org/10.5210/FM.V12I8.1997 (cit. on p. 3)

Rajbahadur, G. K., Wang, S., Oliva, G. A., Kamei, Y., & Hassan, A. E. (2022). The impact of feature importance methods on the interpretation of defect classifiers.

IEEE Transactions on Software Engineering, *48*(7), 2245–2261. https://doi.org/10.1109/TSE.2021.3056941 (cit. on p. 27)

Rector, L. H. (2008). Comparison of wikipedia and other encyclopedias for accuracy, breadth, and depth in historical articles. *Reference Services Review*, *36*, 7–22. https://doi.org/10.1108/00907320810851998 (cit. on p. 3)

Ruprechter, T., Santos, T., & Helic, D. (2020). Relating wikipedia article quality to edit behavior and link structure. *Applied Network Science*, *5*(1). https://doi.org/10.1007/s41109-020-00305-y (cit. on pp. 2, 5)

Schmidt, M., & Zangerle, E. (2019). Article quality classification on wikipedia. In B. Lundell, J. Gamalielsson, L. Morgan, & G. Robles (Eds.), *Proceedings of the 15th international symposium on open collaboration* (pp. 1–8). ACM. https://doi.org/10.1145/3306446.3340831. (Cit. on p. 6)

Smith, E. A., & Senter, R. (1967). Automated readability index. *AMRL-TR. Aerospace Medical Research Laboratories*, 1–14 (cit. on p. 19).

Suzuki, Y. (2015). Quality assessment of wikipedia articles using h-index. *Journal of Information Processing*, *23*(1), 22–30. https://doi.org/10.2197/ipsjjip.23.22 (cit. on p. 5)

Wang, P., & Li, X. (2019). Assessing the quality of information on wikipedia: A deep–learning approach. *Journal of the Association for Information Science and Technology*, *71*(1), 16–28. https://doi.org/10.1002/asi.24210 (cit. on pp. iii, 2, 6, 17–19, 22, 23, 26, 27, 30, 32, 34, 35, 37, 38)

Wikipedia template. (23/06/2023). https://en.wikipedia.org/wiki/Help:Template. (Cit. on p. 14)

Yaari, E., Baruchson-Arbib, S., & Bar-Ilan, J. (2011). Information quality assessment of community generated content: A user study of wikipedia. *Journal of Information Science*, *37*(5), 487–498. https://doi.org/10.1177/0165551511416065 (cit. on pp. 10–14, 39)

Zhang, S., Hu, Z., Zhang, C., & Yu, K. (2018). History-based article quality assessment on wikipedia. *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 1–8. https://doi.org/10.1109/bigcomp.2018.00010 (cit. on p. 5)