

IICON: IDENTIFYING INFORMATIVE COMMENTS IN ONLINE NEWS

A Thesis presented to
the Faculty of the Graduate School
at the University of Missouri

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
ABDULLAH AL MARUF
Dr. KSM Tozammel Hossain, Thesis Supervisor
DECEMBER 2023

The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled:

IICON: IDENTIFYING INFORMATIVE COMMENTS
IN ONLINE NEWS

presented by Abdullah Al Maruf, a candidate for the degree of Master of Science and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. KSM Tozammel Hossain

Dr. Grant Scott

Dr. Abu Saleh Mohammad Mosa

DEDICATION

I dedicate this thesis to my parents, who have been a constant source of inspiration, support, and encouragement throughout my academic journey. I also extend my heartfelt gratitude to my mentors, friends, and classmates for their valuable advice and unwavering encouragement, which played a crucial role in helping me complete this thesis.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor Dr. KSM Tozammel Hossain for his invaluable guidance, unwavering support, and continuous encouragement throughout the entire research process. His insightful feedback and constructive criticism have challenged me to think critically and push the boundaries of my research. They have generously shared their knowledge and provided valuable resources that have significantly enriched the quality of this work.

I am truly grateful for the mentorship that he has provided which extended beyond academic matters, and for being an exceptional role model, inspiring me to reach for excellence in both my academic and personal pursuits. I would like to extend my sincere appreciation to him for believing in my abilities and supporting me every step of the way. His guidance and encouragement have been a driving force behind the successful completion of this thesis. I would like to express my gratitude towards Dr. Grant Scott and Dr. Abu Mosa for their interest to be a part of my thesis committee.

I consider myself incredibly fortunate to have Dr. KSM Tozammel Hossain as my thesis advisor, and will always cherish the knowledge and skills I have gained under his mentorship. Thank you for being an outstanding advisor and for being a significant part of my academic journey.

Finally, I am also deeply thankful to my parents for their constant inspiration and encouragement, and to my friends for their unwavering support during this journey. Your belief in me has been instrumental in reaching this milestone.

Abdullah Al Maruf

Contents

DEDICATION	ii
ACKNOWLEDGMENTS	iii
LIST OF TABLES	vi
LIST OF FIGURES	vii
ABSTRACT	viii
1 Introduction	1
1.1 Informative Comment	2
1.2 Goal	2
1.2.1 Aim 1: Developing a Data Integration Framework for Col- lecting Data from Various Sources	3
1.2.2 Aim 2: Identifying Informative Comments in Online News	3
1.3 Framework Overview	3
1.4 Research workflow	5
1.5 Contribution	5
1.6 Report Organization	6
2 Related Works	7
3 Dataset and Preprocessing	9
3.1 Data Sources	9
3.2 Data Collection	10
3.2.1 Crawler	11
3.2.2 Data Processor	13
3.2.3 Comment Crawler	13
3.2.4 Comment Curation	14
3.2.5 Database	15

3.2.6	Full System Integration	15
3.2.7	Dataset details	16
3.2.8	Characterising Dataset with Histograms	17
3.2.9	Data Curation	19
4	Methodology	21
4.1	DLM: DAG-based Language Model	21
4.1.1	Model Definition	21
4.2	CLM: Context Learning Model	25
4.2.1	Training CLM	26
4.3	Building the Training Dataset	27
4.3.1	Creating the Test Dataset	28
5	Experiments and Results	29
5.1	Experimental Setup	29
5.2	The Baseline Models	30
5.3	Method Comparison with Informativeness	30
5.4	Method Comparison with Editor Picks	31
5.5	Qualitative Analysis of Informative Comments	31
5.6	Ablation Study	33
6	Discussions	35
6.1	Limitations	35
6.2	Future Work	36
6.3	Ethics Statement	37

List of Tables

Table	Page
5.1 Mean precision(MP)(%) for each set of recommended K comments for all of the models. For top- K comments, CLM outperforms the other methods. MP drops with the increase of K	30
5.2 Mean precision(MP)(%) of editor picks for each set of recommended K comments for all of the models.	31
5.3 Mean Precision@K for informativeness score of 4,5,6,7.	34

List of Figures

Figure	Page
1.1 Informative Comment Identification Framework. This framework has two major components: a) news article extraction and b) informative comment identification framework.	4
3.1 A distribution of comment counts in the database. The majority of the news has on average 20-30 comments.	10
3.2 Article and Comment Crawler Architecture.	16
3.3 The histogram of the number of comments per article.	18
3.4 The number of comments per article after the last editor pick is published.	18
3.5 The number of comments and user responses after the last editor pick is published.	19
3.6 The comments activity in time after the last editor pick is published.	20
3.7 The time gap between article publish time and the last editor pick published time.	20
4.1 DLM architecture. This model builds layer by layer. New words that co-occur with existing words in the same sentence create a new layer.	22
4.2 Context Learning Model.	26

ABSTRACT

Many news outlets are discontinuing their comment sections due to moderation challenges as manual moderation for identifying irrelevant and informative comments is inadequate, costly, and time-consuming. Recognizing informative comments serve as an endorsement, giving both the comments and the news more credibility, engaging more readers, and shaping the discourse around the news article. An alternative to manual moderation is automating comment curation using data-driven methods, which reduce human moderation effort as an assistive tool. Most of these methods are based on term matching, such as TF-IDF or BM25, which do not adequately address the issue of identifying comments that do not significantly share terms with the article but are relevant to its context. This paper presents a framework, IICON, specifically designed for online news that takes a news article and the associated user comments as input and determines the most insightful comments. We develop sparse and dense retrieval models to work within IICON. To evaluate these methods, we create a training corpus of 18K news articles having 1M user comments from the Guardian. We also create an expert annotated testbed benchmark for which experiments show that IICON based on dense and sparse retrieval models performs competitively and outperforms existing methods by 2% to 8% in the mean precision.

Chapter 1

Introduction

User engagement with news articles is critical to participatory journalism [1]. Studies show civic engagement helps ensure transparency in news reporting [2], news outlets' self-assessments [3, 4], and story selections [5]. User interactions with news articles also capture mass opinion on various topics.

Approximately 80% of US news readers have read news comments at some point [6]. Despite this large readership of comments, many news outlets are terminating the comment section, as moderating an online news comments section is quite challenging for dealing with trolling, offensive, and inappropriate comments. Although news outlets employ manual moderation with domain experts, it is still inadequate for several reasons:

- Managing the moderation of user comments is getting more and more challenging as the number of users and comments grows over time which turns out to be expensive over time.
- It is hard to update and moderate the section, and users continuously make comments.
- Doing human moderation can be biased.
- The nature/rate of the comments is unpredictable. Hard to moderate if the

moderator doesn't have the full context of the topics the user is discussing [7]

1.1 Informative Comment

An increasing interest is in developing automated comment curation methods that can be used as standalone or assistive tools [8]. Two editorial criteria are usually employed for comment curation [8]:

- Negative criteria to exclude comments, such as ad hominem attacks, profanity, or other abusive behaviors.
- Positive criteria for identifying comments worthy of reading, extending information on the article in a similar context and topics, and possibly responding to other comments.

In this paper, we focus on identifying informative comments which are related to articles and provide additional information in coherence with the context of the article [9]. Those comments, which contain not only the information from the corresponding article, will also contain extra information. These comments can be considered an extension of the original article.

1.2 Goal

The thesis aims to develop a framework that can autonomously assess user comments in an online news platform to identify the comments that provide valuable information. Additionally, the thesis involves constructing a data collection module to gather and organize data for the framework's models.

1.2.1 Aim 1: Developing a Data Integration Framework for Collecting Data from Various Sources

We need a data source that contains publicly accessible user comments that can be collected. Additionally, we need some reference data that includes informative comments such as editor picks and user up-votes/likes. Furthermore, as the writing style and structure of articles and comments evolve over time, we developed a versatile crawler to consistently gather up-to-date data from the source. This data is then pre-processed, cleaned, and stored for further use. It is continuously fed into the models to ensure that the framework’s models are constantly updated.

1.2.2 Aim 2: Identifying Informative Comments in Online News

The predominant methods for identifying informative comments are based on term matching between comments and articles. However, these term-based methods, such as TF-IDF, BM25, and DLITE [10, 11], have limitations: a) these methods may not capture context, b) they are sparse methods, and c) depends on term matching. Recent advancements in deep-learning-based NLP methods, such as transformers, are well-known for capturing context in text analyses. Although these dense-retriever methods are applied to capture helpful reviews in different domains, to the best of our knowledge, they are not adapted to identify informative comments in online news. In this paper, we study the problem of identifying informative comments in an online news article by estimating the context similarity between comments and the article.

1.3 Framework Overview

We present a framework—IICON—Identifying Informative Comments in Online News for informative user comments identification. This framework is pluggable with sparse and dense retriever models (Fig. 1.1). The Data Ingestion and Pre-

processing module fetches news articles and associated user comments from a news site, performs basic preprocessing, and loads them into a document-oriented database. The Informative Comment Extraction module fetches relevant datasets from the database and transforms the article text into features required by the Predictive Modeling module. Within the Predictive Modeling submodule, we implement two methods for detecting informative comments: a) DLM: DAG-language model and b) CLM: Context-learning model. Intuitively, DLM is a probabilistic, sparse retriever and enhances a term-based retriever method. This method captures the context of the article with two sets of term vectors; a comment is matched against these two vectors to assess its informativeness. The other method CLM is a fine-tuned bi-encoder-based transformer that captures the context of an article and its comments and identifies informative comments. Both methods can continuously update the informative comments upon getting new user comments.

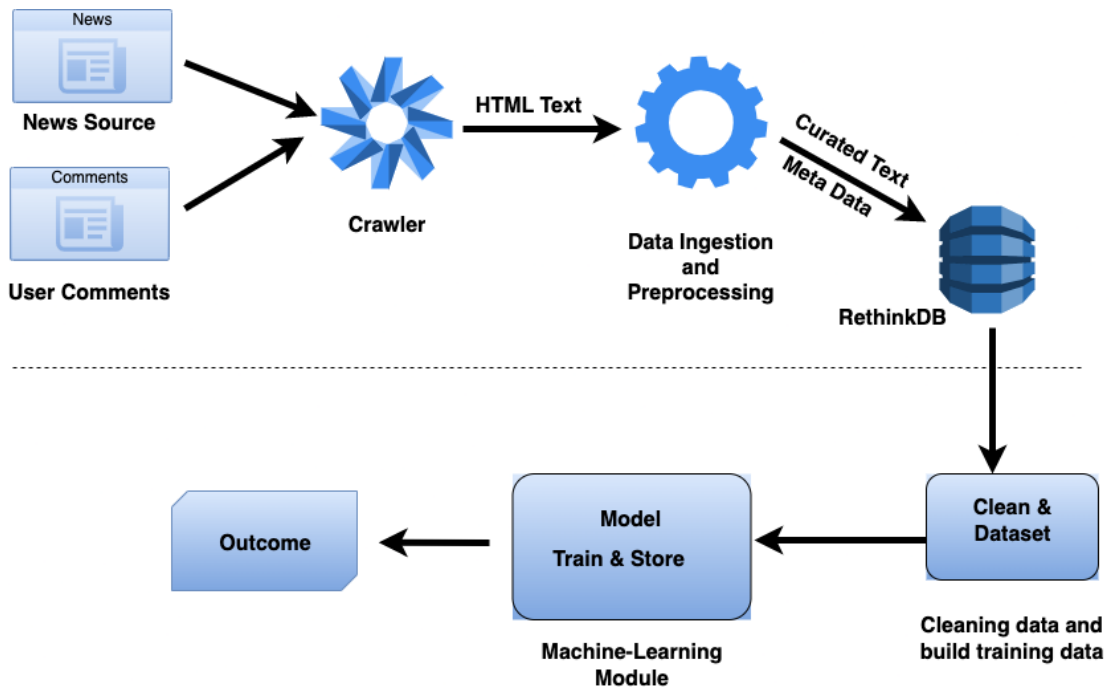


Figure 1.1: **Informative Comment Identification Framework.** This framework has two major components: a) news article extraction and b) informative comment identification framework.

1.4 Research workflow

The research workflow is as follows:

- **Data Collection:** Gather data from various sources and pre-process it, including normalization, to prepare it for training and testing the models.
- **Fine-tuning the Model:** Utilize a pre-trained model and fine-tune it using the processed data to enhance the accuracy and performance of the model. For DLM, create a token corpus.
- **Building Test Dataset:** Develop an annotation guide and have domain experts annotate the data to create a test dataset.
- **Testing and Evaluation:** Test the model's output using the annotated data and build evaluation metrics to measure the model's performance. This evaluation will help determine how much additional training may be needed.

By following this workflow, the research aims to enhance the accuracy of the models, evaluate their performance, and ultimately achieve the desired objectives of the study.

1.5 Contribution

Our key contributions are as follows:

- Constructing a versatile data collection module that functions as an independent service, capable of automatically gathering data from various sources without needing constant human intervention. This system is designed to be scalable and resilient, capable of recovering from failures or disruptions.
- We propose a framework for informative comment identification in online news with two pluggable NLP-based methods, which focus on context matching for selecting informative comments and alleviate the issues related to term-based methods.

- The proposed methods are scalable concerning different numbers of comments and handle comments without any manual feature tuning.
- We demonstrate the efficacy of our methods by conducting extensive experiments on news articles and comments fetched from the British daily newspaper, The Guardian. The results suggest that the proposed methods produce better informative user comments and editors' picks compared to the baselines.

1.6 Report Organization

The report is structured into 5 sections as follows:

- **Introduction:** This chapter presents the problem, discusses the relevant domains, outlines the objectives, and proposes a solution for the problem.
- **Related Works:** The second chapter provides information on existing related works and key research in a similar domain.
- **Datasets and Processing:** The third chapter delves into the datasets used and explains the process of obtaining and preprocessing them.
- **Proposed Models:** The fourth chapter details the proposed models and their specifications.
- **Results and Analysis:** In the fifth chapter, quantitative results are discussed, and a comparison with baseline models is provided. Additionally, qualitative result analysis is presented.
- **Discussion:** The final chapter covers the pros and cons of both models, their limitations, and future work possibilities, and includes an ethics statement.

This structure will help readers to navigate through the report and gain insights into the problem, methodology, results, and conclusions of the research.

Chapter 2

Related Works

Most of the existing work on finding relevant and informative passages or comments is based on term-matching-based models, such as TF-IDF, BM25[10], SVM[12] and DLITE [11]. These models match terms to find the correlation between the articles and comments. But their dependency on having matching words affects their performance. Recent research uses a variety of terms for SEO optimization[13], which requires semantic analysis to discover the matched terms; term-based models cannot perform such semantic analysis to find relevant information. There are other machine-learning-based methods that extract features from articles and comments using phrase matching and perform classification [8, 9]. These methods do not capture semantics while annotating the comments.

Our algorithm for generating language models from an article parallels the topic modeling of documents. Titov and McDonald propose MG-LDA [14, 15] for modeling text's local topics that scatter across the corpus. As every segment may not be related to a comment, MG-LDA is not appropriate for the proposed problem. Corr-LDA [15] is a topic model for correspondence. As Corr-LDA works with a single vector model, a specific comment on a small segment of an article can show a small correlation with the article using this model. To overcome this limitation, Das et al. [16] developed a correspondence topic model (SCTM) that uses multiple topic vectors, which allow comments to be matched with more

correspondence segments of the article.

Mat et al. develop a master-slave topic model (MSTM) and an extended master-slave topic model (EXTM) for summarizing comments [17]. In these models, articles are masters, and comments are slaves; comments are clustered based on their topics. Each model identifies representative comments from the comment clusters. The key assumption that a comment is related to a single topic exclusively is a limitation.

Another direction for identifying relevant is to develop classifiers with article segments and comments. E.g., Sil et al. [18] use supervised and unsupervised techniques to create structural classifiers to match comments with news article segments. This work employs explicit semantic analysis and co-reference features to represent the text in the article and shows that the accuracy of discriminative approaches depends largely on effective feature selection.

A recent trend in deep learning is developing models that understand the context of text or passage. These methods are very effective for extracting relations and information from human conversations[19]. Our proposed method, CLM, depends on a dense representation of passages. The state-of-the-art models for learning context in the text are Transformer [20] and Contriver [21].

Transformers have been applied to product review classification [22], rumor detection [23], fake review identification [24], toxic comments classification [25], depression detection [26], comment sentiment analysis [27] and user stance detection in comments [28]. Most of these tasks are cast as classification problems. Some studies rank user comments on social media, e-commerce, and video-sharing platforms using statistical method [29, 30]. To the best of our knowledge, we did not find any deep learning models that learn and exploit the context of a news article to identify the associated informative comments.

Chapter 3

Dataset and Preprocessing

Selecting an appropriate dataset that includes both informative and non-informative public user comments presents a significant challenge. Subsequently, we will develop a system for collecting data from the chosen source. This section of the report addresses the considerations involved in dataset selection, as well as the processes of data collection, curation, and storage.

3.1 Data Sources

The main challenge encountered in this study is identifying a suitable data source that offers a substantial amount of public user comments and articles. Several platforms were explored, including NYTimes, Guardian, and Al-Jazeera, all of which contain public user comments. However, each platform presents its own issues when it comes to data collection.

NYTimes necessitates a user account to access articles, making the crawling process more challenging. Furthermore, their API does not provide the print version of the articles. Another drawback of NYTimes is that the comments section is heavily moderated, requiring approval from moderators before comments are published. While this moderation ensures relevant user comments, it poses difficulties in obtaining non-informative comments for model training. Bias-checking

is also required when using this data.

Al-Jazeera, although containing user comments, does not allow public reading of those comments, making it less suitable for this study. On the other hand, Guardian stands out as the only platform where user comments are openly visible to the public. Additionally, creating an account is only necessary for commenting, while articles are publicly accessible to anyone. The editor picks features of Guardian that can be utilized as relevant comments.

After careful deliberation, Guardian was selected as the preferred data source due to its extensive archive of news data and significant user engagement.

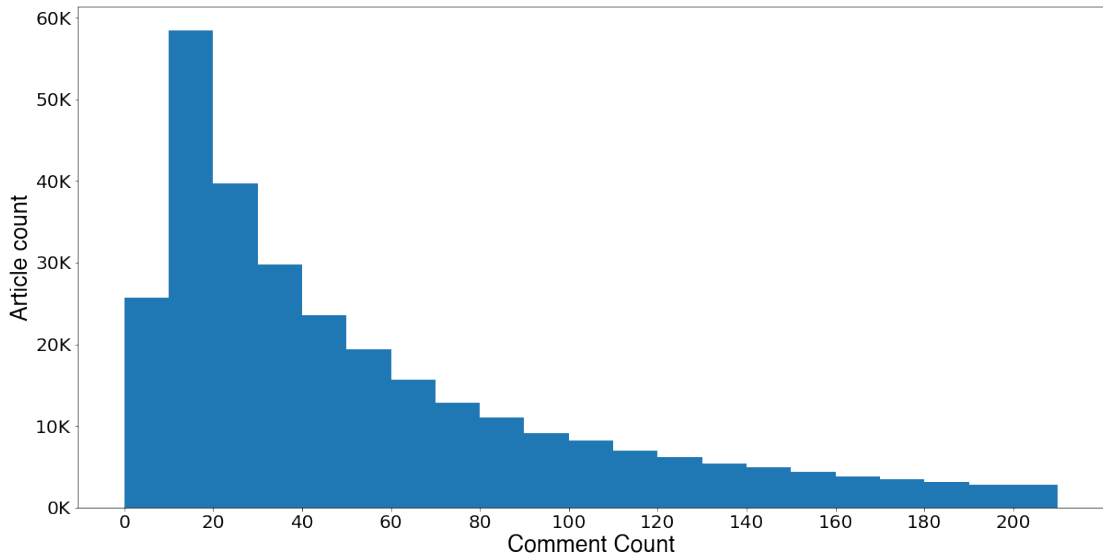


Figure 3.1: A distribution of comment counts in the database. The majority of the news has on average 20-30 comments.

3.2 Data Collection

We develop a crawler framework to extract data from online news sites. This framework has the following functionalities.

- Build an automated data crawler that continuously collects news articles from any news source (see Sec. 3.2.1)
- Build a crawler for collecting user comments from the site (see Sec. 3.2.3)

- NLP processor extracts data from raw web articles and collects news titles, bodies, and metadata. (see Sec. 3.2.2)
- Store this data in a NoSQL db.(see Sec. 3.2.5)

3.2.1 Crawler

We require the development of a persistent crawler that can continuously gather data over time. However, the process of collecting information from various external websites poses challenges. There are multiple factors to consider when constructing the crawler.

- How source render the data?
- Speed of the website?
- If website has cookies or redirection.
- How much concurrent connect site can support?
- Is there any possibility of blocking our IP?
- How to recover from an interruption in the site or our server network?
- Data duplication removal, or tacking which is collected or not?
- How much depth we can crawl and needed?

Initially, we primarily selected websites that render data on the server side, simplifying the process of retrieving the entire rendered page from the HTML. For other websites, where data rendering requires a separate headless module, we utilized the Python Scrapy library as the foundation for our crawler.

The crawler was configured to establish only one connection to any given source, mimicking human user behavior. This approach not only reduced the risk of being blocked by the data source but also enabled the crawler to check for redirection and invalid HTML (e.g., 404 status).

The crawler module specifically focuses on collecting HTML from each link originating from the provided source page. This module is designed as a service, which can be triggered by a scheduler with the appropriate source and configuration. This setup facilitates scalability as we can simply add the source configuration to the scheduler, which will then initiate the crawler service in parallel processes at fixed intervals.

Our scheduler process triggers the crawler every 30 minutes for each source, starting from a specific page. The crawler then proceeds to crawl each link until the configured depth is reached. Since online news platforms typically feature new articles on their homepage, our 30-minute interval allows us to capture most of the recent articles. However, to ensure we don't miss older articles, we initially run the crawler with a larger depth of 10.

As online news pages often contain numerous links that redirect to different sites, the crawler includes filtering mechanisms to ensure we collect valid data. Scrapy, the library we use, has a built-in system for checking URL duplications in each run. To address global duplication, we assign a unique ID to each document generated from the URL hash. This ID is stored along with the URL, guaranteeing that we only store unique URLs from a single source.

By implementing these measures, we aim to capture both new and older articles while filtering out irrelevant links and preventing duplicate data storage.

Both the crawler and scheduler operate as services, allowing for scalability as the number of sources increases. This architecture enables easy expansion to handle a larger volume of sources without significant modifications to the system. As the demand for crawling multiple sources grows, the service can be scaled up by adding additional resources such as computing power and storage capacity. This ensures that the crawler and scheduler can effectively handle the increased workload and maintain efficient data collection from a growing number of sources.

3.2.2 Data Processor

The current setup involves the crawler collecting raw HTML from the sources. However, to obtain the necessary article data, including the title, content, and metadata such as dates and writers, a generic HTML processor library is utilized. This library is designed for extracting news information from raw HTML.

To address the issue of potentially collecting invalid data, a validation step is introduced to determine if a link is valid or not. News sources typically follow a pattern in their published article links, and this pattern is used to validate links before adding them to the scheduler configuration. By ensuring only valid links are included, the processor can accurately extract article information.

To streamline the processing of data and avoid bottlenecks, each crawler service is equipped with its own HTML processor. This allows each crawler to independently process its data without waiting for a central service, ensuring efficient and timely data extraction.

Overall, this approach optimizes the data collection process, validates links before processing, and enables parallel processing by utilizing individual HTML processors for each crawler service.

Based on the assumption that articles are typically written and published by professional writers following a standardized format, the process of text normalization has been skipped. This decision is based on the expectation that the text in the articles already adheres to a consistent and well-structured format.

3.2.3 Comment Crawler

To collect user comments from various news sources, we encountered the need to build separate crawlers and parsers due to the different setups and platforms used by each source. Each crawler is designed to be source-specific unless multiple sources utilize the same platform. In our case, we focused on collecting user comments specifically from the Guardian platform, resulting in the development

of a comment crawler dedicated to this source.

User comments, being publicly available, lack a standardized structure. The data lacks formatting and can include various Unicode characters. To normalize the data, we created a normalization tool that removes different types of Unicode characters and converts different quotation styles to a unified format for consistency and formatting purposes. This normalization module operates as a standalone service responsible for collecting user comments from sources independently, separate from the generic article crawler. After collecting the data, the normalized comments are stored along with the corresponding articles in the database.

By implementing separate crawlers and a normalization tool, we can efficiently collect and organize user comments from different news sources, ensuring data consistency and facilitating further analysis.

3.2.4 Comment Curation

The data crawler framework includes a parser, but further text curation is necessary due to the open nature of the commenting platform. Users can write comments with various unwanted features, such as Unicode characters, URLs, emails, and user taggings, along with multiple types of quotations and punctuation marks.

To address this, a text curation module is employed, which eliminates unwanted punctuation marks and normalizes the text to a uniform format. The module also removes email addresses and user taggings from the comment body. In cases where a comment refers to another article via a URL that contains the title of the referred article, the normalizer replaces the URL with extracted keywords from the URL.

This curation process ensures that the comments are cleaned and standardized, making them more suitable for analysis and natural language processing tasks.

3.2.5 Database

Each article is uniquely identified by hashing its URL, establishing a key-value structure for our system. Given the increasing number of sources in the crawler, our system’s write load becomes significant. To ensure scalability, we have opted for a NoSQL database over a structured database. Specifically, we utilize RethinkDB as our data store, which offers clustering capabilities. This allows our system to distribute the workload and handle increased data volume efficiently.

RethinkDB offers a built-in user interface, enabling easy data exploration and providing useful statistics. To enhance scalability further, we utilize separate tables for different sources. This approach facilitates efficient data searching based on the source, enhancing query performance and streamlining data retrieval.

To ensure data durability and protection against system failures, we have implemented a backup system. Regular backups of the database are taken, allowing us to recover old data if needed in the event of a system failure.

By leveraging a NoSQL database like RethinkDB, employing separate tables for different sources, and implementing backup mechanisms, we have created a scalable and robust data storage solution that supports the growing demands of our crawler system.

3.2.6 Full System Integration

The entire system, including the crawler, parser, normalizer, scheduler, and database, is containerized using Docker, facilitating easy deployment on any system. The system is organized using Docker Compose, making it effortless for anyone to copy the data to a new system and initiate the system there.

The framework operates continuously and has successfully gathered approximately 650,000 articles along with around 58 million user comments contributed by approximately 1.8 million unique users. It is essential to emphasize that the crawler framework does not extract user identities; instead, it solely utilizes sur-

rogate IDs provided by the Guardian for analysis purposes.

The crawler has also been effectively configured to collect data from approximately 20 more news sources, resulting in over 70,000 collected articles from MO (assuming "MO" refers to a specific news source or category). This ongoing expansion ensures the inclusion of a diverse range of data from various sources for further analysis and insights.

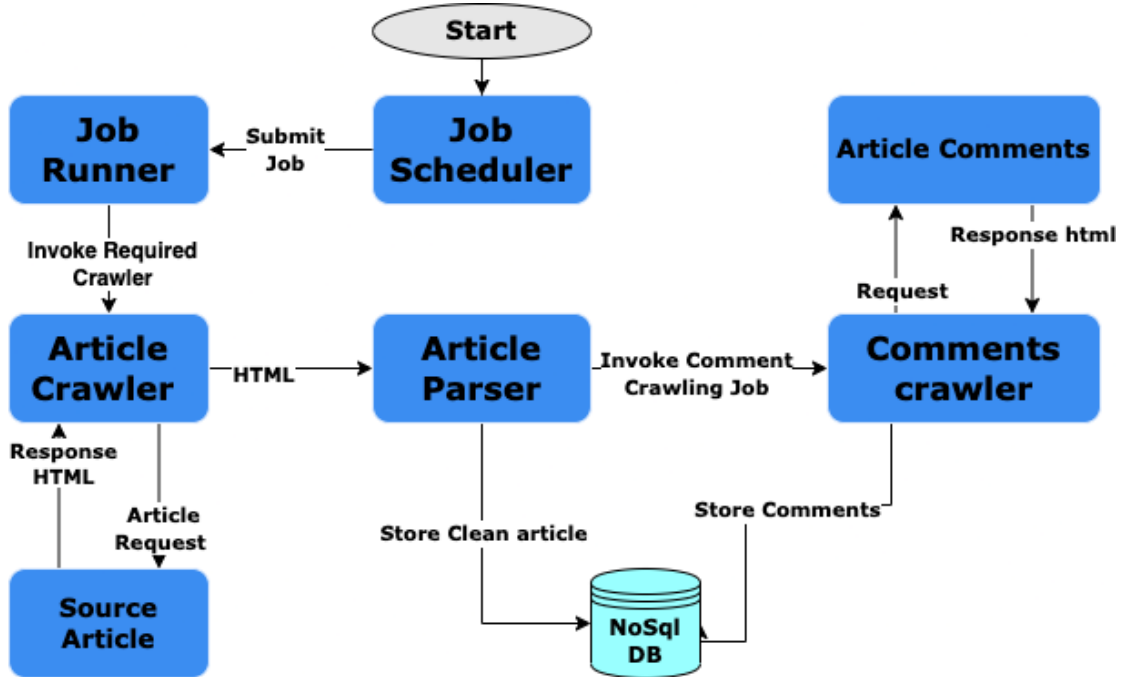


Figure 3.2: Article and Comment Crawler Architecture.

3.2.7 Dataset details

Within the first three months of operation, our crawler successfully collected 650,000 news articles from the Guardian. These articles span a time range from 2009 to 2022. The crawler framework is designed to operate continuously, allowing it to gather data from the Guardian as new articles become available on the platform.

The average size of the articles in our dataset is around 400 words, while the average size of the user comments is approximately 50 words. Our current processed dataset consists of over 58 million user comments contributed by more

than 1.8 million unique users.

For training our model, we carefully selected approximately 17,000 news articles from the dataset. These chosen articles contain approximately 1 million user comments. To ensure meaningful training data, we specifically opted for articles that have between 2-5 editor picks and a minimum of 50 user comments. This selection criterion aims to facilitate the model’s understanding of the context within a passage, as it can be challenging to train the model effectively without sufficient relevant comments and editorial selections.

3.2.8 Characterising Dataset with Histograms

In our exploration of the dataset, we utilize various histograms to examine different characteristics, particularly focusing on the timing of comments and editor picks. Fig.7-10 demonstrates that even after the last editors’ picks are published, there is a significant number of user interactions. This suggests the presence of informative comments even beyond the time of finalizing editors’ picks. This observation serves as a key motivation behind the proposal of IICON, an automation tool for editors’ picks.

Regarding the count of user comments, the majority of articles have approximately 20-40 user comments. However, for particularly interesting topics, a substantial number of articles have over 100 user comments. This abundance of comments can make it challenging for new users to read through all the comments. Fig. 3.3 provides a histogram illustrating the distribution of comments count in articles.

Figures 3.4 and 3.5 highlight an interesting observation that, despite hundreds of user comments, most sources provide users with editor picks to guide their reading. Surprisingly, even after the editor picks are published, a significant number of user comments continue to be posted in the articles. The irregular nature of the comments’ timing makes real-time editor picks challenging.

Fortunately, our framework can address this issue effectively. By automating

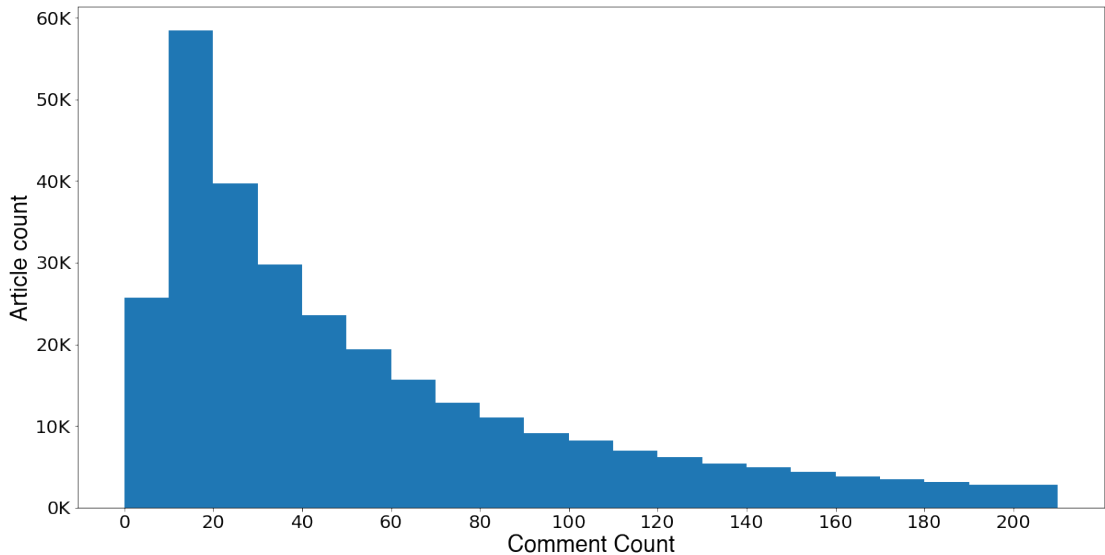


Figure 3.3: The histogram of the number of comments per article.

the process of editor picks with IICON, we can ensure that informative user comments are not overlooked, even after the initial editor picks have been made. This allows for a more comprehensive and inclusive approach to presenting valuable content to readers, enhancing the overall user experience.

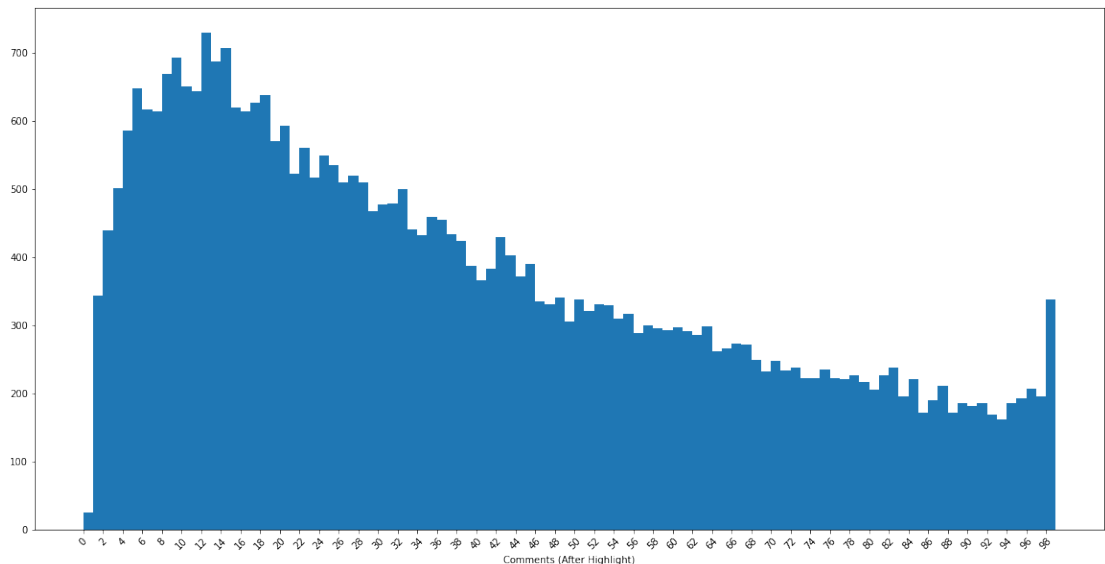


Figure 3.4: The number of comments per article after the last editor pick is published.

Indeed, we observe that even though there is a substantial amount of user activity after the last editor picks are published, there is often a significant time delay before those comments are pushed. In many instances, this time delay

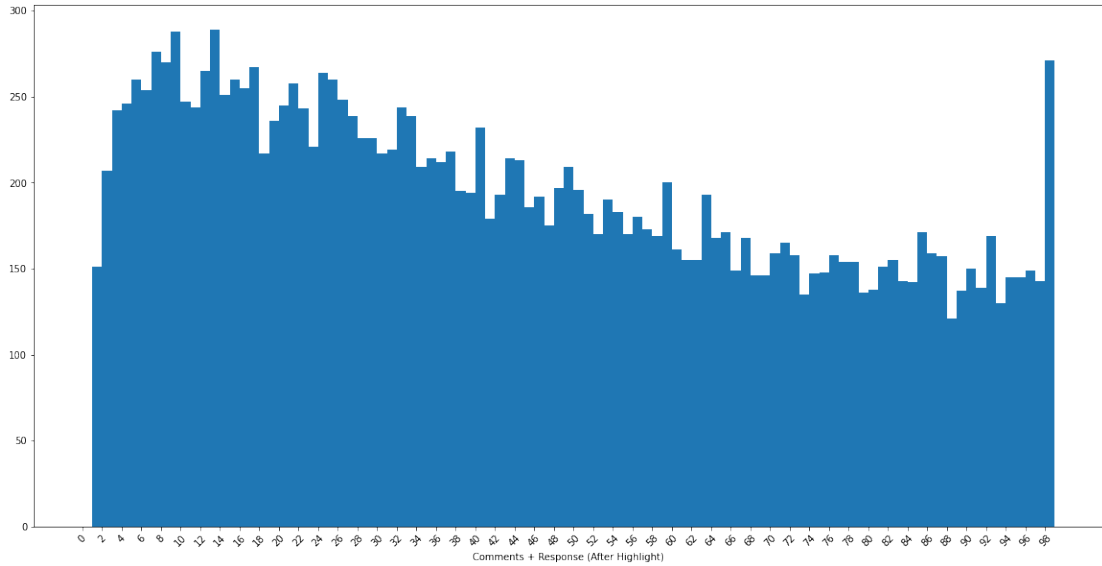


Figure 3.5: The number of comments and user responses after the last editor pick is published.

exceeds 48 hours. Additionally, the editor picks themselves are not immediately published after the article is published; there is an average time gap of 24 hours.

This time gap between user activity and editor picks’ publication can result in valuable comments being missed by readers who primarily focus on the initial picks. However, our proposed IICON framework can bridge this gap by automating the process of editor picks, ensuring that informative comments are highlighted in a timely manner. This way, readers can have access to valuable insights, even if they are posted after the initial publication or editor picks.

3.2.9 Data Curation

A text curation module drops unwanted punctuation marks and normalizes them to a uniform format. The process discards any email address and user tagging from the comment body. Sometimes a comment refers to another article via URL; this URL may contain the title of the referred article. While processing, the normalizer replaces the URL with the extracted keywords from the URL.

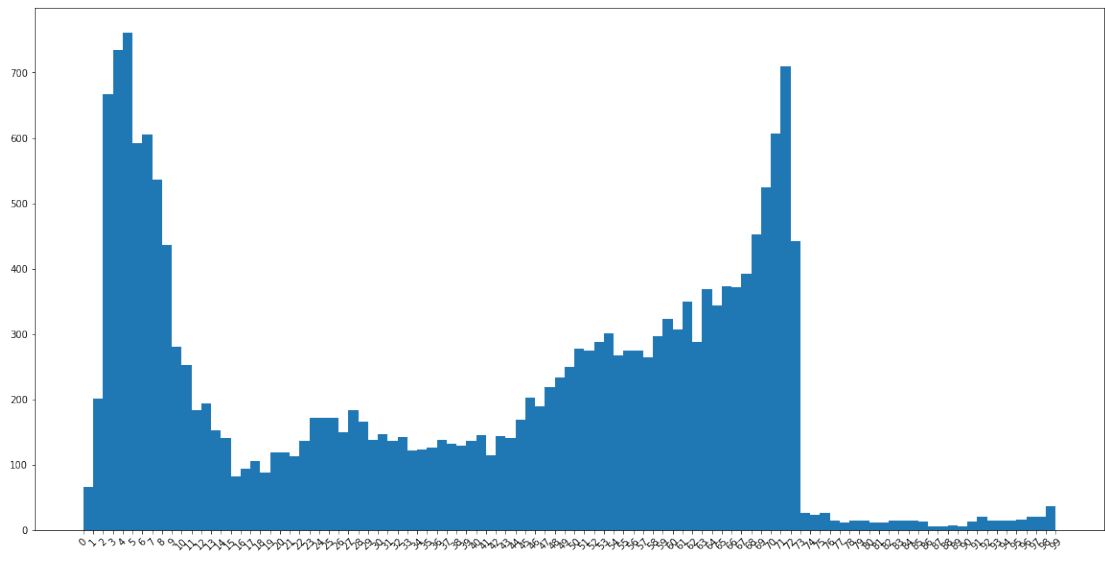


Figure 3.6: The comments activity in time after the last editor pick is published.

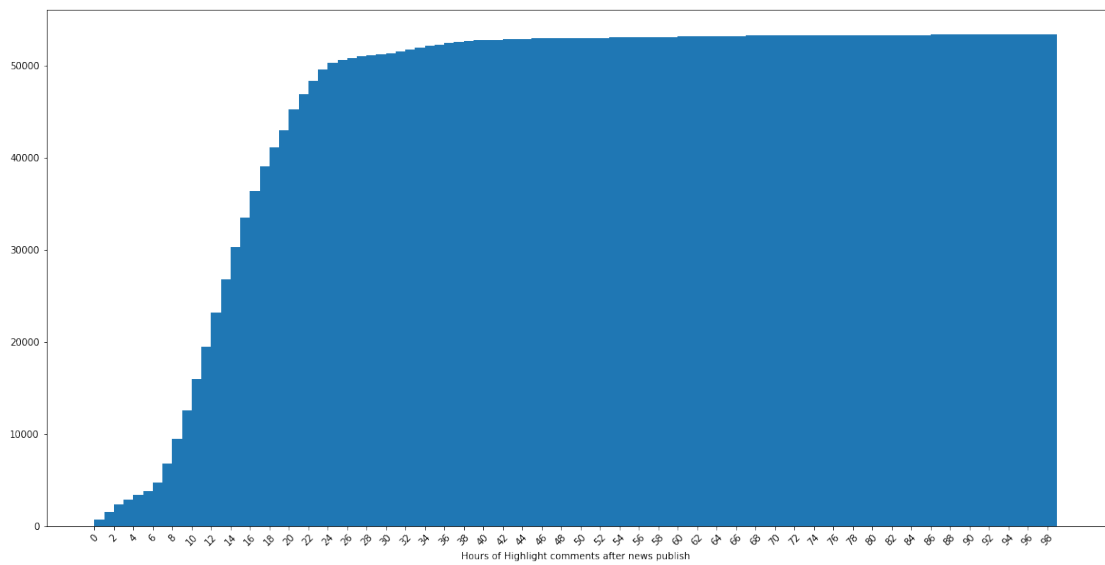


Figure 3.7: The time gap between article publish time and the last editor pick published time.

Chapter 4

Methodology

This section describes our two proposed methods: a) DLM: DAG-Language Model and b) CLM: Context-Learning Model.

4.1 DLM: DAG-based Language Model

DLM treats the informative comments identification task as a language model matching problem. The key idea is similar to the one proposed by [31] with the important difference that we use the news article to generate a query language model, then generate queries to perform a probabilistic ranking of the comments. There are three key steps: a) Construct a query language model from the news article, b) Formulate multiple queries from the language model, and c) Rank the comments against generated queries to determine comments' informativeness. Here, the terms in the corpus of all comments and articles contribute to smooth the query language model.

4.1.1 Model Definition

We weigh the terms of a document according to their importance using the inverted pyramid model [32] [33] [34]. To capture readers' attention, the most important

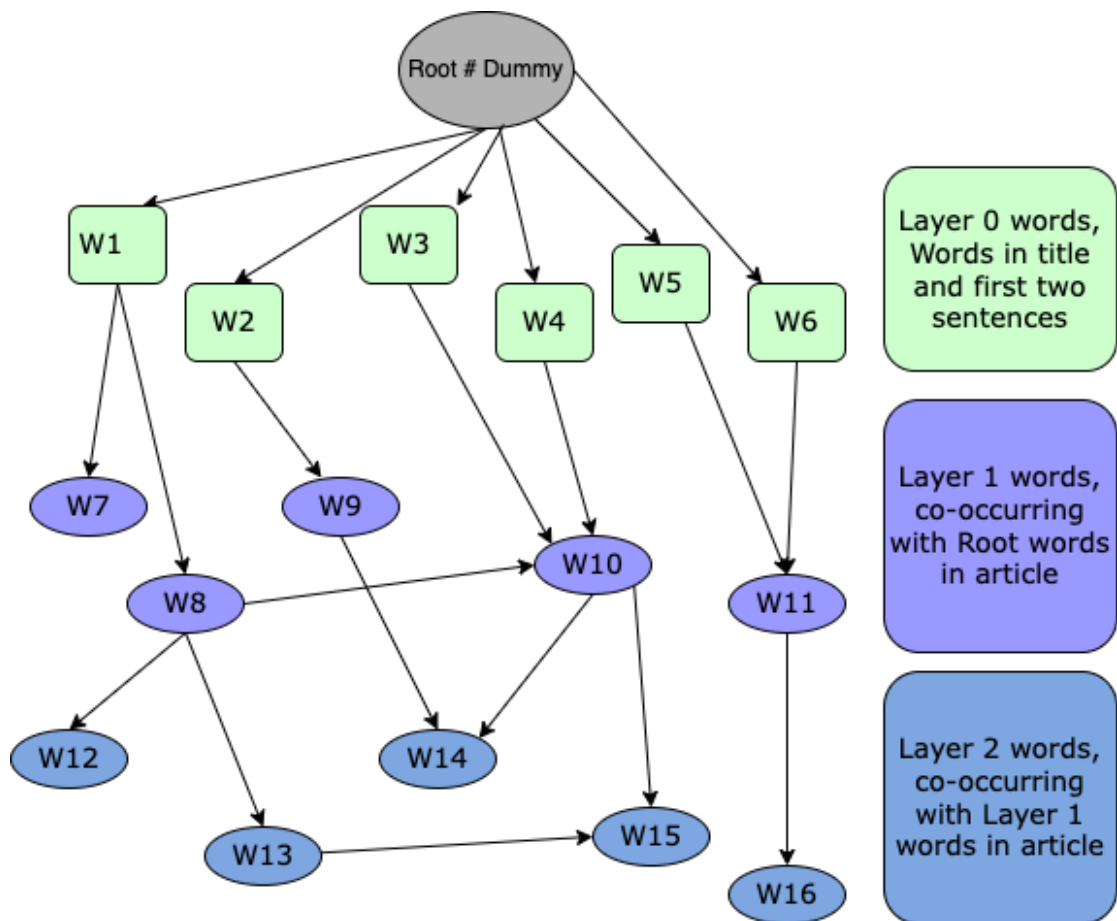


Figure 4.1: DLM architecture. This model builds layer by layer. New words that co-occur with existing words in the same sentence create a new layer.

information related to a story should be presented in the leading paragraph of an article; the subsequent paragraphs should present detail in decreasing order of importance. The title is typically a one-line summary of the article. In our observation, the title and the first few lead sentences are sufficient to capture a story's core. We refer to them as the *lead* and the remaining part of the article as the *context*. The importance of a term for the query model is determined by four parameters:

- Term Frequency.
- Significance level: the minimum distance of a term from the terms occurring in the lead
- Relevance weight: the number of co-occurrences of a term with other significant terms in the article

- Smoothing Factor: the general likelihood of the term’s appearance within the corpus

Let $\mathbf{A} = (A, D_1, D_2, \dots, D_n)$ be a news article, where A is the article text and all D_i are associated comments. Let $\mathcal{C} = \{\mathbf{A}_1, \mathbf{A}_1, \dots, \mathbf{A}_n\}$ be the corpus. Given an \mathbf{A}_i , we aim to identify informative comments in \mathbf{A}_i .

To calculate both the significance and relevance of terms, we construct a directed acyclic graph (DAG), $G = (V, E)$, for an article A where the vertex set is defined as $V = w_i | w_i \in A$; and the edge set as, $E = \{(w_i, w_j) | w_i, w_j \in s_k, s_k \in A\}$. Here w_i is a word in the article, and s_k is a sentence. Before building G , we curate the terms in A by removing stop words and stemming. The DAG has a dummy node as a root. The nodes initially connect to the root are the terms of the title and two lead sentences. For each of the rest of the sentences, if there is any term(s) in this sentence that is already included in the DAG, we create nodes for the other terms in the sentences and connect them with the co-occurred term in the DAG. We keep expanding the graph until there is no new node.

We define the significance level of a word w_i as follows. For $w_i \in lead$, $\sigma_{w_i} = 1$, and for $w_i \notin lead$, $\sigma_{w_i} = 1 + d(\min(w_i, w_j) | w_j \in lead)$. Here the distance is measured as the number of edges between the term pair. The relevance weight of a term w_i is measured as follows:

$$\rho_{w_i} = \sum_{\substack{(w_i, w_j) \in E \\ s_j < s_i}} \max(0, 1 - \log(s_j)) \quad (4.1)$$

From the definition of σ and ρ , the terms that occur in close proximity to words in the lead have more significance than terms that appeared further. Terms that occur frequently with more significant terms have more relevance weight than terms that occurred less often. Consequently, terms occurring in different sentences have varying degrees of importance in the model.

In the definition of σ , only the term with lesser significance, w_i , gets the benefit of co-occurrence – not the term with greater significance, w_j . The frequency weight

of individual terms with the following formula balances this apparent lacking.

$$fw_i = tf \times \frac{s_i + k}{s_i \times (k + 1)} \quad (4.2)$$

Here k is a scaling parameter, which exponentially decreases the importance of a term's frequency (tf) with the increase in its significance. The likelihood of a term contributing to a query is a mixture of normalized relevance and frequency weights:

$$l_{w_i} = \alpha \times \rho_{w_i}^{norm} + (1 - \alpha) \times f_{w_i}^{norm} \quad (4.3)$$

Equation 4.3 is combined with a collection probability term $p(w_i, C)$ using the Jelinek-Mercer interpolated smoothing method [35] to estimate smoothed probability as follows,

$$p_q(w_i) = (1 - \lambda) \times l_{w_i} + \lambda \times p(w_i, C) \quad (4.4)$$

We expect an informative comment to augment the discussion of the article with supporting or alternative arguments. Given a cut-off threshold for all $p_q(w_i)$, we partition the terms into a principal θ_p and auxiliary terms θ_a . Here, θ_p contains the terms that capture the subject matter of the article, and θ_a provides elaboration on the subject matter. An informative comment should have significant overlaps with θ_p as well as with θ_a . The informativeness (I) of a comment D concerning the article is a harmonic mean as follows:

$$I(P, D) = \frac{1}{\frac{\beta}{sim(\theta_p, \theta_d)} - \frac{1-\beta}{sim(\theta_a, \theta_d)}} \quad (4.5)$$

Here, θ_d is the language model for the comment. Given the modeling of the comment is straightforward, unlike that of the article, a simplifying alternative to language model matching is generating a principal and an auxiliary query from the query language model and doing a probabilistic matching of comments against those queries. We implemented this simplified approach for experimental studies presented in this paper.

Note that auxiliary terms can be numerous. To avoid an abundance of auxiliary terms, we disregard terms that are located more than a certain distance from the root. Using these two queries, we then calculate the BM25 [10] scores for each reader’s comment and then rank all comments based on the combined score. Here, the score against the auxiliary query is inverted to reflect that we seek dissimilarity – not similarity – for the second query before adding it to the score against the principal query.

4.2 CLM: Context Learning Model

As DLM is based on term significance and relevance, it is expected to perform well when a reader’s comment has matching terms with the corresponding article. However, a highly informative comment may be lexically different but semantically similar to an article. Hence, there may not be significant overlapping in the principal terms, which could cause DLM to miss such comments. We propose CLM, a fine-tuned transformer-based bi-encoder, to address this issue. The key idea is to drop the dependency on principal and auxiliary terms so that the model can learn comments with significant terms’ match as well as comments that have a similar context to the article [36] [37]. We represent a passage using a dense representation so that passages with a similar context have the nearest representations in dense vector space [38].

We train CLM based on similar informative and context-matched passages. The model has two encoders, one of them encodes the article ($ENC_a(A)$) and the other encodes the user comment ($ENC_c(D)$). Essentially the model is given two passages p_i and p_j as input, and both orders, $\langle p_i, p_j \rangle$ and $\langle p_j, p_i \rangle$ make sense in the discussion context. The similarity between the encoders is calculated using a similarity function, $sim(A, C) = ENC_a(A)^T * ENC_c(D)$.

We need a decomposable distance function in a large network of multiple cross-attention layers. Due to wide usage and its simplicity We choose the inner product

similarity for our encoder training and inference.

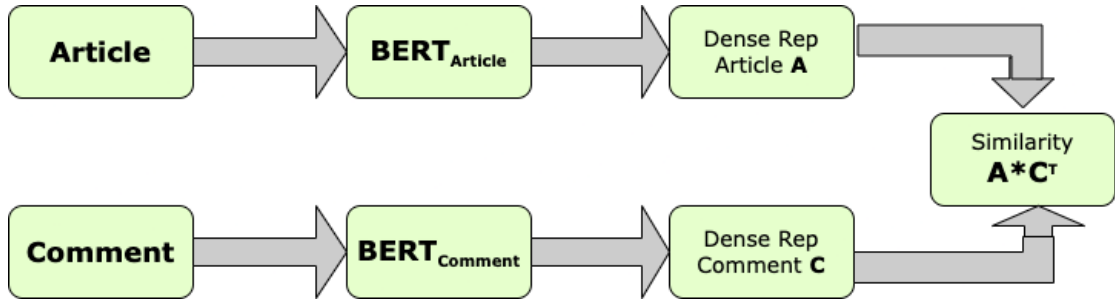


Figure 4.2: Context Learning Model.

4.2.1 Training CLM

We aim to build a dense vector space for each text passage, article, and comment so that texts with a similar context are near in vector space and texts with dissimilar contexts remain at a larger distance.

Given \mathcal{C} (see the definition in Sec. 4.1.1), we build positive and negative examples by pairing articles and comments to train the model.

- Positive instance: Each passage in the pair has similar topics and context.
- Negative instance: Passages in the pair has different topics or context.

Both types of examples ensure that the model learns the similar context passage as well as differentiates between passages with different contexts.

To learn the context of a text, we provide CLM with a pair of continuous texts, where each passage can extend the idea and topics of the other passage. We consider the relevant comments of an article as the positive pairs of the article, as the comment extends the article’s idea with additional information. We also consider each consecutive passage of an article as a positive pair too as each consecutive passage can be a part of a continuous conversation. As for the negative instances, passages from different articles with different topics are used. In addition, relevant comments for different articles pair are used as a negative instance.

For training data, each article has a matched list of comments and an unmatched list of comments. Let the matched list of comments for \mathbf{A}_i be $M_i =$

$\{m_1^i, m_2^i, m_3^i, \dots, m_r^i\} \in \mathcal{C}$ and unmatched comment set $U_i = \{u_1^i, u_2^i, \dots, u_k^i\} \notin M_i$, a subset of comments that comes from other articles. An instance is a triplet $\langle A_i, M_i, U_i \rangle$.

The loss function needs to have the capability to recognize similar contexts and differentiate dissimilar contexts. The key idea is to improve the likelihood of the matched text along with all the unmatched text. So loss function will be optimized for the negative log-likelihood of the positive text.

$$\mathcal{L}(a_i, m_j^{i+}, nm_1^{i-}, \dots, nm_n^{i-}) = -\log \frac{e^{\text{sim}(a_i, m_j^{i+})}}{e^{\text{sim}(a_i, m_j^{i+})} + \sum_{k=1} e^{\text{sim}(a_i, nm_k^{i-})}} \quad (4.6)$$

To increase the number of pairs in the negative set, which helps the model to differentiate between irrelevant comments, we use in-batch negative pairs: for each training batch, the positive comments of each set are considered as negative comments of another set.

4.3 Building the Training Dataset

In the raw dataset, the editors' picks are expert-annotated user comments and can be considered informative and relevant. We apply three strategies to create a training set to fine-tune CLM.

First, We pick 17.5K articles with at least two editors' picks. We pair an article and an editor pick comment to create a positive instance. We also pair the same article with editor picks from other articles to build a negative instance. For these types of instances, we use `jarticle, commentj` and, its inverse, `jcomment, articlej` as distinct instances.

Second, the average number of comments and editor picks in our dataset are ~ 100 and ~ 3 , respectively. As we are using articles from a single source, the editors' picks can have biases. In addition, some comments are based on a single

passage of the article. To improve the context learning from the single passage, we also use passages in an article as a continuous conversation. We build a pair from consecutive passages from the same article as a positive instance. On the other hand, we pair a passage from an article with mid-passages from other articles to create negative instances. We purposely skip the first and last passage of an article from negative instances, as usually introductory and conclusive passages can have similar context and ideas irrespective of the article topics.

Third, we also leverage the output of TF-IDF. Even if TF-IDF may fail with a lower number of matched terms, it can pick the comments with a good number of matched terms. This method provides good support for the top few recommendations. We pair the top-recommended comments with the article to create positive instances. Similarly, we couple the bottom-recommended comments with the article to create negative instances.

With the above three strategies, the final training set contains $\sim 147\text{K}$ positive and negative instances. As DLM develops a language model for each article, we are not required to create training instances for DLM.

4.3.1 Creating the Test Dataset

To evaluate our model on unseen data, we need a set of articles for which the informative and uninformative comments are annotated. Three human annotators, who are Journalism & Communication graduates, annotated ~ 1700 comments in 27 articles. They score each comment on a scale of -1 to 9, where -1 represents a junk comment and 9 represents the comment as highly informative. Any comment having a score of three or more is considered informative. The detailed annotation guide is provided as a supplementary doc.

Chapter 5

Experiments and Results

We address the following research questions.

- How do the proposed methods fare against the baselines and existing methods in terms of informativeness? (see 5.3)
- Do the proposed methods capture editor picks? (see 5.4)
- Do the selected informative comments make sense qualitatively? (see 5.5)

5.1 Experimental Setup

For DLM, we build a language model for each article to extract its informative comments, and we use 0.6 as a cut-off threshold for creating primary (θ_p) and auxiliary queries (θ_a). For the informativeness score (I), we use β as 0.5. As for the CLM model, we leverage the HuggingFace BERT[39] pre-trained model for our bi-encoder model, which helps us build a proper training specification and scheme for the train data. Here, the model’s embedding size is 256, and we use 0.001 as the learning rate with the Adam optimizer. After multiple ablation studies, the inner-product optimized bi-encoder model is the final model.

Even if we annotated the dataset with three annotators, the informative scores given by these annotators vary highly for some comments. This is expected as the annotators have biases and different levels of experience in the topics discussed

in the articles and comments. To reduce the variation in the informativeness scores, we convert the informativeness scale to binary classes—informative and uninformative—using a threshold score of 3.

5.2 The Baseline Models

We compare DLM and CLM against multiple models. We use TF-IDF and BM25 as baseline methods for this study. We also compare our methods against the transformer-based dense passage retriever model (DPR) [38], a supervised model, and Contriver [40], an unsupervised method. We choose DPR and Contriver, as they are well-known for capturing text context.

5.3 Method Comparison with Informativeness

Table 5.1: Mean precision(MP)(%) for each set of recommended K comments for all of the models. For top- K comments, CLM outperforms the other methods. MP drops with the increase of K .

Model/Pre@K	MP@5	MP@10	MP@15	MP@20
TF-IDF	90	86	83	80
BM25	89	88	86	85
DPR	90	89	87	84
Contriver	93	90	86	80
DLM	91	87	85	84
CLM	94	91	90	88

We compare our proposed methods against four other methods using the test dataset (see Sec. 4.3.1). For each article, these methods recommend K most informative comments for $K \in \{5, 10, 15, 20\}$. We then calculate mean precision at K (MP@K) for each of the methods (see Table 5.2). CLM outperforms all of the models in terms of informativeness for all K . DPR and Contriver perform better than the baselines and DLM, which implies that capturing context helps identify more informative comments. Although DLM performs better than TF-IDF and does not outperform BM25, DLM has utility as an online method as it

requires no prior training. DLM also resolves the issues of high dependency on term frequency and document length, which potentially overlooks other important factors like document structure and relevance feedback.

5.4 Method Comparison with Editor Picks

Table 5.2: Mean precision(MP)(%) of editor picks for each set of recommended K comments for all of the models.

Method/EP@K	MP@5	MP@10	MP@15	MP@20
TF-IDF	7	21	21	58
BM25	16	50	66	83
DPR	16	25	41	41
Contriver	14	28	57	64
DLM	16	33	66	66
CLM	16	33	68	68

We also evaluate our methods using editor picks. For each method, we count the number of editor picks in the recommended set of comments for each article (see 5.3) and calculate the mean precision. We observe that BM25 outperforms other methods for $K = 10$ and $K = 20$. CLM and DLM perform as well as BM25 for the rest of the K s. As the number of average editor picks is very low, the precision drops across methods. The other issue with editor picks is that most of them are tagged as editor picks within some hours of publishing the articles (see Appendix 3.2.8). The system does not update the editor picks with the increase of comments. Despite such data quality issues with editor picks, CLM and DLM provides reasonable results for some settings. The reason could be that CLM is trained to understand the context in a general setting.

5.5 Qualitative Analysis of Informative Comments

In this section, we present three case studies on informative comments identified by CLM in top-5 recommendations and discuss the merit of such selection.

Case Study 1

Article: The beach is a melting pot – the perfect place to examine what has shattered our confidence in Europe

Article Topics: EU, Greece, Beach, Economy

User Comment: *"Have just been on a beautiful beach on Syros in Greece all day. Sunshine, grilled sardines, someone playing a guitar. Kids playing, grannies chatting in the shade of an olive tree, everybody being nice to each other. And beautiful blue sea. No wonder the Germany wants to punish this place. Life is better here than it is in Stuttgart, they can keep their BMWs"*

Explanation: The article discusses the Greece and EU economy and how the larger powers handled the economic crisis of Greece. This comment is a kind of satirical, but due to the lack of matched terms, TF-IDF, BM25, and DLM do not identify this comment as informative. DPR and Contriver find this comment in the top 10 and 20, respectively.

Case Study 2

Article: Planning regulations overlook heat – so developers build death traps

Article Topics: EU, City planning, Heating, Warm weather

User Comment: *I'm not sure about the physics involved where the author says that better insulation makes houses warmer in the summer. Surely better insulation makes houses warmer when you are heating them, in the winter, and cooler when the heat source is outside the house ie in the summer. So the emphasis in current building regs on improving insulation should also help with over heating, even where that is not its direct target. Am I missing something here?*

Explanation: The article discusses the city planner, their issues, and the warm weather in the UK. The user comment provides extra information about the cause of the insulation and heating of the houses. Along with giving additional information, it also asks a question for the new reader as well as for the article writer. The other models fail to identify the comment as an informative one.

Case Study 3

Article: Top US court rules for Muslim woman denied Abercrombie job over hijab

Article Topics: Hijab, Religion

User Comment: *I think it's about time for those of us who want freedom from religion to get some rights. That is the worry of so many religions and is why the other religions pitched in. Our world is becoming more polarized by the day and that people should keep their religious practices in their homes and places of worship. And, I am a Christian who believes in God but not in organized religions. So, she has the right to wear a hijab. The store should have the right not to hire her.*

Explanation: The topic of religion is too large that it is hard to capture a person's view with a small set of words. This comment discusses religion and provides information about the existence of too much religion and organized religion. Although the user discusses precisely the same topics as the article, only the contriver identifies this comment as an informative one in the top 10 recommendations.

5.6 Ablation Study

Due to the challenges with determining accurate informativeness scores from annotators, we faced difficulties in assigning proper scores, especially when comments received vastly different scores from different annotators. To address this, we focused on a binary classification approach, considering comments with an informativeness score of 3 or above as informative. However, we also explored various scales of informativeness scores for testing the models.

In most cases, both CLM and DLM performed well, and as the informativeness score increased, the difference in scores also increased. Nonetheless, evaluating the models with discrepant annotation scores proved to be complex.

Another interesting observation with the annotated data was that annotators

seemed to miss assessing the informativeness of comments that lacked common words with the associated article. Instead, they tended to skim through the comments for known keywords already present in the article. This discovery highlights the need to improve the annotation guide to mitigate such issues in the future.

Overall, the research identified several challenges in the evaluation process and identified potential areas for improvement and future tasks, such as refining the annotation guide for more accurate assessments of comment informativeness.

Table 5.3: Mean Precision@K for informativeness score of 4,5,6,7.

method/MP@K	MP@5	MP@10	MP@15	MP@20
DPR	73 / 50 / 29 / 12	63 / 41 / 22 / 8	61 / 38 / 21 / 9	58 / 35 / 18 / 7
TFIDF	63 / 36 / 16 / 6	60 / 34 / 12 / 4	54 / 30 / 11 / 4	54 / 30 / 10 / 4
Contriver	67 / 55 / 36 / 15	65 / 48 / 26 / 11	60 / 42 / 22 / 9	57 / 37 / 18 / 5
BM25	68 / 43 / 20 / 6	68 / 39 / 20 / 7	67 / 37 / 17 / 6	64 / 36 / 15 / 5
DLM	75 / 55 / 27 / 5	70 / 48 / 20 / 4	67 / 44 / 17 / 4	65 / 42 / 14 / 4
CLM	72 / 52 / 24 / 11	68 / 43 / 21 / 9	67 / 38 / 17 / 7	65 / 36 / 15 / 6

Chapter 6

Discussions

Our proposed methods have two different utilities: a) DLM method is an online method and requires no prior training and b) CLM is effective but needs a large training set and is computation-heavy. Although the proposed models outperform the baselines, we prefer CLM over DLM because of its robustness against lexically different yet semantically similar comments. An interesting aspect of this model is that it comprehends the context of the article itself. While it is safe to assume the news article is civil, the dense representation learned by CLM will always be inclined to the civil part of the context. As civility is an important aspect of editors' picks construction and comments moderation, the model has good prospects for addressing that issue. In the future, we plan to extend this research in different directions. E.g., we can remove the dependency on a single news source, vary the embedding size, filter out comments from bots, and improve the training data augmentation process to build better training datasets.

6.1 Limitations

Here we enumerate some of the limitations regarding the datasets and methods.

- We are currently using data from a single source, the Guardian, as it is hard to find large datasets for news articles and user comments that are publicly

accessible and have editor picks. The editors' picks of the Guardian are biased by the perspective of its own editorial board.

- We do not implement a civility detection method as we considered the articles as civil. As the CLM method learns from pair of articles and user comments, the method may fail to include civil if this assumption fails.
- For some comments, the annotators' scores have high variation due to their biases.
- We do not implement bot detection. However, CLM might detect some bots as it uses article data to train itself.
- For CLM, we use a fixed embedding size, which we do not vary.
- Also seems reading an article and later reading its comments also biases the annotator for the purpose of ranking put users to look for common words as comments, which leads to skipping comments not having common words.

6.2 Future Work

The future direction of the work is as follows:

- Introduce automated article detection in the crawler. Right now we are using pattern-based article detection. But it requires an initial manual validation to add new sources. Automated article detection will make it easier to add new sources.
- Add more sources for user comments. The current model is trained and tested on Guardian Dataset. we are already working on introducing NY-Times data.
- One of the major bottleneck is the annotation of data for ground truth. This process is time-consuming and costly. Though it's hard to maintain consistency. We have already built an annotation guide. Result analysis introduces new issues with annotation consistency. We need to improve the

annotation guide to create more consistency.

- Introducing longer dense representation for CLM.
- Training on data pairs from a single source might introduce biases based on sources. Introducing cross-source training data to resolve this issue.
- As of hypothesis of an article from the reputed source being civil. we skipped the civility detection. But introducing more sources might fail this hypothesis. We need to introduce civility detection.

6.3 Ethics Statement

We collected data using the Guardian API, and The Guardian permits their data to be used for research. As for user comments, the Guardian explicitly explained that this data would be public and might be collected by a third party. For more information, please check the following.

- [Open Licence Terms](#)
- [Privacy Policy](#)
- [Terms of Service](#)

The test data was annotated in collaboration with the BRAC University in Dhaka, Bangladesh. The annotators were compensated fairly by BRAC.

Bibliography

- [1] D. Domingo, T. Quandt, A. Heinonen, S. Paulussen, J. B. Singer, and M. Vujnovic, “Participatory journalism practices in the media and beyond,” *Journalism Practice*, vol. 2, no. 3, pp. 326–342, 2008. [Online]. Available: <https://doi.org/10.1080/17512780802281065>
- [2] D. S. Chung and S. Nah, “Community newspaper editors’ perspectives on news collaboration: Participatory opportunities and ethical considerations toward citizen news engagement,” *Journalism Practice*, vol. 16, no. 7, pp. 1306–1326, 2022. [Online]. Available: <https://doi.org/10.1080/17512786.2020.1867621>
- [3] S. Robinson, “Traditionalists vs. convergers: Textual privilege, boundary work, and the journalist—audience relationship in the commenting policies of online news sites,” *Convergence*, vol. 16, no. 1, pp. 125–143, 2010. [Online]. Available: <https://doi.org/10.1177/1354856509347719>
- [4] R. S. Sumpter, “Daily newspaper editors’ audience construction routines: A case study,” *Critical Studies in Media Communication*, vol. 17, no. 3, pp. 334–346, 2000. [Online]. Available: <https://doi.org/10.1080/15295030009388399>
- [5] J. Edson C Tandoc, “Journalism is twerking? how web analytics is changing the process of gatekeeping,” *New Media & Society*, vol. 16, no. 4, pp. 559–575, 2014. [Online]. Available: <https://doi.org/10.1177/1461444814530541>
- [6] C. P. Natalie Jomini Stroud, Emily Van Duyn, “Survey of com-

- menters and comment readers,” <https://mediaengagement.org/research/survey-of-commenters-and-comment-readers/>, 2016.
- [7] M. Green, “No comment! why more news sites are dumping their comment sections,” <https://www.kqed.org/lowdown/29720/no-comment-why-a-growing-number-of-news-sites-are-dumping-their-comment-sections/hyperref>, 2018.
- [8] N. Diakopoulos, “Picking the nyt picks: Editorial criteria and automation in the curation of online news comments,” *ISOJ Journal*, vol. 6, no. 1, pp. 147–166, 2015.
- [9] H. Kobayashi, H. Taguchi, Y. Tabuchi, C. Koleejan, K. Kobayashi, S. Fujita, K. Murao, T. Masuyama, T. Yatsuka, M. Okumura, and S. Sekine, “A case study of in-house competition for ranking constructive comments in a news service,” in *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*. Online: Association for Computational Linguistics, Jun. 2021, pp. 24–35. [Online]. Available: <https://aclanthology.org/2021.socialnlp-1.3>
- [10] S. Robertson, H. Zaragoza *et al.*, “The probabilistic relevance framework: Bm25 and beyond,” *Foundations and Trends® in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.
- [11] D. Yin, Y. Hu, J. Tang, T. Daly, M. Zhou, H. Ouyang, J. Chen, C. Kang, H. Deng, C. Nobata, J.-M. Langlois, and Y. Chang, “Ranking relevance in yahoo search,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’16. New York, NY, USA: Association for Computing Machinery, 2016, p. 323–332. [Online]. Available: <https://doi.org/10.1145/2939672.2939677>
- [12] D. Park, S. Sachar, N. Diakopoulos, and N. Elmqvist, “Supporting comment moderators in identifying high quality online news comments,” in *Proceedings*

- of the 2016 CHI Conference on Human Factors in Computing Systems, 2016, pp. 1114–1125.
- [13] D. Giomelakis and A. Veglis, “Employing search engine optimization techniques in online news,” *Studies in media and communication*, vol. 3, no. 1, pp. 22–33, 2015.
- [14] D. M. Blei and M. I. Jordan, “Modeling annotated data,” in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2003, pp. 127–134.
- [15] I. Titov and R. McDonald, “Modeling online reviews with multi-grain topic models,” in *Proceedings of the 17th international conference on World Wide Web*, 2008, pp. 111–120.
- [16] M. K. Das, T. Bansal, and C. Bhattacharyya, “Going beyond corr-lda for detecting specific comments on news & blogs,” in *Proceedings of the 7th ACM international conference on Web search and data mining*, 2014, pp. 483–492.
- [17] Z. Ma, A. Sun, Q. Yuan, and G. Cong, “Topic-driven reader comments summarization,” in *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012, pp. 265–274.
- [18] D. K. Sil, S. H. Sengamedu, and C. Bhattacharyya, “Supervised matching of comments with news article segments,” in *Proceedings of the 20th ACM international conference on Information and knowledge management*, 2011, pp. 2125–2128.
- [19] H. Zhu, F. Nan, Z. Wang, R. Nallapati, and B. Xiang, “Who did they respond to? conversation structure modeling using masked hierarchical transformer,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 9741–9748.

- [20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [21] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, “Transformer in transformer,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 15 908–15 919, 2021.
- [22] J. Meneghello, N. Thompson, K. Lee, K. W. Wong, and B. Abu-Salih, “Unlocking social media and user generated content as a data source for knowledge management,” *International Journal of Knowledge Management (IJKM)*, vol. 16, no. 1, pp. 101–122, 2020.
- [23] Y. Sujana, J. Li, and H.-Y. Kao, “Rumor detection on twitter using multiloss hierarchical bilstm with an attenuation factor,” *arXiv preprint arXiv:2011.00259*, 2020.
- [24] A. Q. Mir, F. Y. Khan, and M. A. Chishti, “Online fake review detection using supervised machine learning and bert model,” *arXiv preprint arXiv:2301.03225*, 2023.
- [25] R. Rivaldo, A. Amalia, and D. Gunawan, “Multilabeling indonesian toxic comments classification using the bidirectional encoder representations of transformers model,” in *2021 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*. IEEE, 2021, pp. 22–26.
- [26] L. Ilias, S. Mouzakitis, and D. Askounis, “Calibration of transformer-based models for identifying stress and depression in social media,” *IEEE Transactions on Computational Social Systems*, 2023.
- [27] D. Rozado, R. Hughes, and J. Halberstadt, “Longitudinal analysis of sentiment and emotion in news media headlines using automated labelling with transformer language models,” *Plos one*, vol. 17, no. 10, p. e0276367, 2022.

- [28] M. Alam, A. Iana, A. Grote, K. Ludwig, P. Müller, and H. Paulheim, “Towards analyzing the bias of news recommender systems using sentiment and stance detection,” in *Companion Proceedings of the Web Conference 2022*, 2022, pp. 448–457.
- [29] O. Dalal, S. H. Sengemedu, and S. Sanyal, “Multi-objective ranking of comments on web,” in *Proceedings of the 21st international conference on World Wide Web*, 2012, pp. 419–428.
- [30] C.-F. Hsu, E. Khabiri, and J. Caverlee, “Ranking comments on the social web,” in *2009 International Conference on Computational Science and Engineering*, vol. 4, 2009, pp. 90–97.
- [31] C. Zhai and J. Lafferty, “Two-stage language models for information retrieval,” in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR ’02. New York, NY, USA: Association for Computing Machinery, 2002, p. 49–56. [Online]. Available: <https://doi.org/10.1145/564376.564387>
- [32] H. Po“ttker, “News and its communicative quality: the inverted pyramid—when and why did it appear?” *Journalism Studies*, vol. 4, no. 4, pp. 501–511, 2003. [Online]. Available: <https://doi.org/10.1080/1461670032000136596>
- [33] M. Kleemans, G. Schaap, and M. Suijkerbuijk, “Getting youngsters hooked on news,” *Journalism Studies*, vol. 19, no. 14, pp. 2108–2125, 2018. [Online]. Available: <https://doi.org/10.1080/1461670X.2017.1324316>
- [34] T. I. DeAngelo and N. S. Yegiyani, “Looking for efficiency: How online news structure and emotional tone influence processing time and memory,” *Journalism & Mass Communication Quarterly*, vol. 96, no. 2, pp. 385–405, 2019. [Online]. Available: <https://doi.org/10.1177/1077699018792272>

- [35] F. Jelinek, “Interpolated estimation of markov source parameters from sparse data,” in *Proceeding of the Workshop on Pattern Recognition in Practice*, 1980, pp. 381–397.
- [36] D. Chen, A. Fisch, J. Weston, and A. Bordes, “Reading wikipedia to answer open-domain questions,” *arXiv preprint arXiv:1704.00051*, 2017.
- [37] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, “Retrieval augmented language model pre-training,” in *International conference on machine learning*. PMLR, 2020, pp. 3929–3938.
- [38] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, “Dense passage retrieval for open-domain question answering,” *arXiv preprint arXiv:2004.04906*, 2020.
- [39] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [40] G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, and E. Grave, “Towards unsupervised dense information retrieval with contrastive learning,” *CoRR*, vol. abs/2112.09118, 2021. [Online]. Available: <https://arxiv.org/abs/2112.09118>