ASSESSING AND ENHANCING THE QUALITY OF NEWS HEADLINES USING MACHINE LEARNING

AMIN OMIDVAR

A DISSERTATION SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

March 2025

Abstract

Headlines play a pivotal role in capturing readers' attention, and their quality is critical for engaging audiences. In this thesis, we propose various solutions to assist news media in crafting high-quality headlines. First, we delve into headline quality assessment, devising four innovative indicators that automatically evaluate headlines' quality. Our proposed model empowers news outlets to automatically determine the quality of published headlines. We evaluate the quality of headlines from The Globe and Mail using these four indicators and provide insightful results. We then use this labeled data to train our novel headline quality prediction model to predict the quality of unpublished headlines, assisting journalists in selecting high-quality headlines for their articles. Furthermore, we facilitate journalists' work by recommending high-quality headlines for their articles. To accomplish this, we propose a headline generative model that learns to generate headlines using Reinforcement Learning (RL). Our model can be optimized not only with respect to a non-differentiable metric but also based on a combination of two different metrics simultaneously. Additionally, we enhance headline generation in terms of both training speed and the quality of the generated headlines by proposing a novel architecture utilizing state-of-the-art transformer models. In our architecture, after generating candidate headlines using state-of-the-art models, we select the most popular headline using our headline popularity

prediction model. Moreover, we establish a popularity benchmark for evaluating headline generation models based on their ability to generate popular headlines. Lastly, we forecast changes in how people consume news articles, envisioning a shift towards interacting with agents instead of navigating news portals. To address existing challenges and enable this transition, we introduce Semantic In-Context Learning (S-ICL), an innovative approach enabling Large Language Models (LLMs) to deliver updated news in a conversational format, enhancing user engagement and comprehension for news media.

Dedication

To my parents,

For your unwavering love, belief in me, and endless encouragement, from my first classroom to the final chapter of this journey, you have always been my greatest support. With deepest appreciation and all my heart, Thank you.

Acknowledgement

I would like to express my gratitude to Prof. Aijun An for her invaluable guidance and consistent support during my PhD program. Her immense knowledge and experience not only elevated my work but also sharpened my critical thinking skills.

I would also like to extend my thanks to the members of my examining committee, Prof. Natalija Vlajic, Prof. Parke Godfrey, Prof. Manos Papagelis, as well as the members of the EECS department at York University for their invaluable assistance, guidance, and support throughout this journey.

I thank my parents, Hossein Omidvar and Akram Ashayeri, for their unwavering love, support, motivation, and all they have done for me. Without them, I would not have embarked on this path – I did it for them. I would also like to express my gratitude to my brothers, Iman Omidvar and Omid Omidvar, for their constant love and support.

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

1.1. Motivations and Objectives

In this thesis, we tackle several challenges related to news headline evaluation and generation. As the quality of headlines is crucial, we further conduct research on headline quality. However, there was no clear definition of the headline quality. Thus, a better definition and measurement of headline quality is needed. We define high-quality headlines as those that are both accurate (i.e., catching the essence of the article content) and attractive to readers. We propose a novel method to determine the quality of headlines, also predicting the quality of unpublished headlines for news media using users log history.

In addition to measuring headline popularity and qualities, automatic headline generation has also been studied. Almost all of them consider headline generation as a text summarization problem. However, there are some key differences that make the task of headline generation more challenging. Firstly, in the headline generation task, the output should consist of only one sentence, whereas in text summarization, multiple sentences can be generated. This condition makes the headline generation task more difficult, especially when using extractive approaches. Secondly,

the generated headline should be interesting enough to catch the readers' attention. Otherwise, a good article may be ignored by readers. To the best of our knowledge, none of the previous headline generators [1]–[4] have taken into account the popularity of headlines when generating or evaluating them.

We have a prophecy that in the near future, the way people consume news will undergo a significant transformation. A growing number of individuals will turn to conversational agents to stay informed about current events. It would be highly beneficial for news media outlets to adopt chatbots capable of engaging with their end-users. Today, thanks to the advancements in Large Language Models (LLMs), this scenario is on the cusp of becoming a reality. However, deploying LLMs in such a scenario presents several significant challenges. Firstly, many news media outlets possess exclusive articles that are not publicly accessible. Since LLMs are trained predominantly on public datasets, they might lack knowledge of these proprietary pieces of content. Moreover, user queries might pertain to the internal policies of the news media, such as subscription details and user access limits, which are only addressable through the outlet's specific rules and guidelines. Secondly, the process of fine-tuning LLMs with a news outlet's private data is fraught with complications. The substantial size of LLMs, the costs associated with training, the dynamic nature of internal data, and concerns over data privacy all pose significant barriers. For instance, news articles are published daily, and LLMs, if not updated, would not have knowledge of these new developments. This gap could hinder the ability of an LLM-based agent to provide users with up-to-date news or answer questions about current events. Furthermore, fine-tuning options may not even be available for certain LLMs, such as OpenAI's GPT-4. Third, prior research indicates that the formulation of prompts, the selection and arrangement of samples, and the structure of inputs can unpredictably and substantially impact the performance of LLMs (Liu et al. 2022; Min et al.

2022; Sanh et al. 2021; Wei et al. 2023). Lastly, presenting an LLM with all available samples is impractical due to the high computational costs and constraints on input size.

1.2. Contributions

In this thesis, we propose a variety of models to tackle the challenges outlined in the previous section. The contributions of this thesis are summarized as follows:

1. **Defining Headline Quality:** To establish a more precise definition of high-quality headlines, we developed four headline quality indicators in Chapter 3 [5]. Our approach enables the automatic evaluation of headline quality by utilizing the log history of news portals. Our approach to labeling has yielded valuable insights for The Globe and Mail and has been adopted by others [6], [7].
2. **Headline Quality Prediction:** In addition, we have proposed a novel architecture that includes a transformer encoder, topic modeling, and a similarity matching matrix to predict the quality of unpublished headlines based on the four proposed quality indicators in Chapter 3. The model can assist journalists in assessing the quality of their headlines prior to publication [5]. Furthermore, we have enhanced the performance of the headline quality prediction model by incorporating sentiment analysis into our model's architecture using transfer learning techniques [8].
3. **Headline Popularity Prediction:** To mitigate the impact of biases existing in users' log history datasets, we assess headline popularity based on the number of likes received on Twitter. Considering that most news outlets maintain Twitter accounts to disseminate their headlines, we've compiled a new dataset, named HP (Headline Popularity), using statistical

data sourced from Twitter. Subsequently, we designed and trained a model to forecast headline popularity, employing a transformer encoder architecture for this purpose.

4. **Generating High Quality Headlines using RL:** To address the challenges in headline generation, we utilize Reinforcement Learning (RL) to train a generative model [9] for the headline generation task in Chapter 4. Our reward function can integrate multiple metrics—ROUGE [10], METEOR [11], BERTScore [12], and popularity score—allowing the model to be directly optimized based on evaluation metrics during training. The policy, defined by the parameters of our headline generator, guides the model in selecting actions (i.e., predicting the subsequent word at each step). During our experiments, we faced a significant challenge as the model struggled to converge because of the vast number of potential actions, which corresponded to the dictionary's word count. To address this issue, we adopted the approach of [9] combining the cross-entropy loss function with the RL reward function, which effectively accelerated convergence. However, since using RL for headline generation is slow and challenging to converge during training, we decided to propose another method based on Transformers to further enhance the results.
5. **Generating High Quality Headlines using Transformers:** To address the challenges in headline generation, we have proposed a novel headline generation model [13] utilizing transformer generative models to produce higher quality headlines, as detailed in Chapter 5. The proposed model has two major parts: generation and selection. The first part generates headlines for a news article, while the second part predicts the popularity of the generated headlines and selects the one with the highest popularity score. In the proposed method, we first fine-tune state-of-the-art open-source transformer models for the headline generation task and use them to produce multiple headlines for an article. Next, we employ

our headline popularity prediction model (trained on the HP dataset) to predict the popularity of each generated headline and choose the headline with the highest predicted score.

6. **Semantic In-Context Learning:** To address the mentioned problems of using LLMs to inform users regarding the most updated news, we propose Semantic In-Context Learning (S-ICL) which leverages a semantic search engine using an SBERT model [14] and an LLM (ChatGPT with the gpt-3.5-turbo engine) to create a news conversational agent in Chapter 6 [15]. This agent not only benefits from the knowledge of an LLM but also utilizes private knowledge sources to provide accurate answers to inquiries. Additionally, we suggest a flexible architecture that allows experts to apply and compare different prompt engineering approaches. The proposed model was developed and participated in the BEA 2023 Shared Task [16]. The model is versatile and can be used in other domains such as news media, customer service, etc.

1.3. Dissertation Organization

The organization of this thesis is as follows. Chapter 2 presents a survey of the most pertinent prior research in the areas of headline popularity prediction, and headline generation. In chapter 3 we propose four headline quality indicators to represent the quality of headlines [5]. Leveraging the log history of users on news portals, our proposed model can autonomously evaluate the quality of published news headlines. Additionally, we introduce an innovative ensemble deep learning model that employs transformer encoders, topic modeling, and

semantic matrix similarity for assessing the quality of unpublished headlines. We have further enhanced our model by integrating sentiment latent features through transfer learning [8].

In Chapter 4, we discuss the differences between the headline generation task and the summarization task. We also mention the challenges that exist in training models for the headline generation task. To address these challenges, we utilize Reinforcement Learning (RL) to train a headline generator model. We show not only how a generative model can be trained on non-differentiable metrics, but also how it can be trained based on multiple metrics simultaneously for the headline generation task. Additionally, we create a new headline popularity (HP) dataset by crawling statistical information regarding the headlines posted on Twitter.

Then, in Chapter 5, we propose a novel approach to generate high-quality headlines using transformer models. Our proposed model encompasses two main parts: generation and selection. We additionally utilize a transformer model for the headline popularity prediction task and train it on our HP dataset. Furthermore, we introduce an approach to automatically evaluate the headline generator models concerning the generation of popular headlines.

We predict that in the future, many users will interact with Generative AI to stay informed about the news instead of reading articles on news portals. In Chapter 6, we discuss the challenges of using LLMs as conversational agents for this purpose. We also introduce a novel method called Semantic In-Context Learning (S-ICL), which combines semantic search and a Large Language Model (LLM) to address these challenges [15]. We apply our proposed method to BEA's shared task [16], and our model achieved the fourth place in both the development and test phases of the competition. Finally, the thesis is wrapped up with the conclusion and future works in Chapter 7.

1.4. Publications

The research undertaken for my dissertation has culminated in the following published papers.

- Amin Omidvar, Aijun An. *"Empowering Conversational Agents using Semantic In-Context Learning".* In Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023), pages 766–771, Toronto, Canada. Association for Computational Linguistics. (Described in **Error! Reference source not found.**)
- Amin Omidvar, Aijun An. *"Learning to Generate Popular Headlines"***.** IEEE Access. 11, 60904–60914 (2023)(a). https://doi.org/10.1109/ACCESS.2023.3286853. (Described in Chapter 5)
- Amin Omidvar, Hossein Pourmodheji, Aijun An, Gordon Edall. *"A Novel Approach to Determining the Quality of News Headlines"*. Natural Language Processing in Artificial Intelligence—NLPinAI 2020. Springer, Cham, 2021. 227-245. (Described in Chapter 3)
- Amin Omidvar, Hossein Pourmodheji, Aijun An, Gordon Edall. *"Learning to Determine the Quality of News Headlines"*. In: Proceedings of the 12th International Conference on Agents and Artificial Intelligence - Volume 1: NLPinAI,. pp. 401–409. SciTePress (2020). (Described in Chapter 3)

The papers listed below stem from projects I pursued during my PhD studies.

- Amin Omidvar, Hui Jiang, Aijun An**. "***Using Neural Network for Identifying Clickbaits in Online News Media. (eds) Information Management and Big Data"*. SIMBig 2018. Communications in Computer and Information Science, vol 898. Springer, Cham, (2019).
- Lixian Liu, Amin Omidvar, Zongyang Ma, Ameeta Agrawal, Aijun An. *"Unsupervised Knowledge Graph Generation Using Semantic Similarity Matching."* Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing. 2022.
- Sara Gilani, Elizabeth LeRiche, Pierre Duez, Amin Omidvar, Aijun An, Jennifer McArthur. "*Optimizing Energy with Machine Learning Grey-Box Models*". 36th CIB W78 2019

1.5. Summary

In this chapter, we introduced the motivations and objectives of the research. The chapter highlighted the importance of headlines in attracting users' attention and the need for news media to develop approaches to detecting and improving headline quality. We addressed the problem of lacking a concise definition of headline quality and the challenges associated with determining the headline quality. The chapter further explored the challenges in headline generation, including the need for concise and attention-grabbing headlines. In addition, we discussed the limitations of using language models for conversational agents in news media. Finally, the structure of the thesis is outlined at the end of the chapter.

Chapter 2
Literature Review

2.1. News Headline Popularity Prediction

The prediction of news headline popularity has become an important research topic in recent years due to its importance for online news media. Several factors contribute to the popularity of this research area. Firstly, news media have gathered a significant amount of data regarding their users' clickstream. Having this valuable data allows them to conduct research on the popularity of news articles and extract useful insights. Secondly, due to intense competition among online news media, research in this area can play a crucial role in the success of news media in this competitive environment.

2.1.1. Traditional ML Methods

One interesting study measured the click-value of individual words in headlines [17]. The researchers collected 150,000 news articles that were published on Yahoo's home page over a four-month period. They then introduced the Click-Through Rate (CTR) and word Click-Through Rate (wCTR) methods to assess the click-value of news headlines and individual words within headlines, respectively. The researchers hypothesized that individual words have a significant impact on the popularity of news articles by carrying a certain click-value, and their

findings supported this assumption. For instance, they demonstrated that words related to celebrities received more clicks than those related to business, indicating a higher mean click-value for the former. Additionally, they proposed the Headline Click-based Topic Model (HCTM) using Latent Dirichlet Allocation (LDA) to identify words that can generate more clicks for headlines. The authors acknowledged a limitation of their approach, namely the influence of personalization algorithms on their results. Yahoo's personalization algorithm would display the most relevant articles to each user, potentially introducing a personalization bias.

In another related research [18], a useful software tool was developed to help authors compose effective headlines for their articles. The software utilizes state-of-the-art NLP techniques to recommend keywords to authors for inclusion in article headlines, making the headlines more engaging. The researchers calculated two local and global popularity measures for each keyword and employed a supervised regression model to predict the likelihood of headlines being widely shared on social media. Their keyword ranking system was designed based on the assumption that important keywords, which have frequently appeared in past significant articles, should also appear in the given news article. Additionally, they analyzed the shareability of each news article by collecting 700K tweets using Twitter's streaming API. This tool can be an asset in the news industry, assisting editors in effectively targeting their social media strategy. The authors already mentioned that their system fails to recognize when pairs of equivalent but non-identical keywords have the same referent (e.g., GPO and General Post Office). Another drawback of the system is that it depends on Google News' ranking algorithm, which is not only a black box system but also changes over time.

In [19], the authors extracted features from the content of 69,907 news articles in order to find approaches that can help attract clicks. They discovered that the sentiment of the headline

is strongly correlated with the popularity of the news article. Most of the published news articles have negative headlines, and the production rate of negative headlines has remained constant over time [19]. They also showed that headlines that are intensely positive or negative tend to attract more clicks, while neutral ones receive fewer clicks. Additionally, they found that people in a bad mood tend to read more good news than people in a good mood, and vice versa. They gathered a news dataset by crawling data from The New York Times[1], Dailymail Online[2], BBC[3], and Reuters[4] from January 2013 to September 2014. The authors also mentioned a potential limitation of their approach due to the existing bias regarding the number of clicks on Bitly. They admitted they do not know exactly whether the click was due to a news headline or other factors. They also mentioned that users could share the URL of the article in other forums or social media, presenting it completely differently from its original headline.

Multi-dimensional features were extracted from news articles to predict the online popularity of news headlines [20]. The news dataset was collected by crawling Feedzilla, a news feed aggregator. Regression algorithms such as linear regression and support vector machine regression, as well as classification algorithms like support vector machine and decision tree, were employed for the prediction task. In the study conducted by [21], journalism-inspired news values such as Entities, Polarity, Surprise, and Uniqueness, as well as linguistic style features including

---

[1] http://www.nytimes.com

[2] http://www.dailymail.co.uk

[3] http://www.bbc.co.uk

[4] http://www.reuters.com

Brevity, Simplicity, Unambiguity, Punctuation, Nouns, Verbs, and Adverbs, were utilized to forecast the popularity of headlines. The prediction model employed support vector machine regression with an RBF kernel. Additionally, the popularity of each news article was inferred using Facebook and Twitter posts. The two mentioned research works could improve the performance of their methods by training state-of-the-art machine learning models, such as transformers, on their data.

The Reddit community is one source that can be used to infer the popularity of news headlines using likes and comments [22]. Reddit is a social news-aggregation platform consisting of interest-based communities called subreddits. Each subreddit has subscribers who can vote on posts, comment on them, and submit new posts. In a study by Horne et al. [22], Reddit was used to train machine learning algorithms to predict the popularity of news articles. The authors also identified biased news articles by using features such as bias word count and number of hedges, which capture the level of opinion or one-sidedness in each article. As the authors themselves mentioned, the Reddit titles change over time. They found that 85 percent of titles in 2012 and 72 percent of them in 2013 were changed by at least one word. Changing the titles may significantly affect the results of this study in a way that an unpopular headline may turn into a popular one, and vice versa.

A test-rollout strategy has been used at Yahoo's home page to choose the best headline variant in terms of users' points of view [23]. In this strategy, multiple variations of a news headline are shown to randomized equal size user buckets for a defined period. Then the headline variant which gained the most clicks will be selected for the article permanently. This strategy has been used to select the best banner for online advertising as well. However, a large proportion of

the article's clicks occur in its early life since freshness is the major key factor of news article popularity.

2.1.2. **RNN based Methods**

In [24], the authors used only headlines to predict the popularity of news articles. The proposed method was evaluated on Russian and Chinese datasets, totaling over 800,000 samples. They focused solely on headlines because users typically click on an article after reading its headline, making the headline the starting point of the user's desire to open the news article. They used Bi-LSTM to predict the number of views for each news article. The news articles used were from RIA News, a major news media outlet in Russia. They considered the number of views as an indicator of the popularity of the news articles. However, it should be noted that the number of views is not solely dependent on the popularity of the article itself. Other hidden factors, such as the size of the font, placement of the news article on the news portal, and the behavior of the recommendation engine, can also influence the number of views.

In another related research [25], the authors employed Bi-LSTM to predict posted news on Facebook using only headlines. This was the first attempt to predict news popularity on social media using only posted news titles. One drawback of the method is that the popularity of the news on Facebook highly depends on the number of followers of the account that shared that news. However, this factor is not considered in this work.

Multimodal features were extracted to determine the social media popularity [26]. Their proposed framework consists of three main stages: feature extraction, feature fusion, and

popularity prediction. ResNet and LSTM were employed to extract features from images and text, respectively. Then, the extracted features were combined using the hierarchical fusion method. Finally, the popularity was calculated using the regression method. However, the proposed model is restricted to using datasets with multinomial data, meaning that not all social media posts contain pictures.

2.1.3. **Transformer based Methods**

In [6], they utilized a similar approach to [5] to combine received clicks and reading time in order to assess the effectiveness of news headlines. Using these metrics, they categorized headlines into four classes: non-effective, appealing, engaging, and effective. Their dataset consists of 17 million clicks and 7198 articles from Kaleva Media. BERT was employed to classify the headlines into these four classes. The proposed method demonstrated superior performance in terms of classification accuracy compared to the journalists. However, a weakness of the proposed method lies in the metrics used for classification, as the number of clicks is influenced by not only the attractiveness of the headlines but also hidden factors such as location, duration, etc.

BERT was employed and compared with LSTM to determine the popularity of headlines [27]. The researchers applied various techniques, such as sentiment analysis and entity extraction, to the headlines and used them as features. They suggested that using their proposed model to predict the impact of a news article would enable us to prioritize news that can have a greater influence on people, thereby avoiding misinformation and fake news. However, they did not propose any specific methods for detecting fake news or misinformation.

Table 2-1 Summary of headline popularity prediction models

Studies	**Dataset**	**Features**	**Algorithms**	**Task and Performance**
[19]	crawled 69907 news articles	Headline, body of article, comments	Sentiment analysis	Finding a relation between headline's popularity and its sentiment score
[17]	Crawled 150K News articles from Yahoo Homepage	Headline, body of article	Headline Click-based Topic Model (HCTM) based on LDA	Regression AUC = 0.87
[18]	700K tweets were collected	tweet	Entity recognition, Sentiment Polarity, Regularized Linear Regression (RLR), RandomForest (RF) and Gradient Boosting Trees (GBT)	Regression MSE = 37.6
[20]	44,000 articles gathered from Feedzilla	Headline, body of article	linear regression, support vector machine regression	A) Regression $R^2 = 0.36$ C) Classification Accuracy = 0.84

[21]	crawled news articles	Headline, body of article	support vector regression with RBF kernel	Regression MAE base on Twitter = 0.68 Facebook = 1.42
[22]	Crawled 72.7K news articles and 154K posts from Reddit	Headline, body of article, and post	Lasso regression algorithm	Ranking Kendall-tau distance and Precision at K
[24]	Following crawled news datasets: A) RIA New[5] 153000 articles in Russian B) KD-net[6] 500000 articles in Chinease C)Sinovison[7] 150000 articles in Chinese	Headlines	BiLSTM	Regression Mean of Day Acceptable deviation = A) 0.7667 B) 0.5151 C) 0.6955

[5] https://ria.ru/

[6] http://www.kdnet.net

[7] http://www.sinovision.net/about/index.htm

[25]	4090 post from NowThisNews Facebook page[8]	Posts	GloVe + BiLSTM	Binary text classification (popular or unpopular) Accuracy = 0.6711
[28]	Crawled 40000 Chinease news articles	Headline and body	SVM Classifier	Muti-class text classification F-score = 0.753
[5]	The Globe and Mail data set	28751 news articles (headline + body)	GloVe + Similarity Matrix+ 2-d CNN + BERT + NNMF + FFNN	Multi-class text classification RAE = 80.56
[6]	Click-stream data from Kaleva Media	17 million clicks and 7198 articles	BERT	Accuracy of 84 %.
[27]	The data provided by Upworthy through A/B testing	Number of clicks, headlines, articles	BERT	Accuracy of 64 %
[26]	Dataset was collected from	Image and text	ResNet-50, Bi-LSTM for feature	MSE of 2.65

[8] https://www.facebook.com/NowThisNews

	Flickr containing 486K labeled data		extraction. And Gradient Boosting Decision Tree (GBDT) for popularity prediction	

2.2. News Headline Generation

News headline generation can be considered as a type of text summarization that aims to create a high-quality headline for a news article. Text summarization is the automated process of generating a brief and concise summary of a text document. There are two different approaches to automatic text summarization: abstractive and extractive summarization.

Extractive summarization methods create a summary that only includes words or sentences from the original document. Most traditional summarization approaches fall into this category. For instance, TextRank is a well-known extractive summarization technique that has been applied in various applications [29]. On the other hand, abstractive summarization techniques can generate words in the summary that are not presented in the original document. Many recent automatic summarization techniques, particularly those utilizing sequence to sequence models, belong to the abstractive group [30].

2.2.1. Traditional Headline Generation Methods

In [3], well-known phrases were inserted into the news headlines to make them more appealing to users. The proposed model selects a well-known phrase, such as a famous proverb, that aligns with the news article and incorporates it into the headline. The effectiveness of this approach was assessed using 1000 samples, each of which was evaluated by three different human evaluators. However, it is possible for the method to insert phrases that are unrelated to the main content of the article. Therefore, a human evaluator screens the generated headline to determine its suitability for use.

Another work [31] employed an extractive approach to generate headlines for Community Question Answering (CQA). They utilized a learning-to-rank technique to identify the most informative information to be used as headlines. Additionally, they created a QA dataset by leveraging Japanese CQA from Yahoo and crowdsourcing. However, considering only the most important questions as the headline may not adequately represent all the ideas and subjects present in all the questions. Abstractive approaches, if properly trained for such datasets, might be able to provide headlines that cover a broader range of topics discussed.

In [32], an extractive headline generation approach was proposed based on keyword extraction. They obtained background knowledge regarding the article by querying Wikipedia. Then, by analyzing inlinks, outlinks, and the category of the retrieved Wikipedia articles, they extracted keywords. Subsequently, they employed a keyword clustering-based procedure to construct headlines from these keywords. While important keywords can contribute to building a better headline, the readability of the generated headline might be poor in some cases due to the way the words are combined.

2.2.2. **RNN based Headline Generation Methods**

The Washington Post uses Headliner, software owned by The Washington Post, to automatically generate and suggest headlines to their authors [1]. Their strategy is to have several variations of headlines for each news article and test them to see which is best to use. One of the main reasons why The Washington Post uses more than one headline for its news articles is that news stories usually cover different aspects of events, which makes it difficult for a single headline to cover all events while capturing users' attention. However, they are using the OpenNMT library [33], which we demonstrate in section 4.5 that other models can outperform this model in the headline generation task.

Attention-based summarization (ABS), a well-known neural network encoder-decoder model for natural language generation tasks [4], was enhanced by incorporating semantic information, such as named entities, to generate headlines from news articles [34]. To encode the output of the AMR parser into a fixed-length embedding representation, an attention-based Abstract Meaning Representation (AMR) encoder was proposed. AMR is a directed acyclic graph (DAG) that encodes the meaning of a sentence, with nodes representing concepts and edges representing relationships between concepts. The results demonstrated that using AMR could improve the performance of neural network encoder-decoder models in generating shorter and better summaries (i.e., headlines), as measured by ROUGE scores.

In [4], a headline generator model using an attention-based encoder-decoder framework for machine translation [35] was proposed. The authors introduced a convolutional attention-based architecture and demonstrated that their proposed modification outperformed the original framework in the headline generation task. However, instead of using the entire article, they only used the first sentence of the article to generate the headlines.

2.2.3. Transformer based Headline Generation Methods

In [2], a model based on the transformer model [36] was used to generate headlines. Instead of constructing a dictionary for abstractive text generation, the authors composed headlines by extracting all the necessary words from their articles. The proposed model was designed to only generate tokens that are either present in the article or defined in their dictionary. According to their evaluation, human evaluators found it challenging to distinguish the machine-generated headlines from human-written ones. However, they also observed that the generated headlines differed from the ground truths. Consequently, their proposed method would likely receive low scores when evaluated using popular metrics such as ROUGE [10], and they did not include such an evaluation.

In [37], transformer models were employed to generate personalized headlines based on the preferences of individual users. The authors published a new dataset called PENS (Personalized News Headlines), in which the training set consists of news content and news readers' click streams from June 14 to July 12, 2019. For the test set, they hired 103 native students to browse news content and write their desirable headlines for the articles that seemed interesting to them. Additionally, they proposed a novel deep-learning model based on transformers to encode both news content and users' behavior to generate personalized headlines. Finally, the generated headlines were compared to the headlines written by human evaluators. However, the headlines written by students are less accurate compared to the original headlines written by professional journalists. Thus, the headlines generated by students cannot serve as a good test set for examining the effectiveness of headline generators in terms of producing high-quality headlines.

Recent large pre-trained transformer models have achieved impressive results when fine-tuned on downstream NLP applications. In a recent study by Zhang et al. [38], they introduced a new pre-training task to train a transformer-based encoder-decoder model that performs well on summarization tasks. They also collected a massive dataset of 1.5 billion news articles to pre-train the transformer model using various pre-training tasks. The authors concluded that aligning the pre-training domain more closely with summarization tasks leads to more efficient transformer models. Although the authors did not publicly release their dataset, it is currently the largest news dataset ever.

Table 2-2 Summary of Headline Generation models

Studies	**Dataset**	**Features**	**Algorithms**	**Task and Performance**
[1]	The Washington Post	Headline, body of article, users' log history	Additive attention + Seq2Seq	Headline generation Bleu = 9.62
[39]	CNN/Daily Mail [40], contains 312,085 news articles	Body of article, 4 bullet points	Additive attention + Seq2Seq	Headline generation ROUGE-1 = 40.06
[34]	New York Times	Headline, body of an article	Attention-based AMR encoder	Headline generation ROUGE-L = 28.54

	(annotated gigaword) [41]			
[3]	Valtteri system [42], wiki quote[9]	Headline, body of an article	Valtteri system [42] + phrase copying	Headline generation Human evaluation
[37]	PENS dataset	Headline, Clickstream	Transformer model and RNN	ROUGE scores

[9] https://en.wikiquote.org/wiki/English_proverbs_(alphabetically_by_proverb)

Chapter 3
Learning to Determine the Quality of News Headlines

3.1. Introduction

People's attitudes toward reading news articles are changing, with more people now being willing to read online articles rather than paper-based ones. In the past, individuals would purchase a newspaper and skim through the pages, focusing on the headlines and reading articles that caught their interest [43]. Headlines played a crucial role in providing readers with a clear understanding of the article's topic.

However, today, online news publishers are changing the role of headlines in a way that makes them the most important tool for gaining readers' attention. One significant reason for this shift is that online newspapers rely on the income generated from subscriptions and clicks made by their readers [19]. Additionally, publishers need to attract more readers than their competitors in order to succeed in this highly competitive industry.

Therefore, having a mechanism that can predict the quality of news headlines before publication can help authors choose headlines that not only increase readers' attention but also satisfy their expectations. Additionally, using high-quality headlines will help bring more traffic from social media platforms such as Twitter and Facebook to the news portal, thereby increasing revenue through both advertisements and subscriptions. However, determining the quality of headlines is not an easy task due to a lack of concise definition in previous studies. Some

previous works have considered quality as the level of interest generated among readers, measured by the number of clicks or likes on social media. Others have focused on accuracy, considering headlines that are closely related to the content of their articles as high quality. In our study, we define a high-quality headline as one that is both attractive and relevant to its article's content. To determine headline quality, we establish four quality indicators, which are described in section 3.2.1.

In general, previous studies can be classified into two categories: misleading headlines detection and headline popularity prediction. The former is typically approached as a binary text classification problem, where the objective is to determine whether a headline is misleading or not. Misleading headlines are generally considered to be of low quality, as they fail to meet the expectations of readers. However, not all non-misleading headlines are necessarily of high quality, as they may not be interesting enough to capture the attention of news readers.

On the other hand, the focus of studies in the headline popularity prediction category is to predict the popularity of news headlines, mostly in terms of the number of visits, shares, or likes by newsreaders. Almost all of these studies have considered headline popularity prediction as a regression problem. Unpopular headlines have been considered low-quality headlines since they fail to engage newsreaders. However, not all popular headlines are of high quality, as they may be misleading. Therefore, neither misleading headline detection nor popularity prediction models can determine high-quality headlines. In this work, we propose a novel approach for predicting headline quality that can distinguish high-quality headlines from low-quality ones.

The main contributions of this research are as follows:

- We propose a novel approach for determining the quality of published headlines using dwell time and click counts of the articles. We provide four headline quality indicators. With this approach, we can automatically label the quality of headlines in a news article dataset of any size, eliminating the need for human annotators. Using human annotators to label data is expensive and time-consuming, and it can lead to inconsistent labels due to subjectivity. To the best of our knowledge, no previous research has addressed headline quality detection in this manner.
- We develop a predictive model based on a deep neural network that incorporates advanced features to predict the quality of unpublished headlines. This model utilizes the headline quality indicators proposed earlier and takes into account the similarity between the headline and its corresponding article.

3.2. Proposed Headline Quality Indicators

In this section, a novel approach is introduced to calculate the quality of published headlines based on users' interactions with articles. This approach can help us to label the data automatically.

3.2.1. Quality Indicators

Due to the high cost of labeling data using human annotators, large datasets are not available for most NLP tasks [44]. In this section, we calculate the quality of published articles using click

count and dwell time measures. By using the proposed approach, we can automatically label any size of database and use those labels as ground truths to train deep learning models. The dwell time for article a is computed using formula (3-1).

$$D_a = \frac{\sum_u T_{a,u}}{C_a} \tag{3-1}$$

Where C_a is the number of times article a was read and $T_{a,u}$ is the total amount of time that user u has spent reading article a. Thus, the dwell time of article a (i.e., D_a) is the average amount of time spent on the article during a user visit. The values of read count and dwell time are normalized in the scale of zero to one.

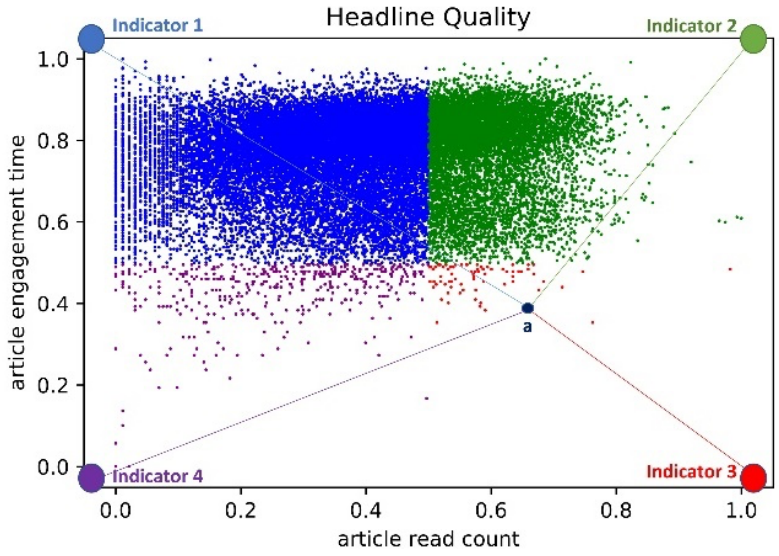

Figure 3-1: Representing News Headlines' quality with respect to the four quality indicators.

By considering these two measures for headline quality, we can define four quality indicators, which are shown by the four corners of the rectangle in Figure 3-1. We did not normalize articles' dwell time by articles' length since the correlation and mutual information

between articles' reading time and articles' length were 0.2 and 0.06, respectively. This indicates a very low dependency between these two variables in our dataset.

- **Indicator 1:** High dwell time but low read count. Articles with this indicator had high user engagement as users spent a significant amount of time reading them. However, the headlines of these articles were not compelling enough to encourage users to click on them.
- **Indicator 2:** High dwell time and high read count. Articles close to indicator 2 had interesting headlines, as they were opened by many users. Additionally, these articles were engaging due to their high dwell time.
- **Indicator 3:** Low dwell time but high read count. Articles close to this indicator have a high read count but a low dwell time. These headlines were interesting for users, but their articles were not. We call this type of headlines misleading since the articles do not meet the expectations of the readers. As we can see in Figure 3-1, very few articles reside in this group.
- **Indicator 4:** Low dwell time and low read count. The headlines of these articles were not successful in attracting users, and those who opened them did not spend much time reading them.

The probability that article a belongs to each quality indicator i (i.e. $P_{a,i}$) is calculated using formula (3-2) which $\| \|_2$ is the L_2 norm. Softmax function is used to convert the calculated similarities into probabilities.

$$\begin{bmatrix} P_{a,1} \\ P_{a,2} \\ P_{a,3} \\ P_{a,4} \end{bmatrix} = Softmax \begin{pmatrix} \sqrt{2} - \|(C_a, (1 - D_a))\|_2 \\ \sqrt{2} - \|((1 - C_a), (1 - D_a))\|_2 \\ \sqrt{2} - \|((1 - C_a), D_a)\|_2 \\ \sqrt{2} - \|(C_a, D_a)\|_2 \end{pmatrix} \quad (3\text{-}2)$$

3.3. Headline Quality Prediction

In this section, we propose a novel model to predict the quality of unpublished news headlines. To the best of our knowledge, we are the first to consider latent features of headlines, bodies, and the semantic relation between them in order to determine the quality of news headlines.

3.3.1. Problem Statement

We consider the task of headline quality prediction as a multiclass classification problem, where a class is a quality indicator defined in the last section. Given a dataset $D = \{(H_i, B_i, C_i)\}_i^N$ of N news articles where each news article contains a header H_i and an article body B_i and is labelled with a quality indicator C_i, the goal of training a headline quality prediction model from D is to find the model parameter θ that maximizes the probability of C_i given H_i and B_i for all the articles in D:

$$\theta = argmax_\theta \prod_i^N P(C_i \mid H_i, B_i, \theta)$$

3.3.2. A Proposed Headline Quality Prediction Model

In this section we propose a deep learning model to predict the quality of headlines before publication. The architecture of the proposed model is illustrated in Figure 3-2.

3.3.2.1. Embedding Layer

The embedding layer converts the one-hot encoding of each word in headlines and articles into dense word embedding vectors. We initialize the embedding vectors using GloVe embedding vectors as it shows superior performances specially in text classification task [45]–[47]. We found that 100-dimensional embedding vectors resulted in the best performance. Other studies have also opted to utilize 100-dimensional GloVe embeddings for their experiments, with some demonstrating superior performance with this dimensionality [48]–[51]

One possible reason for this could be that larger embedding dimensions are more prone to overfitting due to their increased capacity to capture subtle semantic relationships. They may capture more intricate semantic nuances in the training data, which could hinder generalization to unseen data.

3.3.2.2. Similarity Matrix Layer

Since one of the primary attributes of high-quality headlines is their relevance to the article's content, the primary objective of this layer is to ascertain the degree of correlation between each headline and the body of the article. The inputs to this layer are embedding vectors of the words from both the headline and the first paragraph of the article. We use the first paragraph of the

article because it is widely utilized in headline generation tasks, as it holds significant importance in representing the entire news article [52].

In Figure 3-2, each cell $c_{i,j}$ represents the similarity between words h_i and b_j from the headline and its article, respectively, which is calculated using the cosine similarity between their embedding vectors using formula (3-3).

$$C_{ij} = \frac{\vec{z_i}^{\,T}\,\vec{t_j}}{\|\vec{z_i}\|.\|\vec{t_j}\|} \tag{3-3}$$

Using the cosine similarity function will enable our model to capture the semantic relation between the embedding vectors of two words z_i and t_j in the article and header, respectively. Also, the 2-d similarity matrix allows us to use 2-d CNN which has shown great performance for text classification through abstracting visual patterns from text data [53]. Indeed, matching headlines and articles is regarded as an image recognition problem, and a 2-d CNN is employed to address it.

3.3.2.3. Convolution and Max-Pooling Layers

Three convolutional network layers, each of which contains 2-d CNN and 2-d Max-Pooling layers, are used on top of the similarity matrix layer. The entire similarity matrix is scanned by the first layer of 2-d CNN to generate the first feature map. Different levels of matching patterns are extracted from the similarity matrix in each convolutional network layer based on the formula (3-4).

$$x_{i,j}^{(1,k)} = f\left(\sum_{y=1}^{v_k}\left(\sum_{m=1}^{v_k} w_{y,m}^{(l+1,k)} \cdot x_{i+y,j+m}^{(0)}\right) + b^{(1,k)}\right) \quad (3\text{-}4)$$

Then, different levels of matching patterns are extracted from the Similarity Matrix in each Convolutional Network Layer based on the formula (3-5).

$$x_{i,j}^{(l+1,k)} = f\left(\sum_{p=1}^{n_l}\left(\sum_{y=1}^{v_k}\left(\sum_{m=1}^{v_k} w_{y,m}^{(l+1,k)} \cdot x_{i+y,j+m}^{(l,p)}\right) + b^{(l+1,k)}\right)\right) \quad (3\text{-}5)$$

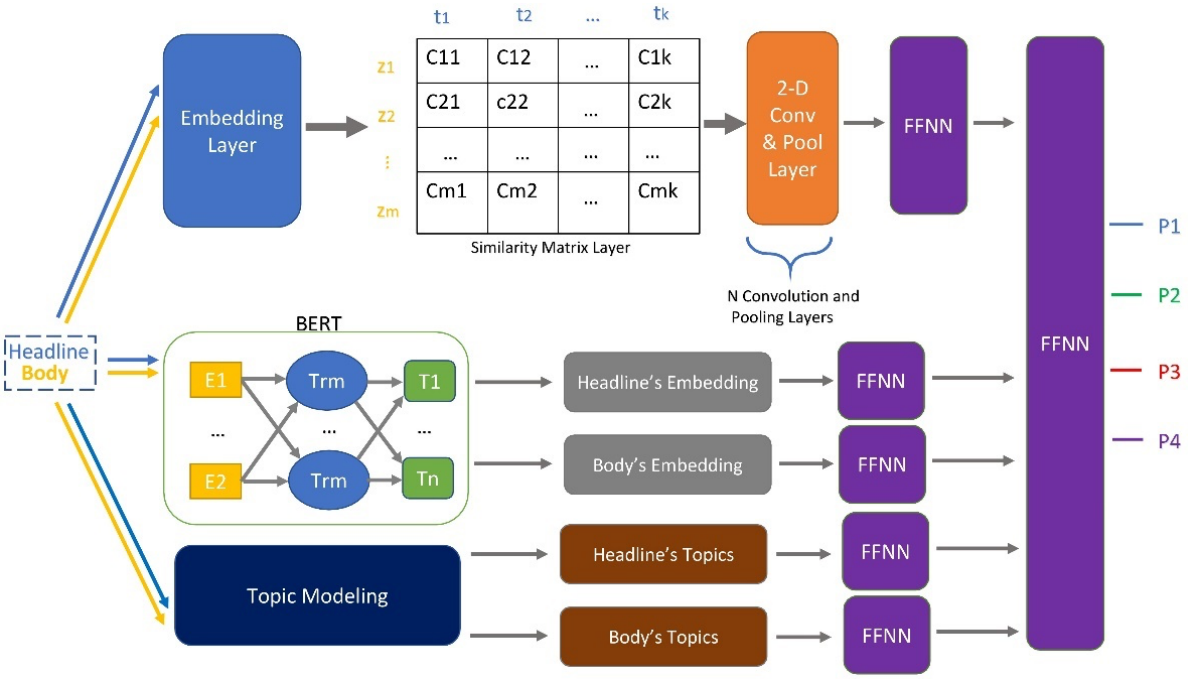

Figure 3-2 The proposed model for predicting news headlines' quality according to the four quality indicators.

Where $x^{(l+1)}$ is the computed feature map at level $l+1$, $w^{(l+1,k)}$ is the k-th square kernel at the level $l+1$ which scans the whole feature map $x^{(l)}$ from the previous layer, v_k is the size of

the kernel, $b^{(l+1)}$ are the bias parameters at level $l+1$, and ReLU [54] is chosen to be the activation function f.

3.3.2.4. BERT

Google's Bidirectional Encoder Representations from Transformers (BERT) [36] is used to convert variable-length inputs, such as headlines and articles, into fixed-length vectors. This is done to identify the underlying features of both headlines and articles. BERT aims to create a language model using the Transformer model, and further information on how the Google Transformer functions can be found in [55].

BERT is pre-trained on a large dataset to learn general knowledge that can be used and combined with acquired knowledge from a small dataset. We utilize the publicly available pre-trained BERT model (i.e., BERT-Base, Uncased)[10] published by Google. After encoding each headline into a fixed-length vector using BERT, a multi-layer perceptron is employed to project each encoded headline into a 100-dimensional vector. The same procedure is applied to the articles as well.

3.3.2.5. Topic Modeling

We use Non-Negative Matrix Factorization (NNMF) [56] and Latent Dirichlet Allocation (LDA) [57] from the Scikit-learn library to find topics from both headlines and articles. Since

[10] https://github.com/google-research/bert\#pre-trained models

headlines are significantly shorter than articles, we use separate topic models for headlines and articles. Even though both NNMF and LDA can be used for topic modeling, their approaches are totally different from each other. NNMF is based on linear algebra, while LDA relies on probabilistic graphical modeling. We find that NNMF extracts more meaningful topics than LDA on our news dataset.

We create matrix A, in which each article is represented as a row and columns are the TF-IDF values of the article's words. TF-IDF is an acronym for term frequency - inverse document frequency, which is a statistical measure that shows how important a word is to an article in a group of articles. The term frequency (TF) calculates how frequently a word appears in an article divided by the total number of words in that article. The inverse document frequency (IDF) weighs down the frequent words while scaling up the rare words in an entire corpus.

Then we use NNMF to factorize matrix *A* into two matrices *W* and *H* which are document to topic matrix and topic to word matrix, respectively. When these two matrices multiplied, the result is the matrix *A* with the lowest error (formula (3-6)).

$$A_{n \times v} = W_{n \times t} H_{t \times v} \tag{3-6}$$

In formula (3-6), *n* is the number of articles, *v* is the size of vocabulary, and *t* is the number of topics ($t \ll v$) which we set it to 50. As it is shown in Figure 3-2, we use topics (i.e., each rows of matrix *W*) as input features to the Feedforward Neural Network (FFNN) part of our model.

3.3.2.6. FFNN

As we can see in Figure 3-2, FFNN layers are used in different parts of our proposed model. The rectifier is used as the activation function for all layers except the last one. The activation function of the last layer is Softmax, which calculates the probability of the input example being in each quality indicator. We find that using a batch normalization layer before the activation layer in all layers helps to reduce the loss of our model. This is because a batch normalization layer normalizes the input to the activation function, ensuring that the data are centered in the linear part of the activation function.

3.3.3. **Results**

3.3.3.1. Baselines

For evaluation, we compared our proposed model with the following baseline models.

EMB + 1-d CNN + FFNN

The embedding layer is similar to the embedding layer of the proposed model, which converts the one-hot representation of words into dense 100-dimensional vectors. A dropout layer is applied on top of the embedding layer to randomly drop 20% of the output units. Additionally, GloVe embedding vectors are used to initialize the word embedding vectors [45].

The next layer is a 1-dimensional CNN, which is effective in identifying patterns within single spatial dimension data, such as text, time series, and signals. Many recent NLP models have utilized 1-dimensional CNNs for text classification tasks [58]. The architecture consists of two convolutional layers stacked on top of the embedding layer. The final layer is a single-layer feed-forward neural network (FFNN) that uses softmax as its activation function.

Doc2Vec + FFNN

Doc2Vec[11] is an implementation of the Paragraph Vector model, which was proposed by Le and Mikolov in [59]. It is an unsupervised learning algorithm that can learn fixed-length vector representations for different lengths of text, such as paragraphs and documents. The goal is to learn the paragraph vectors by predicting the surrounding words in contexts obtained from the paragraph. It consists of two different models: the Paragraph Vector Distributed Memory Model (PV-DMM) and the Paragraph Vector without word ordering Distributed Bag of Words (PV-DBOW). The former has much higher accuracy than the latter, but the combination of them yields the best result.

We convert headlines and bodies into two separate 100-dimensional embedded vectors. These vectors are fed into a Feedforward Neural Network (FFNN), which comprises two hidden layers with sizes of 200 and 50, respectively. ReLU is used as the activation function for all FFNN layers, except the last layer, which employs the softmax function.

EMB + BGRU + FFNN

This is a Bidirectional Gated Recurrent Unit on top of the Embedding layer. GRU employs two gates to trail the input sequences without using separate memory cells which are reset r_t and update z_t gates, respectively.

[11] https://radimrehurek.com/gensim/models/doc2vec.html

EMB + BLSTM + FFNN

This is a Bidirectional Long Short Term Memory (LSTM) [60] on top of the Embedding layer. Embedding and FFNN layers are similar to the previous baseline. The only difference here is using LSTM instead of using GRU.

3.3.4. **Evaluation Metrics**

Mean Absolute Error (MAE) and Relative Absolute Error (RAE) are used to compare the result of the proposed model with the result of the baseline models on test dataset. As we can see in formula (3-7), RAE is relative to a simple predictor, which is just the average of the ground truth. The ground truth, the predicted values and the average of the ground truth are shown by P_{ij}, $\hat{P}_{ij}$, and $\bar{P}_j$, respectively.

$$RAE = \frac{\sum_{i=1}^{N}\sum_{j=1}^{4}\left|P_{ij} - \hat{P}_{ij}\right|}{\sum_{i=1}^{N}\sum_{j=1}^{4}\left|P_{ij} - \bar{P}_j\right|} \times 100 \tag{3-7}$$

3.3.5. **Experiments**

Our data is provided by The Globe and Mail, a major Canadian newspaper. It includes a news corpus dataset, which contains articles and their metadata, and a log dataset, which contains interactions of readers with the news website. Whenever a reader opens an article, writes a comment, or performs any other trackable action, it is detected on the website and stored as a

record in a log data warehouse. Each record typically includes 246 captured attributes, such as event ID, user, time, date, browser, IP address, etc.

The log data can provide useful insights into readers' behaviors. However, there is noise and inconsistencies in the clickstream data that need to be cleaned before calculating any measures, applying any models, or extracting any patterns. For example, users may leave articles open in the browser for a long time while engaging in other activities, such as browsing other websites in another tab. In such cases, certain news articles may receive artificially high dwell times from certain readers.

The results of the proposed model and baseline models on the test data are shown in Table 3-1. We use the Adam optimizer for the proposed model and all our baseline models [61]. Additionally, we split our dataset into train, validation, and test sets using 70%, 10%, and 20% of the data, respectively.

Our proposed model achieved the best results with the lowest RAE compared to all other models. Surprisingly, TF-IDF outperformed the other baseline models. This could be attributed to the relatively small size of our dataset, which consists of only 28,751 articles. As a result, more complex baseline models may have been overfitting to the training data.

Also, we are interested in determining the significance of the latent features in relation to the semantic relationship between headlines and articles. Therefore, we have excluded the embedding, similarity matrix, and 2-d CNN layers from the proposed model. As a result of these modifications, the RAE has increased by 6 percent compared to the original proposed model. This indicates that evaluating the similarity between the headline and body of an article is advantageous for predicting headline quality.

Table 3-1 Comparison between the proposed model and baseline models

Models	MAE	RAE
EMB + 1-D CNN + FFNN	0.044	105.08
Doc2Vec + FFNN	0.043	101.61
EMB + BLSTM + FFNN	0.041	97.92
EMB + BGRU + FFNN	0.039	94.38
TF-IDF + FFNN	0.038	89.28
Proposed Model without Similarity Matrix	0.036	86.1
Proposed Model	**0.034**	**80.56**

3.4. Incorporating Sentiments in Headline Quality Prediction Model

3.4.1. Overview

The intuition here is that the sentiment of a headline may be strongly correlated with its popularity. Other studies have also shown that intensely positive or negative headlines tend to attract more clicks, while neutral ones receive fewer clicks [19]. To investigate the validity of our idea, we incorporate a new component into our existing headline quality predictor model. This component is a deep FFNN that has been pre-trained on sentiment data, using a feature-based transfer learning technique [8].

We use the Yelp Dataset to pretrain a deep FFNN and then utilize the pretrained model in our architecture. The dataset consists of 6,685,900 reviews, each with a star rating ranging from 1

to 5. Previous studies have demonstrated that transfer learning techniques are more effective when the source domain closely resembles the destination domain [38], [62]–[64]. Therefore, we curate our source dataset by following these steps:

- Firstly, we remove non-English samples.
- Secondly, we remove reviews that have special patterns (e.g., characters, emojis, etc.) which are not commonly used in news headlines.
- Last but not least, we remove short and lengthy reviews from our source domain.

Our task is ranking learning, as it combines characteristics of both classification and regression tasks. It is similar to a multi-class classification task, as we have five classes (i.e., stars) and the goal is to assign each sample to only one of these classes. However, it also has characteristics of regression, as there is an order between the classes ($5 > 4 > 3 > 2 > 1$), making the cost of misclassifying 1 with 5 much greater than 4 with 5. To address this, we follow the approach outlined in [65] and pre-train our neural network using the ranking learning technique. First, we convert our labels from a one-hot encoding format to the suggested format for ranking learning. Then, we change the activation function of the last layer from Softmax (typically used for multi-class classification) to the Sigmoid function. This modification enables the network to classify each sample into one or more classes. Therefore, the input of the model is BERT's semantic embedding, and the output can be one or more classes.

We modify the pre-trained neural network by removing its last layer before incorporating it into our headline quality architecture. The reason for removing the output layer is that it is too closely aligned with our target function during the pre-training phase, resulting in a bias towards the labels of the source domain. The pre-trained model enhances our headline quality model by

incorporating sentiment features. In this setup, the input to the model is the BERT embedding of news headlines, and the output is a 100-dimensional dense representation of sentiment features, which is then passed to the next layer of our headline quality prediction model. The modified architecture of the headline quality prediction model is depicted in Figure 3-3.

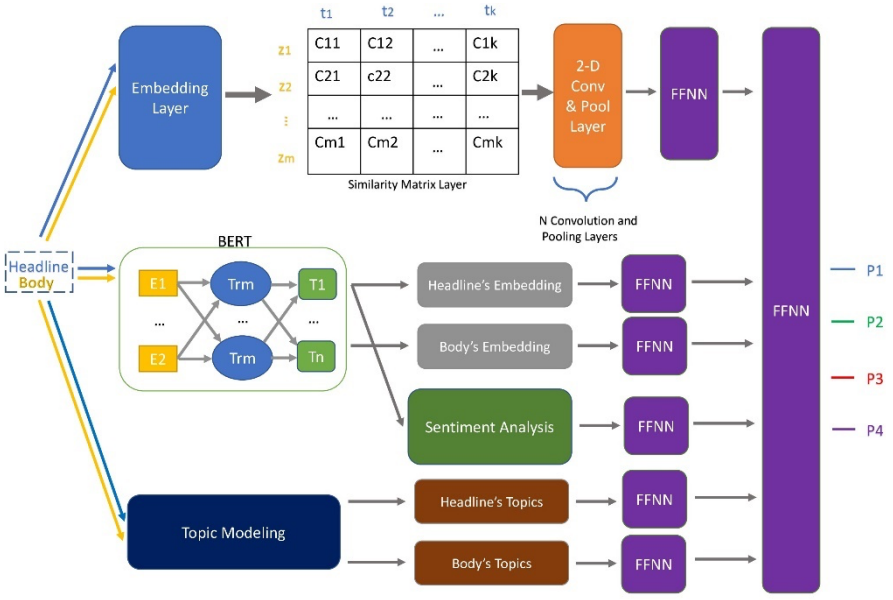

Figure 3-3 The modified version of headline quality prediction model by adding semantics.

We observed that incorporating sentiments into the headline quality prediction model resulted in improvements in the evaluation metrics used in Table 3-2. The result shows that using the pre-trained sentiment analysis component could help decrease the RAE by 2.

Table 3-2 A comparison between the two versions of headline quality prediction model with and without having the sentiment analysis capability

Models	MAE	RAE
Proposed model without using sentiment analysis	0.034	80.56
Proposed model with using sentiment analysis	0.030	78.23

3.5. Conclusion

We defined the quality of news headlines using four proposed quality indicators. Additionally, we proposed a method to calculate the quality of news headlines based on users' browsing history data and the four quality indicators. This approach allows news media to analyse the quality of their published headlines and create a labelled training dataset without relying on human evaluators. Our proposed labelling approach was warmly received by The Globe and Mail and subsequently has been adopted by others such as Kaleva Media.

Furthermore, we introduced a novel model that predicts the quality of headlines prior to their publication. This model combines sentiment, topics, latent features of headlines and articles, and their similarities. The experiment was conducted using a real dataset obtained from The Globe and Mail. The results demonstrated that our proposed model outperformed all the baselines in terms of Mean Absolute Error (MAE) and Relative Absolute Error (RAE) measures.

Our proposed approach enables authors to predict the popularity of unpublished headlines and determine their accuracy by matching them to the article content. Ultimately, it assists authors in selecting the best headline among other candidates for their unpublished articles.

Chapter 4
Generating High-Quality and Attractive Headlines Using Reinforcement Learning

4.1. Introduction

Headlines are crucial in news articles as they provide a concise summary of the entire article and capture people's attention. They can also drive web traffic in online news portals, bringing users from social media and increasing news media revenue. Headline generation is a subtask of text summarization, aiming to create high-quality headlines for news articles. There are two approaches: extractive and abstractive. In the extractive approach, the headline generator uses words or phrases directly from the article. In contrast, the abstractive approach may include words that are not present in the original document. Most recent generative methods, particularly those utilizing deep learning models, fall under the abstractive approach.

Although the goal of both summarization and headline generation is to produce a short, concise summary of the original document (i.e., a news article), there are some key differences that make the task of headline generation even more challenging.

First, in the headline generation task, the output summary should consist of only one sentence, whereas in text summarization, multiple sentences can be generated. This condition makes the headline generation task more challenging than the text summarization task especially for the extractive methods.

Second, the generated headline should be interesting to catch the readers' attention; otherwise, a good article may be ignored by readers. There are some works on headline generation [1]–[4], [37], [66]–[68] that employed generative models to generate headlines. However, to the best of our knowledge, none of the previous works considers the popularity of headlines when generating or evaluating the headlines.

Additionally, there is another challenge that applies to both summarization and headline generation tasks. Most generative models are trained using the cross-entropy loss function (word-level loss function) to maximize the probability of predicting the correct next word [35], [69]–[72]. However, two headlines can convey the same meaning while using different arrangements of words or phrases. For example, the following two headlines have identical meanings but differ in word order:

- "City Mayor Announces Bold Plan to Revive Economy"
- "Bold Plan to Revive Economy Announced by City Mayor"

The cross-entropy loss function does not account for such variations, as it penalizes any mismatch of words at corresponding positions between the generated headline and the ground truth. In contrast, similarity metrics such as METEOR [11] and BERTScore [12] account for synonyms, paraphrasing, or more complex linguistic phenomena, making them more suitable for evaluating such cases. However, scores such as METEOR and BERTScore cannot be used as a loss function in neural network training because they are not differentiable.

To address the aforementioned challenges in headline generation task, we utilize Reinforcement Learning (RL) based on self-critical sequence training (SCST) [73] to train a headline generator model. This approach allows us to not only directly optimize the model based

on non-differentiable functions (e.g., ROUGE-1) but also combine multiple evaluation metrics (e.g., using both a popularity score and ROUGE-1).

We investigate how using different similarity metrics as a reward function would work in headline generation task. We also introduce a popularity score and use it as metric in both evaluation and reward function. We calculate the popularity of headlines by considering the number of likes they receive on Twitter. Additionally, we fine-tune a BERT-based model [36] to predict the popularity of the generated headlines and incorporate this information into our proposed reward function.

Our main contributions are as follows:

- We employ different reward functions using similarity scores, such as ROUGE [10][12], METEOR [11], and BERTScore [12], and the popularity score to train a headline generator model using SCST in reinforcement learning. To the best of our knowledge, this is the first attempt to train a headline generator model based on RL and the popularity score.
- We propose a new evaluation metric to compare the attractiveness of headline generator models.

---

[12] The ROUGE score measures the lexical similarity between two pieces of text by capturing overlapping sequences in both texts. While it can provide a measure of textual similarity, it may not effectively reflect true semantic similarity, especially when different wording is used to convey the same meaning. However, the use of ROUGE as a reward function is conceptually better than using cross entropy as the loss function because cross entropy favors exact match, while ROUGE captures text overlapping. We include ROUGE in our method as an alternative design for the reward function for comparison purposes.

- We create a new Headline Popularity dataset from Twitter. Our dataset is not biased based on hidden factors from news portals.

4.2. Problem Formulation

Headline generation is a special type of text summarization. It involves creating concise, informative, and engaging headlines for news articles. The goal is to summarize the core content or main idea of a news article in a limited number of words, capturing the essence of the article while also being attention-catching. Most of the recently proposed headline generation models are based on neural networks [74]. Given a dataset $D = \{(H_i, B_i)\}_i^N$ of N news articles where each news article contains a header and an article body represented as H_i and B_i, respectively, the goal of training a headline generation model from D is to find the model parameter θ that maximizes the probability of H_i given B_i for all the articles in D:

$$\theta = argmax_\theta \prod_i^N P(H_i \mid B_i, \theta) \tag{4-1}$$

4.3. Methods

4.3.1. Proposed Framework

The framework for training the headline generators using RL is depicted in Figure 4-1. It consists of a headline generator model that takes an article body as input and generates a headline. The generated headline and the ground truth headline are then provided to the reward component,

which calculates the reward based on the specified functions. Finally, the calculated reward is used by the optimizer to optimize the parameters of the headline generator model.

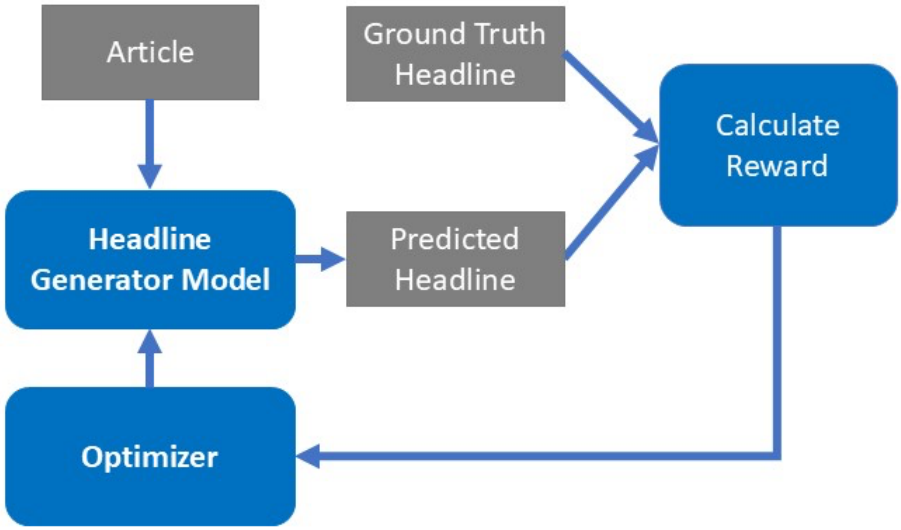

Figure 4-1 An RL-based headline generator model

4.3.2. **Model Parameters**

We use the model proposed by [9] for generating headlines, as shown in Figure 4-2. This model was selected for our headline generator for several key reasons. Firstly, it incorporates successful text generation techniques such as pointer generation. Additionally, it introduces a novel type of attention mechanism known as intra-decoder attention, which aims to reduce redundancy in the generated texts. Furthermore, the model demonstrated state-of-the-art results when trained using RL [9].

The first layer consists of word embedding vectors. On top of the embedding layer, the model incorporates a bidirectional LSTM in the encoder and an LSTM in the decoder, depicted in green and blue colors respectively in Figure 4-2. The LSTM block encompasses an update gate (i_t), a forget gate (f_t), an output gate (o_t), a hidden state (h_t), and a memory cell (c_t).

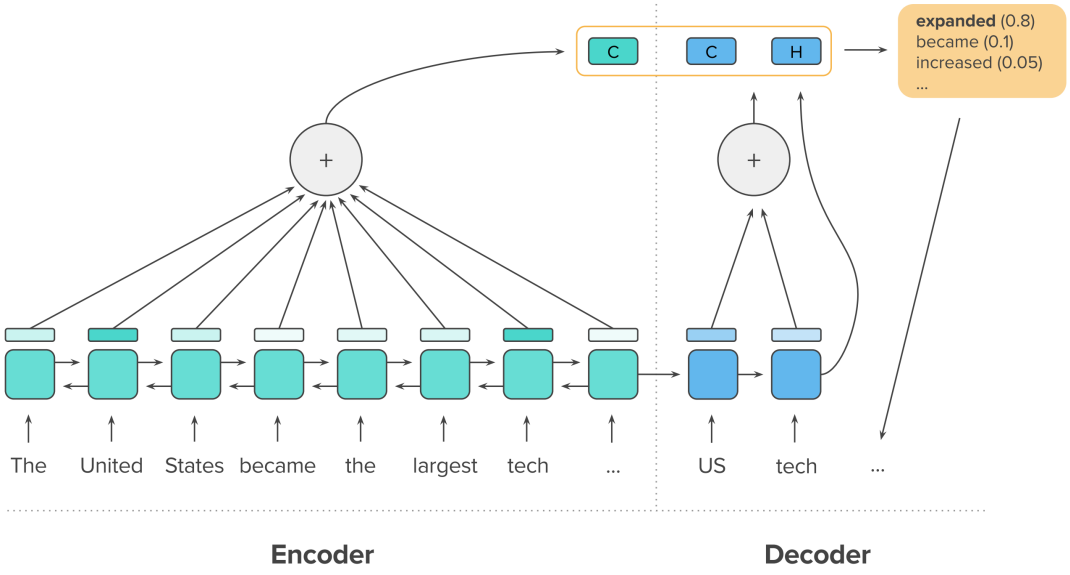

Figure 4-2 The architecture of the model used for the headline generation task [9]

$$i_t = sigmoid(W_i[x_t, h_{t-1}] + b_i) \tag{4-2}$$

$$f_t = sigmoid(W_f[x_t, h_{t-1}] + b_f) \tag{4-3}$$

$$\tilde{c}_t = tanh(W_c[x_t, h_{t-1}] + b_c) \tag{4-4}$$

$$c_t = i_t * \tilde{c}_t + f_t * c_{t-1}$$

(4-5)

$$o_t = sigmoid(W_o[x_t, h_{t-1}, c_{t-1}] + b_o) \tag{4-6}$$

$$h_t = o_t * \tan h(c_t) \tag{4-7}$$

On top of the bi-directional LSTM in the encoder part, the model contains Intra-Attention layer to enable the model to allocate different levels of attention to different parts of the input sequence. Additionally, it has been demonstrated in [35] that intra-attention can reduce repetitions, particularly when the input sequence is a long document.

The attention score e_{ti} is calculated using a bilinear function based on formula (4-8). In this formula h_t^d is the decoder hidden state at time t and h_i^e is the hidden input state of the encoder.

$$e_{ti} = h_t^{d^T} W_{attn}^e h_i^e \tag{4-8}$$

Then all the attention scores are normalized in a way that input tokens that have high attention scores in the past decoding steps would be penalized.

$$e'_{ti} = \begin{cases} \exp(e_{ti}) & if\ t = 1 \\ \dfrac{\exp(e_{ti})}{\sum_{j=1}^{t-1} \exp(e_{ji})} & otherwise \end{cases} \tag{4-9}$$

All the attention scores are then normalized across the inputs in formula (4-10) and used to compute the context vector c_t^e based on formula (4-11).

$$a_{ti}^e = \frac{e'_{ti}}{\sum_{j=1}^{n} e_{tj}} \tag{4-10}$$

$$c_t^e = \sum_{i=1}^{n} a_{ti}^e \, h_i^e \tag{4-11}$$

Moreover, the decoder part considers information about previously generated tokens by utilizing intra-decoder attention, which helps to reduce repetition. For each step *t*, the model computes a new context vector c_t^d. First, the attention scores are calculated using the bilinear function (formula (4-12)). Then, the attention scores will be normalized according to formula (4-13), and finally, the decoder context vector c_t^d is calculated based on formula (4-14).

$$e_{tt'}^d = h_t^{d^T} W_{attn}^e h_{t'}^d \tag{4-12}$$

$$a_{tt'}^d = \frac{exp\left(e_{tt'}^d\right)}{\sum_{j=1}^{t-1} exp\left(e_{tj}^d\right)} \tag{4-13}$$

$$c_t^d = \sum_{j=1}^{t-1} a_{tj}^d \, h_j^d \tag{4-14}$$

Furthermore, to handle unknown words, especially in articles with expert jargon (e.g., the name of a new virus), the model [9] employs a switch function to determine whether to use a generated token or a word directly from the input sequence using a pointer mechanism [75]. The switch u_t is the binary value calculated by formula (4-15). In this formula, $\|$ and σ are the concatenation operator and sigmoid function respectively.

$$p(u_t) = \sigma\left(W_u \left[h_t^d \parallel c_t^e \parallel c_t^d\right] + b_u\right) \tag{4-15}$$

If the value of the switch is zero, the next word would be predicted based on the formula (4-16). Otherwise, the pointer mechanism uses the temporal attention weights a_{ti}^e as the probability distribution to copy the input token x_i.

$$p(y_t \mid u_t = 0) = softmax\left(W_{out} \left[h_t^d \parallel c_t^e \parallel c_t^d\right] + b_{out}\right) \tag{4-16}$$

4.3.3. **Learning Objective**

We optimize the parameters of the headline generator model within our framework using Self-critical Sequence Training (SCST) [73]. SCST is a policy gradient algorithm, a class of reinforcement learning (RL) methods designed to learn policies that maximize expected rewards. The policy gradient approach typically involves the following steps:

- Sampling Actions: Generating actions based on probabilities determined by the current policy.
- Evaluating Rewards: Calculating the reward associated with the actions taken.
- Gradient Ascent: Updating the policy parameters to increase the likelihood of actions that yield higher rewards.

In policy gradient methods, the baseline might be used to improve the efficiency of training by reducing the variance in gradient estimates. In essence, the baseline serves as a reference point against which the rewards obtained by the agent are compared. By subtracting this baseline from the actual returns, the resulting advantage estimates focus more on the relative merit of

actions taken by the agent, rather than just the absolute rewards. SCST is a type of policy gradient method that uses a baseline in computing a reward. What differentiates SCST from other methods is that it employs the model itself in the inference mode to calculate the baseline reward. Experimentally, it was found in [73] that SCST could have lower variance and be more effectively trained.

In the headline generation task using the SCST method, the agent is a deep neural language generative model, which takes an action w_t^s (i.e., select a word from a vocabulary) in each timestep t based on the policy P_θ (parameters of the model) in the environment (i.e., the context vector and the words that are the inputs to the agent at each timestep). When a model reaches to the end of the sequence, it will observe a reward $r(w_1^s, \dots, w_T^s)$. In the context of the headline generation, the reward would be calculated by comparing the generated headline with the ground truth using an evaluation metric.

The goal of the training is to set the parameters of the model in a way that maximizes the reward (formula (4-17)) on the training set. The goal is to maximize the reward, so if we multiply -1 by the reward, the goal would be to minimize the negative reward function (formula (4-18)) that could be considered as a loss function.

$$maximize\ \mathbb{E}_{[w_1^s,\dots,w_T^s]\sim P_\theta}[r(w_1^s, \dots, w_T^s)] \tag{4-17}$$

$$minimize\ -\mathbb{E}_{[w_1^s,\dots,w_T^s]\sim P_\theta}[r(w_1^s, \dots, w_T^s)] = L(\theta) \tag{4-18}$$

By following the steps explained in [73], the expected gradient can be approximated using (4-19).

$$\begin{aligned}\nabla_\theta L(\theta) = & -E_{[w_1^s,\ \ldots, w_T^s]\sim P_\theta}[r(w_1^s,\ \ldots, w_T^s)\ \nabla_\theta\ \log P_\theta(w_1^s,\ \ldots, w_T^s)] \\ & \approx \mathrm{r}(w_1^s,\ \ldots, w_T^s)\ \nabla_\theta\ \log P_\theta(w_1^s,\ \ldots, w_T^s) \\ & \approx \left(r(w_1^s,\ \ldots, w_T^s) - r_b\left(w_1^g,\ \ldots, w_T^g\right)\right)\ \nabla_\theta\ \log P_\theta(w_1^s,\ \ldots, w_T^s)\end{aligned} \tag{4-19}$$

In policy gradient algorithms, the baseline could be any function if it does not vary with the action w^s. In policy gradient with baselines, it is shown [73] that the baseline $r_b\left(w_1^g, \ldots, w_T^g\right)$ can reduce the variance of the gradient while it does not change the expected gradient. By using a baseline, actions that yield a sampling reward greater than the baseline reward will be encouraged, and vice versa. In self-critical sequence training (SCST) [73] the baseline $r_b\left(w_1^g, \ldots, w_T^g\right)$ is the reward obtained using a greedy approach by the model under the test time mode. The gradient of the model could be calculated using a chain rule according to the formula (4-20):

$$\nabla_\theta L(\theta) = \frac{\partial L(\theta)}{\partial \theta} = \sum_{t=1}^{T} \frac{L(\theta)}{\partial s_t}\ \frac{\partial s_t}{\partial \theta} \tag{4-20}$$

In formula (4-20), s_t is the input to the softmax function. The gradient of the loss with respect to the softmax is estimated as formula (4-21) [73]. Formula (4-21) indicates that the selected word w_t^s serves as a surrogate ground-truth for the output distribution.

$$\frac{\partial L(\theta)}{\partial s_t} = \left(r(w_1^s, \ldots, w_T^s) - r_b\left(w_1^g, \ldots, w_T^g\right)\right)\ \left(P_\theta(w_t \mid w_{t-1}^s, h_t, c_{t-1}) - 1_{w_t^s}\right) \tag{4-21}$$

The formula (4-21) is nearly identical to the gradient of a multiclass logistic regression classifier. In logistic regression, the gradient represents the difference between the prediction and the actual 1-of-N representation (i.e., $1_{w_t^s}$) of the target word (formula (4-22)).

$$\frac{\partial L(\theta)}{\partial s_t} = \left(P_\theta(w_t \mid w_{t-1}, h_t, c_{t-1}) - 1_{w_t}\right) \tag{4-22}$$

Hence, we use cross-entropy loss to train our headline generator model similar to equation 18 in [76].

$$L(\theta) = \left(r(w_1^s, \dots, w_T^s) - r_b\left(w_1^g, \dots, w_T^g\right)\right) \sum_{t=1}^{T} \log\left(P_\theta(w_t^s | B, w_1^s, w_2^s, \dots, w_{t-1}^s)\right) \tag{4-23}$$

Where $r(w_1^s, \dots, w_T^s)$ and $r_b\left(w_1^g, \dots, w_T^g\right)$ are reward functions. We use ROUGE, METEOR, BERTScore, and the popularity score as reward functions, with descriptions of these functions provided in subsection 4.5. We discovered that training our model solely with the RL loss function (formula (4-23)) results in slow convergence. This is because the model needs to wait until the end of the sequence to evaluate its performance. Additionally, considering the size of the action space is 50k (i.e., the size of our vocabulary), for an average sequence length of 10, the total number of ways to end the sequence is $50K^{10}$. Moreover, since the parameters of the headline generator model are initialized with random values, the model performs many random actions at the beginning, leading to poor results. Therefore, to improve training, we follow [9] by using the teacher forcing technique, defining the maximum log-likelihood (ML) loss function G as formula (4-24) where B is the body of the article and w_1, w_2, …, w_T are the tokens of the ground truth headline.

$$G(\theta) = -\sum_{t=1}^{T} \log\left(P_\theta(w_t | B, w_1, w_2, \dots, w_{t-1})\right) \tag{4-24}$$

Following the approach outlined in [9], we define the loss function as the weighted combination of equations (4-23) and (4-24) using the parameter β. This parameter β enables the

adjustment of the importance of equations (4-23) and (4-24) in calculating the final loss, allowing for higher significance to be allocated to each of these functions.

$$K(\theta) = (\beta)L + (1 - \beta)G \tag{4-25}$$

Although the headline generator model can be trained using the loss function (4-25), the training process is computationally expensive since three separate headlines are generated for each article. First, a headline is generated using the teacher forcing technique to compute the cross-entropy loss function (4-24). Second, another headline is generated based on the sampling method for the SCST approach. In both cases, the model operates in training mode, meaning that gradients are computed during the loss calculation. Since SCST requires a baseline for comparison, a third headline is generated using a greedy approach, where the model runs in inference mode. Additionally, the reward function introduces further computational overhead, particularly when the evaluation metric (e.g., a popularity score) is computed using another deep learning model.

4.4. Experiment

4.4.1. Datasets

4.4.1.1. Newsroom

To train the headline generator models, we utilize the Newsroom dataset [77], which comprises 1.3 million news articles from 38 prominent news media outlets spanning the years from 1999 to 2017 (Figure 4-3).

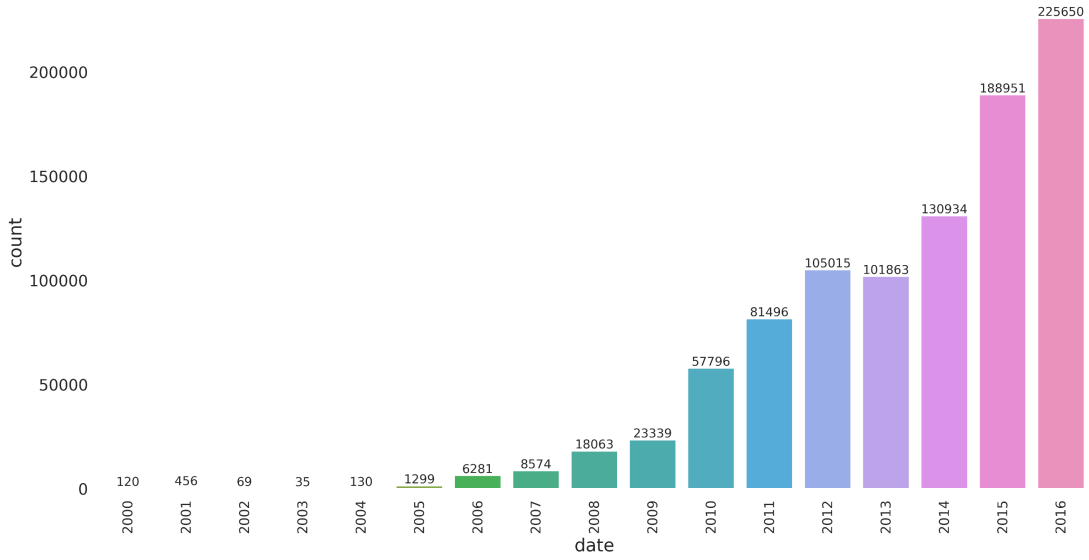

Figure 4-3 A number of articles per year in NewsRoom dataset

We decided to use the Newsroom dataset rather than The Globe and Mail dataset from Chapter 3, as the Newsroom dataset is much larger, containing 1.3 million articles from major news media outlets. This larger dataset is better suited for training our headline generator models. Additionally, unlike The Globe and Mail dataset, the Newsroom dataset is publicly available, which ensures that our research results can be reproduced by others.

Nearly all previous studies that have utilized this dataset have focused on using the main body and summaries of news articles. This study marks the first time that the headlines from this dataset have been utilized. We cleaned the dataset by removing articles with short headlines (less than 4 words) or long headlines (more than 20 words). We found that most headlines with less than 4 words were just the name of the editor or author of the article. Also, we removed long headlines from our data because we observed that most of them were not the actual headlines of the articles but rather summaries. We also removed short articles (less than 40

words) or long articles (more than 1000 words). We found that most short articles were not the full version of the article or contained only URL addresses. Similarly, articles longer than 1000 words were rare in our dataset. After implementing these changes, the dataset size was reduced from 1,212,740 to 950,071.

Table 4-1 The size of the Newsroom dataset is compared before and after data cleaning.

Data splits	**Original**	**Processed**
Train	995041	779205
Validation	108837	85480
Test	108862	85386

The distributions of articles by title length and article length for the unprocessed and processed versions of Newsroom are illustrated in Figure 4-4 and Figure 4-5, respectively. As shown in Figure 4-4, there are few articles with long titles and long articles. After processing, as illustrated in Figure 4-5, the majority of data (between quantiles one and three) has a title length between 7 and 11, with an average of 9.1 words, and an article length between 237 and 674, with an average of 460.3 words.

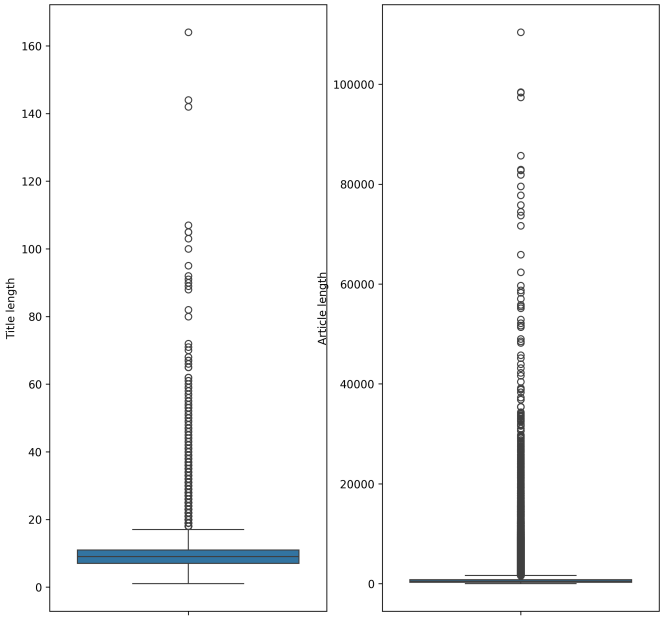

Figure 4-4 The distribution of articles in the Newsroom dataset based on the article length (right boxplot) and title length (left boxplot).

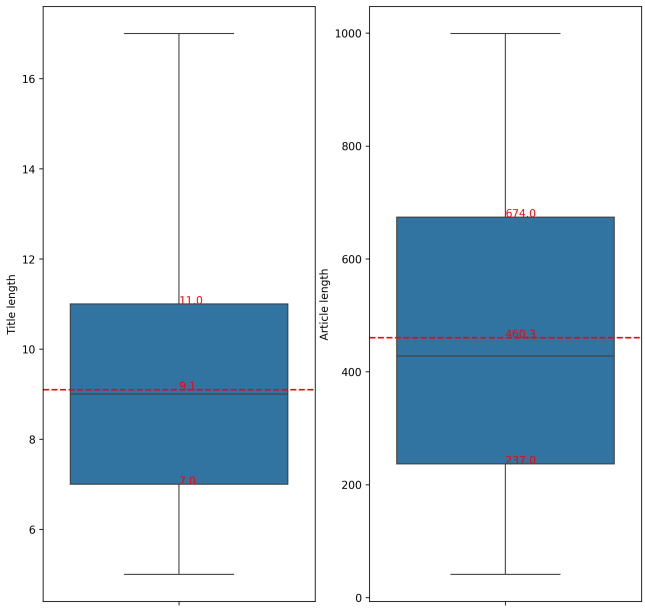

Figure 4-5 The distribution of articles in the processed Newsroom dataset based on the article length (right boxplot) and title length (left boxplot)

4.4.1.2. Headline Popularity Dataset

Most of the existing headline popularity datasets [5], [6], [78]–[83] used users' log history on a news portal to calculate the popularity of headlines using measures such as the number of clicks, views, etc. However, the view or click count of headlines does not depend solely on the attractiveness of the headlines themselves. It can also be influenced by other factors that are often not included in existing headline popularity datasets, such as the manner and duration each headline was displayed on the main page, the position of the headline on the page, or how frequently each headline was recommended to users by the news media's recommendation engine.

To eliminate the influence of the factors mentioned above, we define the popularity of the headlines based on the number of likes they received on Twitter. Nearly all news media outlets have established Twitter accounts through which they disseminate most of their headlines. We utilized tweet IDs from a dataset[13] provided by the University of Weimar and employed the Twitter API to gather social media statistics. The crawled information regarding the tweets is shown in Table 4-2.

The distribution of the data by the posted date is shown in Table 4-3. As we can see, most of the news in our dataset was published in 2016. Also, the distribution of the published news by the news media is depicted in Figure 4-6. We consider the number of likes to determine the popularity of each headline.

[13] https://zenodo.org/record/5530410

Table 4-2 A description of the data fields of the HP dataset.

Field	**Description**
ID	a unique Tweet identification number
Title	a posted headline on Twitter
Body	a news article
favorite count	a number of likes
retweet count	a number of retweets
created at	a timestamp of a Tweet
users followers count	a number of followers of a news media's Twitter account
news media	a news media's username
news media friends count	a number of Twitter accounts a news media is following

Table 4-3 Number of published articles in HP dataset by year.

Year	**Published Year**
2015	2428
2016	83930
2017	15535

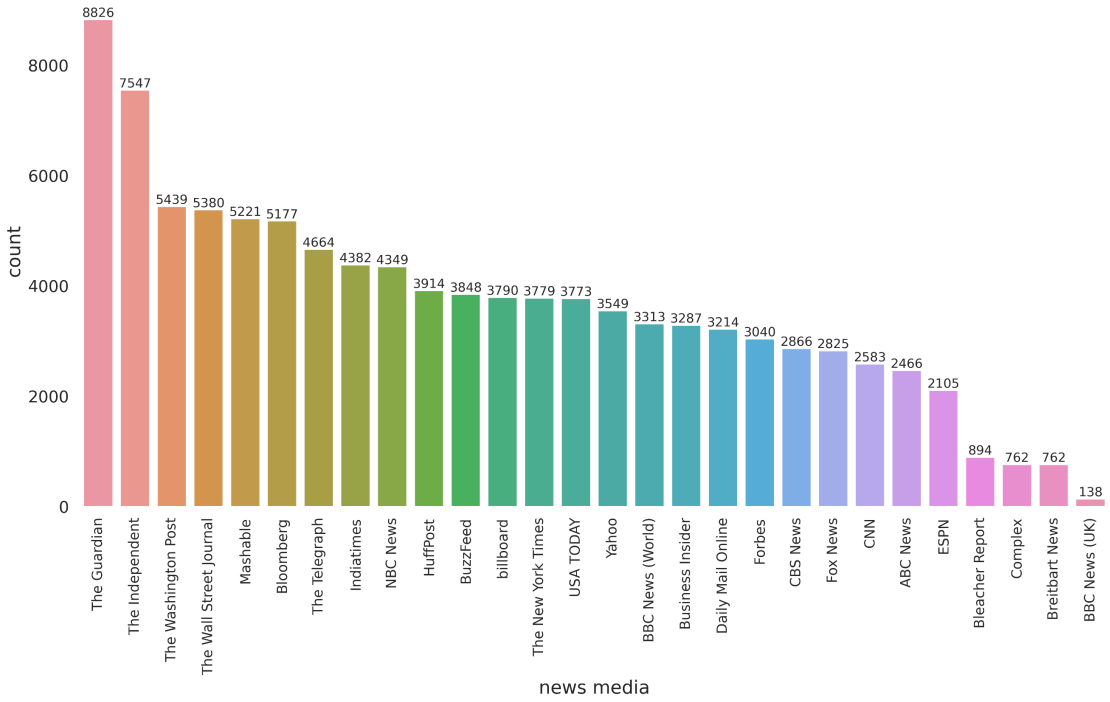

Figure 4-6 Number of headlines per News Media Twitter account in the HP dataset.

4.4.2. **Headline Generator Models**

We train different headline generator models on the training set of the Newsroom dataset. The first set of models is based on the headline generation architecture proposed by [9] that is explained in subsection 4.3.2. The number of learnable parameters of these models is 48,234,833. We choose the Adam optimizer [61] and the batch size of 200 to train the headline generator model. We train the headline generator model based on the loss function defined in subsection 4.3.3 for 15 epochs on the training set. We observed that using a high value for β (e.g., 0.9984, as used in [9]) the model fails to converge. This occurs because the majority of the

loss is dominated by the RL component, and given the large number of possible actions, the model struggles to achieve convergence. For the models trained using formula (4-25), we set the value of the parameter β to 0.5. We leave further investigation into determining the optimal value of the parameter β for each reward function to future work.

Depending on the choices, such as the function used for the reward calculation, we name these models as shown in Table 4-4.

Table 4-4 The specification of the RL based headline generator models

Model name	Loss function	Reward function
G	Formula (4-24)	No reward function
G-L-B	Formula (4-25)	BERTScore
G-L-M	Formula (4-25)	Meteor
G-L-P	Formula (4-25)	Popularity Score
G-L-P-R1	Formula (4-25)	Popularity Score + ROUGE-1
G-L-R1	Formula (4-25)	ROUGE-1
G-L-RL	Formula (4-25)	ROUGE-L

For G-L-P-R1, where both the Popularity score and ROUGE-1 are used in the reward function, we combine these two scores using formula (4-26). The value of α in formula (4-26) is set to 0.5. Further investigation into optimizing this value is left for future work.

$$r = \alpha ROUGE_1 + (1 - \alpha)P \tag{4-26}$$

The rest of the headline generator models are as following:

T5-finetuned: We define a new task to fine-tune T5 on the entire training set and call it T5-finetuned. To fine-tune the model for a new task, we append "headline:" to the beginning of each input article. We train T5 using the cross-entropy loss function and the teacher forcing technique.

T5: We are interested in evaluating the performance of the T5 model without applying fine-tuning. In this case, we prepend 'summarize:' to the beginning of each article and utilize the T5 model pre-trained for the summarization task.

OpenNMT[14]**:** We use the PyTorch version of the sequential model from the open-source (MIT) neural machine translation framework (OpenNMT) [33]. The model uses LSTMs for both the encoder and decoder parts.

4.5. Evaluation Metrics

We use ROUGE [10], BLEU [84], METEOR [11], BERTScore [12], and the popularity score to evaluate and compare the performance of the headline generator models. A description of each of these metrics is as follows:

ROUGE scores:

[14] https://github.com/OpenNMT/OpenNMT-py

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics commonly used for the evaluation of automatic summarization and machine translation systems by calculating the overlap between the generated text with the ground truth examples. The most commonly used ROUGE metrics are ROUGE-1, ROUGE-2, and ROUGE-L. ROUGE-1 is a metric evaluates the overlap of unigrams (individual words) between the generated text and the reference summary. ROUGE-1 is useful for capturing the basic content coverage by comparing how many words in the reference text are present in the generated summary. ROUGE-2 measures the overlap of bigrams (two consecutive words) between the generated text and the reference summary. ROUGE-L is another variant of the ROUGE metric that specifically focuses on the longest common subsequence (LCS) between a system-generated output and a reference text. We refer to ROUGE-1, ROUGE-2, and ROUGE-L as R-1, R-2 and R-L, respectively,

BLEU scores:

BLEU is a set of four (i.e., BLEU1, BLEU2, BLEU3, and BLEU4) automatic evaluation metrics primarily used to measure the quality of machine-generated texts by comparing them with one or more reference texts. Depending on which measure is used, BLEU calculated how many n-grams from the generated text appear in the reference text. BLEU is a precision-based metric because it focuses on what proportion of n-grams from the generated text exists in the reference text. We refer to BLEU1, BLEU2, BLEU3, and BLEU4 as B-1, B-2, B-3 and B-4, respectively.

METEOR:

METEOR (Metric for Evaluation of Translation with Explicit ORdering) is an automatic evaluation metric designed to compare generated texts with reference (ground truth) texts. Unlike BLEU, METEOR emphasizes recall by using a weighted harmonic mean of recall and

precision, where recall is weighted 9 times more than precision. Additionally, METEOR includes a fragmentation penalty to discourage disjointed matches. If matching words between the generated text and the reference appear in a fragmented or disordered manner, the score is penalized accordingly.

BERTScore:

BERTScore [12] is an automatic evaluation metric for assessing text similarity. In the context of our data, it aims to compare the generated responses with the ground truth data and indicate the quality of the generated responses in terms of Precision, Recall, and F-measure. Higher values of these metrics indicate that the generated responses are more semantically similar to the ground truth responses. In BERTScore, rather than focusing on the exact match of tokens between the generated responses and the ground truth, it calculates the cosine similarity score between the embedding vectors of each token in the generated response and the embedding vectors of each token in the ground truth. It then employs a greedy matching approach, where each token in a generated response is matched to the most similar token in the ground truth. It is worth noting that the embedding vectors for both generated responses and the ground truth are obtained from the embedding vectors generated for each token from the last hidden layer of the transformer encoder model.

Popularity Score:

We use our popularity dataset, described in Section 4.4.1.2, to train a transformer model [36] for a regression task. The input to the transformer model is a headline, and the output is a score between 0 and 1, where a score closer to 1 indicates higher attractiveness of the headline. The trained transformer model is then used to assign scores to the headlines generated by the

headline generator, with the obtained score serving as the reward during training and a metric in evaluation.

4.6. Evaluation Results

In this section, we compare the results of the headline generator models in terms of both automatic evaluation metrics, such as ROUGE, BLEU, and METEOR, as well as qualitative metrics from human points of view. For sequential models, the <start> and <eos> tokens are removed from the data to eliminate their impact on the evaluation metrics. Additionally, for the ROUGE score, we calculate the metric without applying stemming to the predicted and reference headlines, and we report the F-score for the ROUGE metric.

Table 4-5 A comparison between the headline generator models on the test set.

	R-1	R-2	R-L	Meteor	B-1	B-2	B-3	B-4	BERT-P	BERT-R	BERT_F1	P
G-L-M	0.248	0.101	0.233	0.195	0.473	0.144	0.068	**0.033**	0.866	0.854	0.86	0.502
G	0.251	0.101	0.235	0.194	**0.483**	**0.145**	**0.069**	0.032	0.867	0.852	0.859	0.501
G-L-B	0.247	0.1	0.228	0.211	0.408	0.121	0.057	0.027	0.862	0.858	0.86	0.477
G-L-P	0.232	0.092	0.219	0.177	0.473	0.133	0.061	0.026	0.86	0.847	0.854	0.507
G-L-P_R1	0.227	0.087	0.213	0.174	0.457	0.126	0.059	0.028	0.863	0.849	0.855	0.488
G-L-RL	0.235	0.095	0.221	0.186	0.453	0.13	0.062	0.029	0.862	0.85	0.856	0.509
G-L-R1	0.244	0.099	0.229	0.192	0.467	0.141	0.067	0.032	0.864	0.853	0.858	0.503
T5-finetuned	**0.313**	**0.139**	**0.286**	**0.262**	0.246	0.101	0.046	0.027	**0.882**	**0.879**	**0.88**	**0.639**
T5	0.161	0.045	0.144	0.14	0.079	0.016	0.005	0.002	0.85	0.851	0.85	0.518
OpenNMT	0.118	0.031	0.113	0.065	0.348	0.04	0.013	0.007	0.826	0.815	0.82	0.151

The results of the evaluation of the headline generator models on the test set are shown in Table 4-5. The T5-finetuned achieved better results than other models in terms of R-1, R-2, R-L, METEOR, BERTScore (i.e., BERT-P, BERT-R, and BERT-F1) and popularity score. This could be attributed to the fact that T5-finetuned is based on the transformer architecture (i.e., T5

model) and has a better capability to learn from the data. Additionally, unlike the other models which are trained from scratch on the Newsroom dataset, T5-finetuned is already pre-trained on a vast amount of data during its pre-training phase, which could enhance its performance on downstream tasks. Also, the transformer architecture helps the model learn better dependencies among the input sequences while LSTM based models tend to struggle with longer sequences in complex tasks like summarization.

Although transformer models are pre-trained on extensive amounts of data, a comparison between the results of T5 and T5-finetuned highlights significant performance improvements when the T5 transformer model is fine-tuned for the headline generation task. Furthermore, our findings demonstrate that a model trained specifically for summarization (T5) does not perform effectively in the context of headline generation. One of the main reasons is that summarization models often produce longer summaries than the headline, resulting in lower precision (lower BLEU score).

The results of the different variants of the headline generator model (subsection 4.4.2) show that the performance of the models is quite similar. However, among the models trained using reinforcement learning, those that utilized METEOR and BERTScore as reward functions demonstrated better performance, indicating that these metrics are more effective than ROUGE and the popularity scores. Additionally, training using only reinforcement learning proved to be unsuccessful. This is due to the large number of possible actions, which makes it highly unlikely for the model to find a successful path to the end with a reasonable reward value.

The similarity in performance among the G model (trained using only cross-entropy loss) and the different variants of the G-L models (e.g., G-L-B, G-L-M, G-L-RL, G-L-R1) can be attributed to the model reaching its learning capacity. Since all these models share the same

underlying LSTM-based architecture, their ability to improve further is constrained by the representational power of the architecture. While the G-L variants employ reinforcement learning with different reward functions to optimize specific metrics, these enhancements do not fundamentally alter the model's overall capacity for learning. Consequently, the differences in performance between G and the G-L variants are marginal, highlighting the architectural limitations in achieving substantial improvements.

The architecture of all the models in Table 4-4 is based on LSTM. Despite incorporating attention mechanisms in both the encoder and decoder, these models cannot match the performance of transformer-based architectures like T5-finetuned. Transformers are highly effective at capturing long-range dependencies and enabling efficient parallel processing, leveraging their self-attention mechanism, which outperforms the sequential processing approach of LSTMs. This architectural limitation explains why LSTM-based models, even when optimized with various reward functions, consistently underperform compared to transformer-based models like T5-finetuned in headline generation tasks.

Training sequential models like LSTMs is significantly slower compared to transformer-based models. This is because LSTMs process input sequences token by token, which inherently limits their ability to leverage parallel computation. In contrast, transformer models utilize self-attention mechanisms that allow computations to be performed in parallel across all tokens in a sequence. This parallelism not only accelerates the training process but also enables transformers to handle longer sequences more efficiently, making them a more practical choice for tasks like headline generation.

We conducted a human assessment on 50 articles using Amazon Mechanical Turk (MTurk), with five evaluators assigned to each article. The average rating from the evaluators

was used to determine the overall score for each article. Each article, along with its model-generated headlines, was shown to the evaluators. They rated each headline on a scale from 1 to 10, with 1 being the lowest and 10 being the highest score. The ratings were based on three criteria: conciseness, attractiveness, and readability. We do not have demographic information, such as sex, age, etc., about our human evaluators, as Amazon Mechanical Turk does not provide these details about the labelers.

The findings from this experiment are displayed in Table 4-6. Among all the models, T5_finetuned significantly outperformed other models, achieving the highest scores in terms of conciseness, attractiveness, and readability. This suggests that the model excels at generating headlines that are not only concise but also engaging and easy to read, making it the most suitable choice for headline generation among the other models used. One the other hand, the T5 model performed reasonably well, achieving scores that positioned it above most of the "G" family models but noticeably below its fine-tuned counterpart, T5_finetuned. These results seconds that while T5's base model is good in summarization, it benefits significantly from fine-tuning, which helps it better align with the headline generation task.

The "G" family of models showed moderate performance overall, with G-L-B standing out as the best among them. These improvements suggest that the modifications in the G-L-B variant contribute positively to generating more effective headlines. However, their scores were not competitive with T5_finetuned, highlighting a gap in overall effectiveness. OpenNMT emerged as the weakest performer across all criteria, with scores significantly lower than the other models.

Table 4-6 The human evaluation result for the headline generator models

Model Name	conciseness	attractiveness	readability
G	4.008	4.888	5.304
G-L-B	4.288	5.328	5.716
G-L-M	4.204	5.172	5.568
G-L-P	3.76	4.848	5.184
G-L-P-R1	3.848	4.788	5.148
G-L-R1	4	5.04	5.304
G-L-RL	4.024	4.98	5.356
OpenNMT	2.656	3.66	3.928
T5	4.064	5.2	5.128
T5_finetuned	**5.84**	**6.544**	**7.212**

Overall, the results underline the importance of fine-tuning and model design in achieving optimal performance. T5_finetuned's dominance across all metrics demonstrates the effectiveness of both tailored training and the power of the transformer architecture.

Some sample headlines generated by the models are as follows, with the articles selected from the Newsroom dataset.

Article 1:

"Former New Orleans Saints Player Will Smith Shot Dead After Traffic Accident"

"NEW ORLEANS—Former New Orleans Saints player Will Smith was shot and killed in what appeared to be a case of road rage by a man who had rear-ended Mr. Smith's car, city police said Sunday. There is no indication that Mr. Smith, 34, and the man accused of killing him knew each other, Police Superintendent Michael Harrison said Sunday afternoon.

Investigators don't have any evidence so far that the shooting was "anything other than this accident…that turned violent," Mr. Harrison said. Cordell Hayes, 28, waited for police to show up and was arrested on a charge of second-degree murder in Mr. Smith's death, Mr. Harrison said. Mr. Harrison said the former defensive end had dined just hours before the Saturday night crash with a police officer who had once been sued by Mr. Hayes, whose father was shot and killed by police in 2005. However, it wasn't clear if that played any role in the confrontation. Mr. Hayes's Humvee H2 rear-ended Mr. Smith's Mercedes G63 about 11:30 p.m. Saturday in the upscale Lower Garden District of New Orleans. Mr. Smith's vehicle was pushed it into a Chevrolet Impala driven by two of Mr. Smith's acquaintances. Mr. Smith and Mr. Hayes argued, and Mr. Hayes allegedly shot both Mr. Smith and his wife, Racquel Smith, police said. She was wounded in the leg and taken to a hospital. The Smith family said in a statement that they were thankful for the outpouring of support but requested privacy as the family grieves for a "devoted husband, father and friend." Will and Racquel Smith have three children…"

G:

new orleans saints player will smith is shot and killed

G-L-B:

new orleans saints player will smith killed in road rage incident police say

G-L-M:

new orleans saints player shot and killed in road rage

G-L-P:

new orleans saints player will smith killed in road rage

G-L-P-R1:

saints player will smith shot and killed in new orleans

G-L-R1:

saints player will smith killed in road rage

G-L-RL:

new orleans saints player will smith killed in road rage

OpenNMT:

former nfl player shot dead in

T5:

new: police say no indication that the shooting was a road rage. new:

T5-finetuned:

Former Saints player Will Smith shot and killed in road rage

For this example, the T5-finetuned and G-L-P models stand out as the most effective in preserving critical information while maintaining conciseness and clarity. These models accurately capture the central event, which is Will Smith's death, and add important context to it such as road rage. However, they, along with most other models, fail to include the detail about the "traffic accident" which is mentioned in the ground truth headline. Also, G-L-P doesn't explicitly state that he was "shot". The mentioned two headlines balance brevity and informativeness which they correctly mentioned Will Smith, his connection to the New Orleans Saints, and the cause of death as a shooting. The two least effective models to generate headline for this article are OpenNMT and T5. In this example, OpenNMT is incomplete, omitting critical elements such as the name, team, or context. T5 diverges entirely from the event, introducing irrelevant or contradictory information. G-L-B is slightly verbose due to "police say", which is unnecessary unless attributing a source is relevant. G-L-M lacks specific mention of "Will Smith", which makes it less informative. G-L-P-R1 doesn't mention "traffic accident" or "road rage" omitting some important details. Both G-L-R1 and G-L-RL fail to explicitly mention that he was "shot" and lacks location context.

Article 2:

The perfect McDonald's breakfast for every type of hangover

"Starting Tuesday, McDonald's will be offering their breakfast menu all day, which means hungover people can finally access the food that was intended for their consumption. Previously, McDonald's offered their breakfast menu until 10:30 a.m. on weekdays and 11:00 a.m. on

weekends, which is far too early for any severely hungover person. It's hard enough to get up before noon after a night filled with drinking, but 11 a.m. is just outrageous.

So now that McD's is finally giving intoxicated people what they want, we decided to pair up some of their breakfast menu items with the different types of hangovers their customers will inevitably have.

Unfortunately, McDonald's all-day breakfast menu will be limited, so some substitutions may be needed depending on your area. Have something to add to this story? Share it in the comments."

G:

mcdonalds offers breakfast menu for breakfast

G-L-B:

mcdonalds to introduce breakfast menu all day

G-L-M:

mcdonalds to offer breakfast menu

G-L-P:

mcdonalds to introduce breakfast menu

G-L-P-R1:

mcdonalds to offer breakfast menu

G-L-R1:

mcdonalds to offer breakfast menu

G-L-RL:

mcdonalds to offer breakfast menu

OpenNMT:

mcdonald 's to open thanksgiving breakfast

T5:

McDonald's will be offering breakfast menu all day. hungover people can finally access

T5_finetuned:

McDonald's will offer all-day breakfast menu to hungovers

The generated headlines show varying degrees of success. Most focus narrowly on the factual aspect of the "all-day breakfast" announcement but fail to capture the humor and thematic relevance of hangovers that are central to the article. Headlines such as G-L-M, G-L-P, and G-L-B are accurate but overly generic, offering little to engage the reader beyond the basic news. Others, like T5 and T5_finetuned, take a step closer to the ground truth by incorporating both the all-day breakfast feature and the hangover theme. The worst example form OpenNMT,

"McDonald's to open Thanksgiving breakfast," is entirely off-topic, highlighting a failure to understand the article's context.

A headline generated by G had repetitive phrasing ("breakfast menu for breakfast"). It also misses "all-day" feature, which is a significant announcement in the article. Also "all-day" information is omitted from the headlines generated by G-L-M, G-L-P, G-L-P-R1, G-L-R1, and G-L-RL as well. The weakest headlines are those generated by OpenNMT and G, as the former is irrelevant, and the latter suffers from redundancy. On the other hand, the best ones are generated by T5-Finetuned, T5, and G-L-B.

Article 3:

Grace Glofcheskie remembered by hundreds at Arnprior, Ont., funeral

"Hundreds of people gathered at a church in Arnprior, Ont., Saturday afternoon for the funeral of Grace Glofcheskie, the 24-year-old University of Guelph student killed in a hit and run six days earlier.

Glofcheskie was struck and killed by a stolen 2007 Ford Escape SUV in Guelph's downtown at around 2:30 a.m. on Dec. 13 as she was returning home from a night out with friends.

The vehicle rolled, and the driver — who had been fleeing a spot check — ran away.

Glofcheskie had been working on her master's degree at the University of Guelph and was a member of the university's golf team.

""[She] was a very, very close friend. I had met her family. She had been there every step along the way of my academic journey,"" said Shawn Beaudette, one of about 20 of Glofcheskie's

university colleagues who made the nearly 500-km drive from Guelph to Arnprior, her hometown, to pay their respects.

Beaudette, who knew Glofcheskie for about five years, said the ""tight-knit"" biomechanics department at the University of Guelph where they studied together was still in shock.

""It's like a family, so it's been very, very sombre since we heard the news,"" said Beaudette. ""It's been a long week.""

The university has a longstanding tradition of painting an antique naval cannon that sits in the heart of campus, and this week the cannon was painted in Glofcheskie's memory, he said.

Shawn Beaudette said the 'tight-knit' University of Guelph community where he and Grace Glofcheskie studied is still in shock over her death.

About 300 people attended the funeral at St. John Chrysostom Catholic Church in Arnprior, said parish member Leo Lawlor — a sign that Glofcheskie was ""well-liked"" in the community of about 8,000.

Two days after the hit and run, police charged 21-year-old Curtis Henri of Guelph with a number of offences, including dangerous driving causing death, criminal negligence causing death, and possession of property obtained by crime over $5,000.

Glofcheskie's parents have started a memorial fund in their daughter's name to support academics and athletics in Arnprior and at the University of Guelph."

G:

hundreds gather for grace glofcheskie funeral

G-L-B:

hundreds attend funeral for university of guelph student grace glofcheskie

G-L-M:

hundreds gather for grace glofcheskie funeral

G-L-P:

hundreds gather for funeral of grace glofcheskie

G-L-P-R1:

hundreds gather for funeral for slain student

G-L-R1:

hundreds attend funeral for grace glofcheskie

G-L-RL:

university of guelph student killed in hit and run

OpenNMT:

funeral held for <unk> in

T5 :

a stolen 2007 Ford Escape SUV hit and run in downtown of a university of gu

T5_finetuned:

Hundreds gather for funeral of University of Guelph student Grace Glofcheskie

The generated headlines vary in their ability to replicate the effectiveness of the ground truth headline. Headlines, such as "hundreds gather for Grace Glofcheskie funeral" by G-L-M and G and "hundreds attend funeral for University of Guelph student Grace Glofcheskie," that is generated by G-L-B, successfully capture some factual elements. Simpler headlines, such as "hundreds gather for Grace Glofcheskie funeral," generated by G-L-M are functional but fail to evoke empathy or provide broader context.

Several generated headlines demonstrate significant weaknesses. Headlines like "university of guelph student killed in hit and run" generated by G-L-RL shift focus away from the funeral to the hit-and-run incident, which is a peripheral detail in the context of this story. Others, like "funeral held for <unk> in," by OpenNMT is incomplete and fails to convey any meaningful information. And the headline generated by T5 is incomplete and overly detailed in mentioning the vehicle involved. These shortcomings highlight the challenges these models face in maintaining focus on the primary event.

Among the generated headlines, the strongest performer is "Hundreds gather for funeral of University of Guelph student Grace Glofcheskie," produced by the T5_finetuned model. This headline combines Grace's name, her university affiliation, and the scale of the event. A similar headline, "Hundreds attend funeral for University of Guelph student Grace Glofcheskie," by G-L-B is also competitive with the one generated by T5-finetuned. A headline generated by G-L-R1 is similar to the one generated by G, but substitutes “gather” with “attend” and lacks the locational context.

4.7. Conclusion

In this study, we employed self-critical sequence training (SCST) within Reinforcement Learning (RL) for the task of headline generation. Our work demonstrated how to leverage SCST to optimize a headline generator model based on the combination of multiple metrics simultaneously. Furthermore, we introduced a headline popularity dataset, derived from Twitter, and trained a transformer model using this dataset to assess the attractiveness of generated headlines. We also incorporated the trained popularity prediction model as a reward function during the training of the headline generation model. To the best of our knowledge, this is the first time a headline generation model has been trained using this type of reward function.

Since we were unable to train the headline generation model using SCST alone, we followed previous work and combined the RL loss with the cross-entropy loss function. However, the training is computationally expensive because the model generates three separate headlines for each article during the training phase. In addition to this, the reward function introduces further computational overhead, especially when it involves another deep learning model.

Moreover, our experiments revealed that transformer-based models, particularly the T5 model fine-tuned for headline generation, significantly outperformed LSTM-based architectures across various automatic and human evaluation metrics. This underscores the effectiveness of the transformer architecture in handling the complexities of headline generation, as reflected in diverse performance metrics. Based on this observation, we propose a novel transformer-based method in the next chapter to generate popular headlines that can be trained not only faster but also more efficiently.

For future work, hyperparameter tuning could be applied to find the optimal balance between the RL loss and the cross-entropy loss function. Additionally, exploring newer RL methods, such as Proximal Policy Optimization (PPO) and Group Relative Policy Optimization (GRPO), for training headline generator models could be valuable.

Chapter 5
Learning to Generate Popular Headlines

5.1. Introduction

Transformer models have started a new era in natural language processing, showing unprecedented performances in different NLP tasks including text summarization. Due to their extraordinary capabilities and their advantages over other NLP methods such as RNNs, we decided to employ them for headline generation task to generate better headlines for news articles.

In this chapter, we propose a novel method for generating popular headlines for news articles based on transformers. In the proposed method, we first fine-tune a state-of-the-art open-source transformer model for the task of headline generation and use it to produce multiple headlines for each article. Next, we employ our headline popularity prediction model to predict the popularity of each generated headline and choose the headline with the highest predicted score.

The main contributions of our work are summarized as follows:

- We propose an approach to generating popular headlines using a combination of state-of-the-art open-source transformer models and a headline popularity prediction model.

- We propose a method that utilizes a popularity benchmark to automatically assess the capability of generative models to generate popular headlines.

5.2. Generating Popular Headlines

5.2.1. Overview

An overview of our proposed headline generator approach is shown in Figure 5-1. The proposed approach consists of four main components: Tokenization, Fine-tuning, Generation, and Selection. The first three components pertain to a transformer generator model, which is responsible for generating multiple candidate headlines for a given input article. While the last component, a customized version of a transformer encoder for regression, is utilized to select the best headline among the generated candidates.

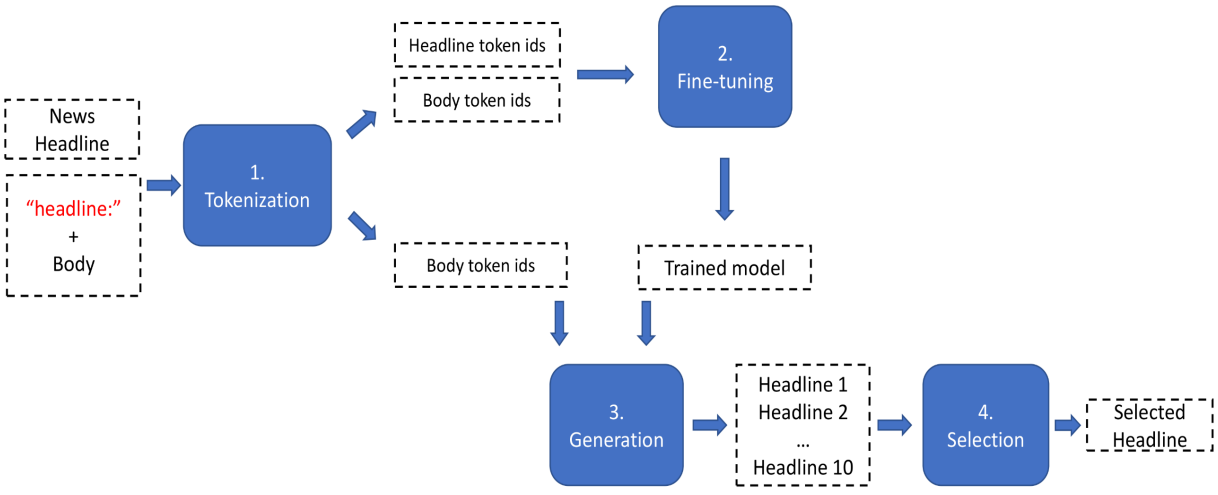

Figure 5-1 A proposed headline generation approach to generating popular headlines.

We provide a detailed description of the first three components in the generation subsection and the fourth component in the selection subsection.

5.2.2. Generation

Tokenization: In this part, we process the headlines and bodies of the articles in training, validation, and test parts of the Newsroom dataset, using the appropriate tokenizer for the transformer model. The tokenizer converts the input texts (headlines or bodies) into tokens ids. We define a new task to train the models for headline generation by appending "headline:" at the beginning of each article's body before tokenization. Inserting "headline:" at the beginning of the input to the models allows them to recognize that this is a different task from the summarization task for which they have already been trained.

Fine-tuning:

In this phase, we fine-tune all layers of the transformer models to generate headlines that resemble the ground truth data (i.e., the actual headlines of the articles). We can use any kind of generative transformer models in our approach, but we use the state of the art models, namely BART [64], ProphetNet [85], and T5 [62]. For fine-tuning, we use a greedy approach to generate each token, compare each generated token to its ground truth counterpart, and calculate the loss using the cross-entropy loss function. At the end of each epoch during model fine-tuning, the model is evaluated on the validation dataset and the model with the lowest loss value is kept as the best model.

Generation:

In this phase, we initially utilize a fine-tuned headline generation model from the previous stage (i.e., fine-tuning component) to produce 10 distinct candidate headlines for each input article. The rationale behind generating multiple headlines is that the generator was fine-tuned to generate only a concise headline describing the article's content. However, depending on the training dataset, the generated headline may not be appealing to readers. Therefore, we employ the generator to generate multiple titles that accurately describe the article, and subsequently, we select the most attractive one from them.

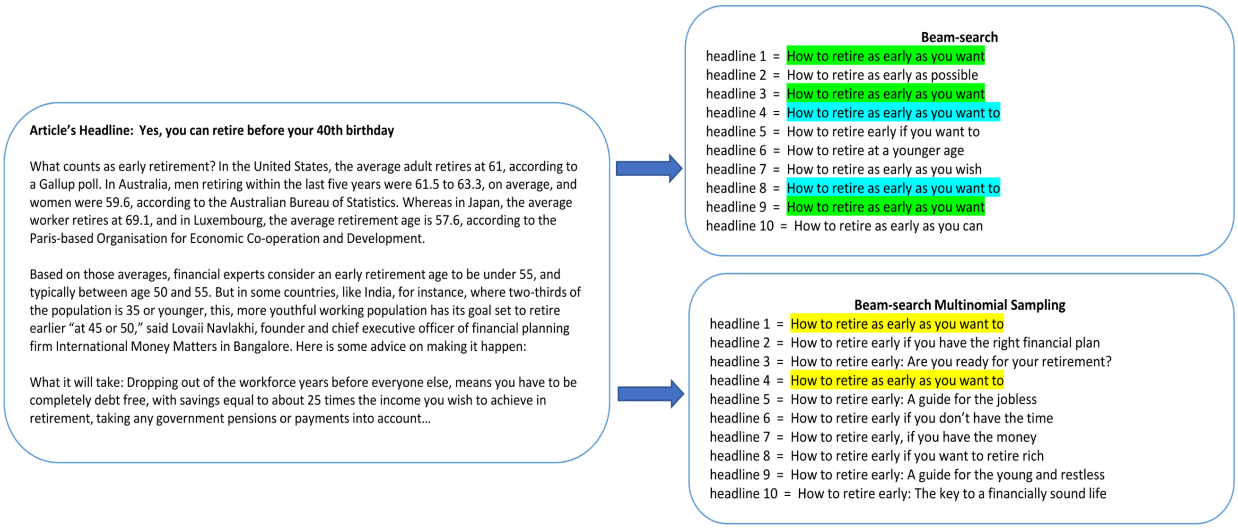

Figure 5-2 An example of using greedy search versus using multinomial sampling techniques in beam search. The highlighted headlines are repeatedly generated more than once. The beam search with multinomial sampling generates fewer duplicated headlines.

We use the beam search technique to generate multiple headlines. Since the top headlines generated from the beam search are similar to each other (Figure 5-1), we need to have a strategy to diversify the generated headlines. To avoid generating identical or similar headlines, we replace the greedy approach used in the beam search with the multinomial sampling technique.

In this method, the transformer model provides a probability distribution over the tokens for each generated token, based on the input sequence (i.e., an article's body) B and previously generated tokens $w_{\{1:t-1\}}$ according to formula (5-1).

$$w_t \sim P\left(w \mid w_{\{1:t-1\}}, B\right) \tag{5-1}$$

The model chooses the next word w_t randomly sampled from this probability distribution using multinomial sampling technique, giving a higher probability to tokens with higher probabilities in the distribution.

5.2.3. **Selection Phase**

This part of the model uses a headline popularity prediction model to determine the popularity score for the candidate headlines and then select the one that achieves the highest score. The architecture of the headline popularity prediction model is shown in Figure 5-3.

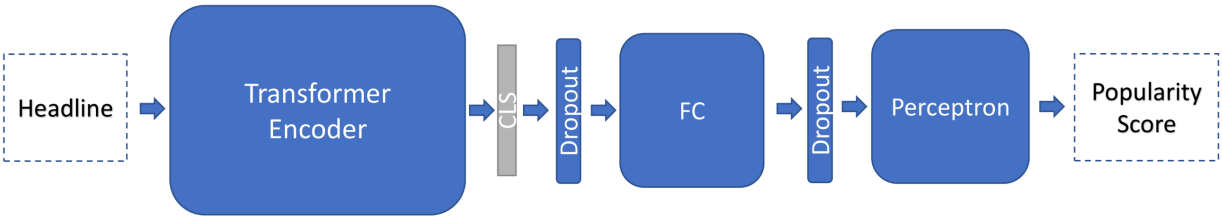

Figure 5-3 The architecture of the Headline Popularity Prediction Model.

Given a news headline (denoted as $x_1, x_2, \ldots, x_m$, where x_i is a token), the prediction model first uses a transformer encoder to convert the input headline $x_1, \ldots, x_m$ into a semantic representation (i.e., *H*). We then extract the embedding representation of the CLS token from the last hidden layer of the model (i.e., $H^0_{\{last\}}$ in formula (5-3) below), and feed it into a fully-connected layer (FC) whose input and output layers has the same size as the embedding vector and tanh is as the activation function. On top of all the previous layers, a regressor is used, which is a perceptron having a linear activation function, to predict a popularity score. This process can be expressed as:

$$H = Encoder(x_1, \ldots, x_m) \tag{5-2}$$

$$p = Perceptron\left(FC\left(H^0_{\{last\}}\right)\right) \tag{5-3}$$

A major challenge in developing the popularity prediction model is obtaining a labeled training data set. Since no such training data is publicly available, we created a new headline popularity dataset (HP) as described in the previous chapter. This dataset is used both in the previous chapter and the current chapter.

5.3. Datasets

To train our headline generation models, we use Newsroom dataset [77] that we described it in 4.4.1.1. Also, to train our headline popularity prediction model we use our headline popularity

dataset that was described in 4.4.1.2. However, our methods can be applied to other datasets that contain the same types of information.

Since headlines are from different news media, and news media have different number of followers, we compare the number of likes of the headlines with respect to the number of likes of headlines from the same news media, as described in the following steps:

- We remove the articles that belong to the four least frequent news media sources.
- We remove the news articles whose body is not available.
- To detect outlier headlines in terms of popularity, we compare the popularity of each headline with that of other headlines from the same publisher (i.e., news media). To do that, we calculate a quantile of 90% for favorite count per each news media. Then headlines that their favorite count is greater than their quantile 90%, are considered outliers. Instead of removing the outliers, we clip their favorite count to the value of their quantile 90%. Figure 5-4 and Figure 5-5 show the distribution of favorite count for each news media before and after handling the outliers, respectively.
- Then, we apply minmax normalization technique to favorite count values for each news media (Figure 5-6).
- In the end, we divide the data into training, validation, and test data sets with a percentage of 90%, 5%, and 5% percent, respectively (Table 5-1)

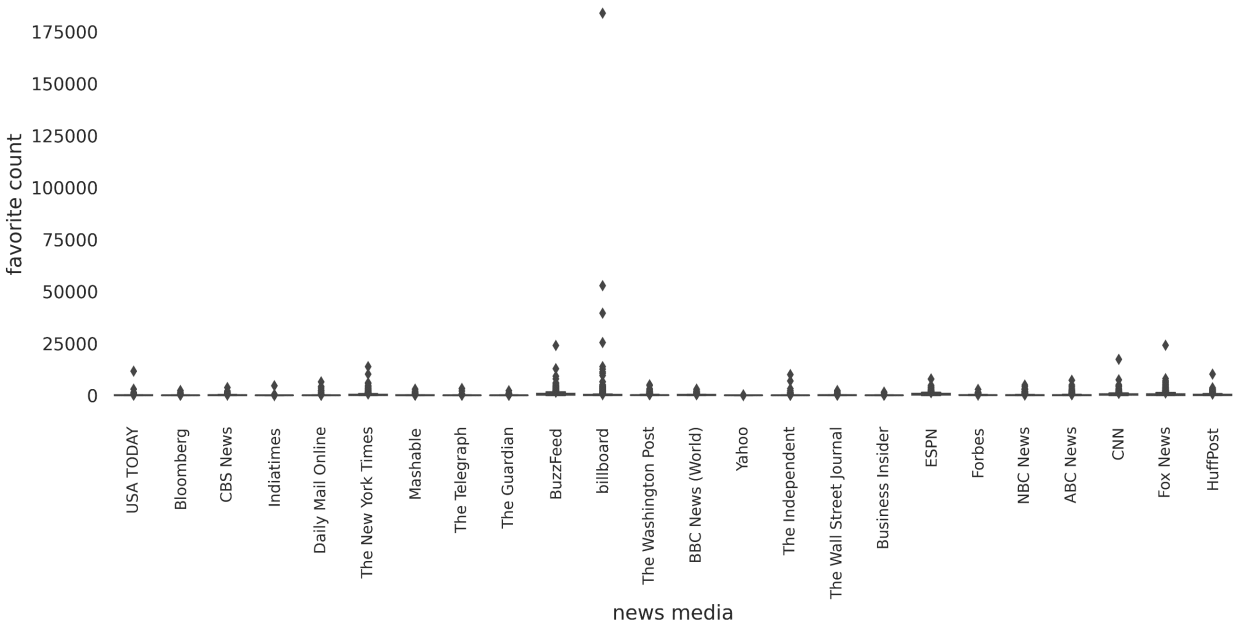

Figure 5-4 A box plot of the favorite count values for each news media, including outliers.

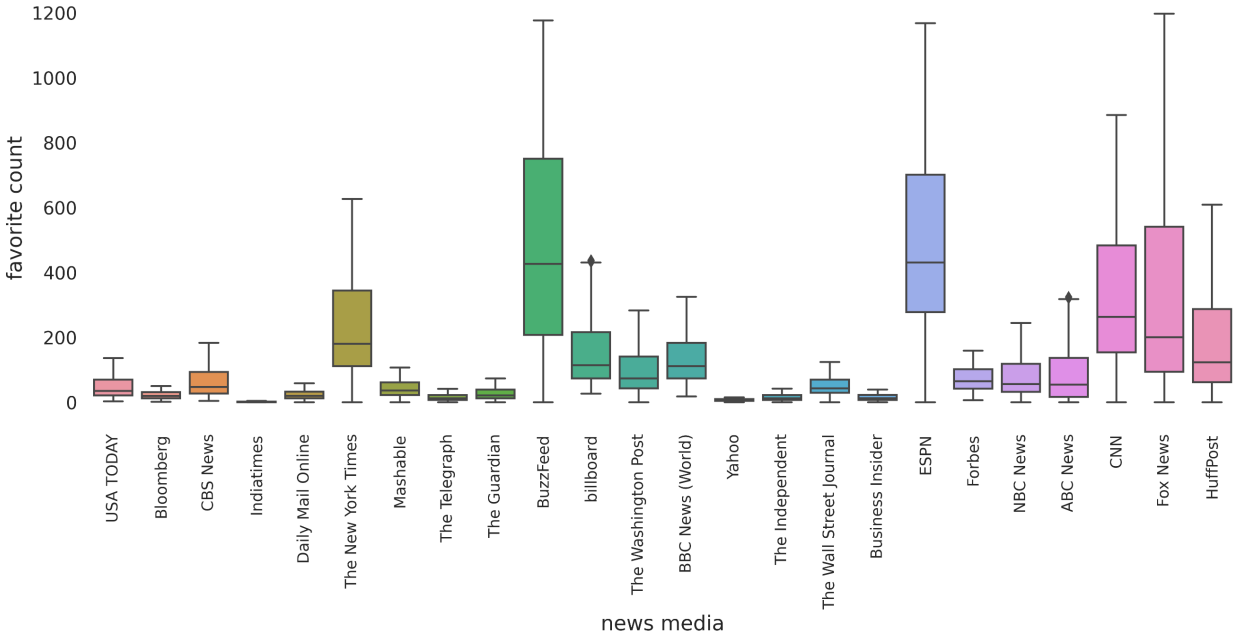

Figure 5-5 A box plot of the favorite count values for each news media, excluding outliers.

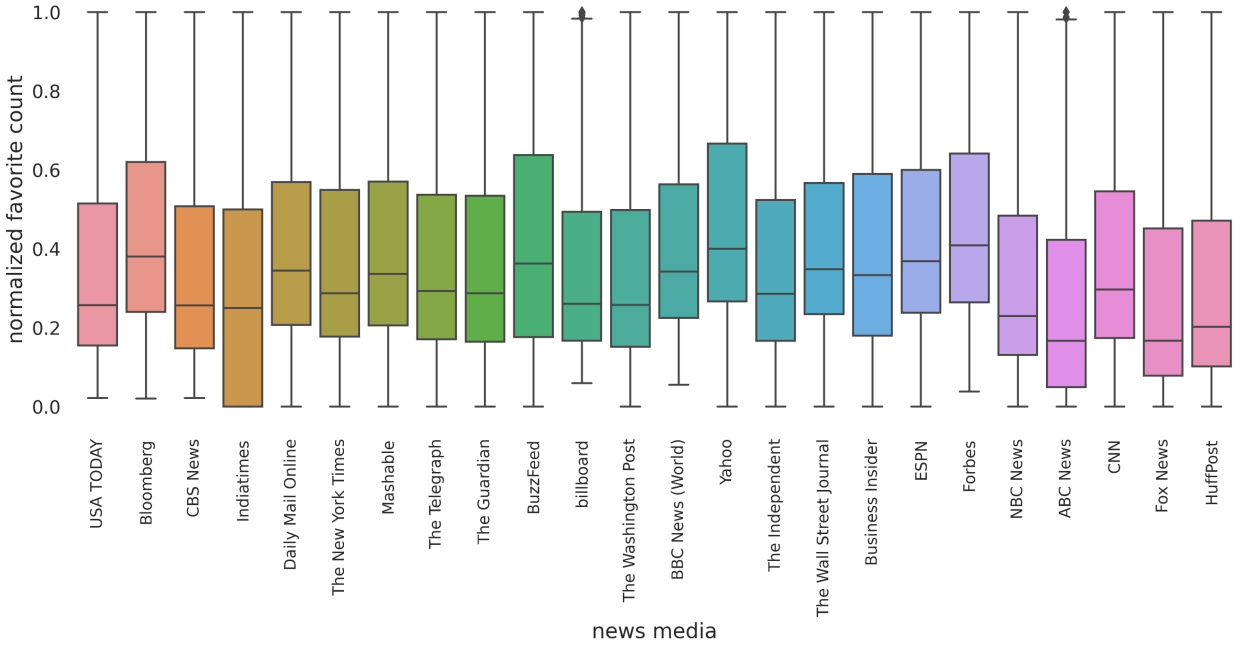

Figure 5-6 A box plot of normalized favorite count values for each news media, with outliers removed.

Table 5-1 Size of the training, validation, and test sets of the HP dataset after Pre-processing.

Set	Size
Train	80070
Validation	4448
Test	4449

5.4. Evaluation

This section consists of two parts. In the first part, we generate headlines using the state-of-the-art summarization models. We then train these models on a headline generation task using the training set of the Newsroom dataset. After each epoch of model fine-tuning, the model's

performance is evaluated on the validation dataset. The model with the lowest loss value is considered the best model and is kept for further use.

In the second part, we generate headlines using the proposed headline generation approach described in subsection 5.2.2. We also create a popularity benchmark to investigate whether using the proposed approach leads to the generation of more popular headlines or not.

5.4.1. Evaluation of Pre-trained and Fine-tuned Models on Headline Generation

First, we conduct our experiment using the state of the art pre-trained versions of BART [64], ProphetNet [85], and T5 [62] models, which, that according to their reported results, have demonstrated noteworthy performance in summarization tasks [15].

T5 [62]**:** The T5 transformer model, a powerful natural language processing (NLP) tool developed by Google, is based on the standard encoder-decoder transformer architecture proposed in [55]. The T5 model uses the transformer architecture [55], which employs self-awareness mechanisms to process and generate text. It has achieved state-of-the-art performance on various NLP tasks, such as question-answering, text classification, and summarization.

During pre-training, 15% of the tokens are randomly removed and replaced with sentinel tokens. Then, in the decoder, the target is the dropped-out tokens delimited by their respective

[15] These models were fine-tuned on the summarization task based on models pre-trained on other tasks such as masked token prediction.

sentinel tokens. The main characteristic of T5 is that it was trained with unified pre-training objective. In this approach, T5 learns to perform various downstream tasks by representing them as text-to-text transformations.

BART [64]**:** The BART model is another powerful transformer model for NLP tasks that is developed by Facebook. This model is based on the standard transformer architecture [55] having a bidirectional encoder similar to BERT and a left-to-right decoder like GPT [86]. BART was pre-trained on large amounts of text data, which makes it very effective for a wide range of NLP tasks such as text summarization, question answering, and language translation. The pre-training tasks for BART include token masking, text infilling, token deletion, document rotation, and sentence permutation.

The main characteristic of BART that distinguishes it from other transformers is its pre-training objective task, where corrupted versions of the input are given to the model, and the model learns to reconstruct the original input.

ProphetNet [85]**:** ProphetNet, a transformer model that is developed by Microsoft, has achieved impressive results in summarization tasks. ProhpetNet is based on the standard transformer architecture [55]. The main characteristic of ProphetNet that distinguishes it from other transformer models is that it introduces a novel objective function that instead of predicting the next token, the model predicts the next *n* tokens simultaneously.

Baseline: We trained the model proposed in [87] on Newsroom dataset and compare its result with the mentioned generative transformer models. The proposed model consists of a transformer encoder-decoder model [55] and a customized attention layer.

We use the pre-trained versions of these models to generate headlines for the test set of the Newsroom dataset. The generated headlines are compared to the ground truths using ROUGE

[10], BLEU [84], and METEOR [11] scores. Then, we fine-tune the above models on a training portion of the Newsroom dataset. During fine-tuning, we define a new task by appending 'headline:' to the beginning of the input text for the model.

We use the batch size of 16 and the AdamW [88] optimizer with the learning rate of $2 * 10^{\{-5\}}$ and weight decay of 0.01. We also use a half precision mode (i.e., 16-bit floating-point) to speed up the training. The results of the models on the test set of the Newsroom dataset are shown in Table 5-2.

Table 5-2 A comparison between pre-trained summarization models and their fine-tuned versions for headline generation task.

Model	Version	R1	R2	RL	B1	B2	B3	B4	METEOR
BART	fine-tuned	0.387	0.1941	0.3525	**0.3384**	**0.1603**	**0.0843**	**0.0522**	**0.3524**
	Pre-trained	0.1684	0.0544	0.1479	0.1273	0.0367	0.0117	0.0051	0.1525
T5	Fine-tuned	0.3259	0.1523	0.2994	0.2917	0.1281	0.0615	0.0373	0.2755
	Pre-trained	0.1704	0.0516	0.1512	0.1266	0.0332	0.0106	0.0045	0.1513
ProphetNET	Fine-tuned	**0.3873**	**0.1942**	**0.3551**	0.2996	0.1314	0.0632	0.0361	0.3203
	Pre-trained	0.1946	0.0645	0.172	0.1263	0.0341	0.0101	0.004	0.1799
Baseline	Trained	0.2834	0.1199	0.2588	0.2693	0.1216	0.0623	0.0358	0.2494

As we can see in Table 5-2, the fine-tuned models (headline generation models) drastically outperform the pre-trained versions (summarization models). Even though the pre-trained BART, ProphetNet, and T5 models perform well on the summarization task [62], [64], [85], they underperform on the headline generation task. However, when we fine-tune them specifically on a headline generation task, their results improve significantly on all the performance measures listed in Table 5-2.

In terms of BLEU and METEOR scores, the fine-tuned version of BART is the best model on the headline generation task. However, in terms of ROUGE scores, the fine-tuned version of ProphetNet is the best. Interestingly, this pattern holds for the pre-trained versions of the models as well that the pre-trained ProphetNet performs by far the best among the three pre-trained models in terms of ROUGE scores while the pre-trained BART is slightly better than the pre-trained ProphetNet in terms of the BLEU scores. However, unlike what we saw for the fine-tuned models, the pre-trained ProphetNet outperforms the two other pre-trained models in terms of the METEOR score.

We also train the baseline [87] on the training set of the Newsroom dataset and compare its results with the other models. Although the baseline could beat all the pre-trained models, it could not compete with any of the fine-tuned ones in terms of the evaluation metrics used. In the next section, we use these fine-tuned models as the headline generation models and evaluate our proposed approach to generating popular headlines.

5.4.2. **Evaluating the Proposed Approach on Popularity of Generated Headlines**

In this section, we create a popularity benchmark and then evaluate how well our proposed approach, as outlined in subsection 5.2.3, performs in generating popular headlines. The benchmark comprises popular headlines (and associated articles) from the HP dataset we previously constructed. If our approach outperforms baseline models in terms of text generation

metrics (such as BLEU, ROUGE, and METEOR) on popular articles, indicating that the headlines we generate are more similar to popular ones than those generated by the baselines, then our model is considered better at generating popular headlines.

The steps we follow to create the popularity benchmark are illustrated in Figure 5-7.

Figure 5-7 Steps for creating the popularity benchmark.

In the first step, we select the headlines with a popularity score above 0.8 from the test and validation sets of our HP dataset. As a result, 1084 popular news articles remain. However, an article in the benchmark data set may occur in the training set of the Newsroom dataset. Thus, we remove the articles from the popularity benchmark whose headline is similar to at least one headline from the training set of the newsroom dataset. To do this, in the second step, we convert all headlines in the benchmark and Newsroom datasets into their semantic embedding representations using SBERT.

SBERT [14] is a type of transformer encoder model trained specifically to convert sentences into their embedding representations. The key component of SBERT is the Siamese architecture, which contains two identical transformer encoders in terms of architecture and

shares the same weights during its training phase. Additionally, it employs a triplet loss function [89] to train the learning parameters in a way that the cosine similarity of two similar sentences (the similarity score of an anchor sentence and a positive sample) becomes larger than the similarity score of dissimilar sentences (the similarity score of an anchor sentence and a negative sample).

In the final stage of creating the popularity benchmark, we compute the cosine similarity score between each headline in the benchmark and all the headlines in the Newsroom dataset. We retain the maximum similarity score as the headline's similarity score in the benchmark with respect to the Newsroom dataset. If the calculated similarity score exceeds 0.6, we remove the headline from the benchmark. After this step, the remaining number of headlines is 677.

Again, the idea behind using the popularity benchmark is that if the proposed model could improve the baselines on the text generation metrics, we can conclude that the proposed architecture can enable the headline generator models to generate more popular headlines, since all the ground truth headlines in the benchmark are highly popular. In this way, we can assess how good the generated headlines are in terms of popularity.

After creating the popularity benchmark, we use each fine-tuned and pre-trained models mentioned in Table 5-2 to generate headlines for the articles in the popularity benchmark. Then, we employ each of these models in our proposed architecture described in subsection 6.3 and investigate whether using our proposed approach could improve the results in terms of the evaluation metrics. The results of the proposed approach are indicated by appending * to the end of the name of the model.

For the selection component, as explained in section 5.2.3, we train our headline popularity prediction model on the training set of the HP dataset. We retain the best performing snapshot

of the model, achieving the lowest Mean Squared Error (MSE) of 0.065101 on the validation dataset. We use the batch size of 64, the AdamW optimizer with a learning rate and weight decay of $2 * 10^{\{-5\}}$ and $1 * 10^{\{-5\}}$ respectively, and MSE as the loss function. Regarding the generation component, we conduct our experiment with each fine-tuned model as the generation component of the proposed approach (subsection 5.2). We also run the experiment using each pre-trained model as the generation component of the proposed approach.

The results of this experiment on the popularity benchmark are shown in Table 5-3. As we can observe in Table 5-3, regardless of the type of model used (i.e., T5, BART, or ProphetNet), and whether it is fine-tuned or pre-trained, our proposed approach can improve the results in terms of all evaluation metrics in the popularity benchmark. For example, comparing BART:fine-tuned to BART:fine-tuned*, the latter outperformed the former in terms of all the evaluation metrics used. This also applies to all fine-tuned and pre-trained versions of the other models.

In order to investigate whether the differences between BART:fine-tuned and BART:fine-tuned* are significant, we apply a paired t-test. However, before applying the paired t-test, we need to determine whether it is appropriate by using the Shapiro-Wilk normality test. The results of the Shapiro-Wilk normality test for BART fine-tuned and BART fine-tuned* are 0.4221 and 0.4194, respectively.

*Table 5-3 The comparison between headline generator models with and without using the proposed approach on the popularity benchmark. The methods marked with a * use our popularity prediction model to select the most popular headline from 10 candidates generated by our proposed strategy for generating multiple headlines.*

Model	Version	R1	R2	RL	B1	B2	B3	B4	Meteor
BART	Fine-tuned	0.3328	0.1563	0.2959	0.2532	0.1021	0.0450	0.0233	0.2578
	Fine-tuned*	0.3376	0.1579	0.2990	**0.2568**	**0.1055**	**0.0463**	**0.0241**	**0.2655**
	Pre-trained	0.2242	0.0912	0.2005	0.1710	0.0646	0.0310	0.0189	0.1849
	Pre-trained*	0.2419	0.1042	0.2155	0.1862	0.0732	0.0358	0.0214	0.2019
T5	Fine-tuned	0.2875	0.1307	0.2597	0.2412	0.0999	0.0482	0.0278	0.2157
	Fine-tuned*	0.2982	0.1383	0.2685	0.2456	0.1041	0.0486	0.0268	0.2261
	Pre-trained	0.1629	0.0529	0.1435	0.1244	0.0363	0.0171	0.0104	0.1269
	Pre-trained*	0.1793	0.0647	0.1579	0.1362	0.0446	0.0220	0.0130	0.1433
Prophetnet	Fine-tuned	0.3375	0.1627	0.3025	0.1711	0.0596	0.0275	0.0126	0.2503
	Fine-tuned*	**0.3430**	**0.1663**	**0.3080**	0.1742	0.0621	0.0301	0.0135	0.2574
	Pre-trained	0.2095	0.0814	0.1834	0.1007	0.0260	0.0116	0.0059	0.1751
	Pre-trained*	0.2241	0.0900	0.1949	0.1040	0.0284	0.0128	0.0069	0.1897

Since both p-values are greater than 0.05, we fail to reject the null hypothesis of normality. This indicates that the data for both groups are normally distributed, allowing us to proceed with a paired t-test. After applying the paired t-test, the t-statistic and p-value are -4.165 and 0.0042, respectively. Since the p-value is less than 0.05, we reject the null hypothesis. This suggests that there is a statistically significant difference between the BART fine-tuned and BART fine-

tuned* results. The reason could be that, as all the metrics show small but consistent improvement, the t-test can detect this pattern, even if the changes appear minor. We conducted the same experiment, first performing the normality test followed by the paired t-test for the following comparisons: T5-finetuned vs. T5-finetuned*, ProphetNet-finetuned vs. ProphetNet-finetuned*, BART-pretrained vs. BART-pretrained*, T5-pretrained vs. T5-pretrained*, and ProphetNet-pretrained vs. ProphetNet-pretrained*. For each of these pairs, the paired t-test indicated that the differences were statistically significant.

Another interesting finding, similar to the one previously mentioned, is that the fine-tuned version of ProphetNet outperforms the fine-tuned version of BART in terms of ROUGE scores, whereas BART excels in BLEU metric. This discrepancy could stem from the distinct nature of ROUGE and BLEU scores; ROUGE is a recall-based metric, while BLEU is precision-based. Consequently, it's plausible to infer that fine-tuned ProphetNet may exhibit superiority over fine-tined BART in terms of recall, whereas fine-tuned BART may excel in precision.

However, unlike the previous results, this pattern no longer holds true for the pre-trained models, as the pre-trained version of BART outperforms the pre-trained version of ProphetNet with respect to all the evaluation metrics used.

We also conducted a similar experiment, but instead of using the popular headlines, we used the least popular ones (i.e., the popularity score is below 0.2). After removing unpopular headlines that are similar to the Newsroom training set (i.e., similarity score above 0.6), 1402 unpopular headlines remain. We then ran the same experiment that we did on the popularity benchmark with the unpopular headlines. The result of this experiment is shown in Table 5-4.

*Table 5-4 The comparison between headline generator models with and without using the proposed approach on the unpopular headlines. The methods marked with a * use our popularity prediction model to select the most popular headline from 10 candidates generated by our proposed strategy for generating multiple headlines.*

Model	Version	R1	R2	RL	B1	B2	B3	B4	Meteor
BART	Fine-tuned	0.3286	0.1496	0.2922	**0.2493**	**0.0997**	0.0440	0.0235	0.2563
	Fine-tuned*	0.3276	0.1487	0.2911	0.2477	0.0996	**0.0445**	**0.0241**	**0.2576**
	Pre-trained	0.2157	0.0812	0.1909	0.1602	0.0557	0.0253	0.0152	0.1777
	Pre-trained*	0.2231	0.0872	0.1969	0.1658	0.0598	0.0273	0.0161	0.1854
T5	Fine-tuned	0.2887	0.1263	0.2599	0.2393	0.0952	0.0424	0.0222	0.2198
	Fine-tuned*	0.2911	0.1288	0.2615	0.2376	0.0949	0.0425	0.0223	0.2243
	Pre-trained	0.1662	0.0515	0.1458	0.1249	0.0351	0.0155	0.0092	0.1314
	Pre-trained*	0.1721	0.0554	0.1512	0.1280	0.0374	0.0167	0.0097	0.1374
Prophetnet	Fine-tuned	**0.3345**	**0.1562**	**0.2997**	0.1764	0.0596	0.0267	0.0134	0.2505
	Fine-tuned*	0.3312	0.1550	0.2974	0.1718	0.0582	0.0264	0.0131	0.2511
	Pre-trained	0.2027	0.0733	0.1770	0.0966	0.0239	0.0106	0.0057	0.1707
	Pre-trained*	0.2066	0.0767	0.1802	0.0964	0.0240	0.0105	0.0054	0.1754

As we can see, the best results in terms of ROUGE scores, BLEU-1 and BLEU-2 no longer belong to the proposed approach. Also, the effect of the proposed approach varies between models, sometimes improving the outcome (e.g., BART:pre-trained vs. BART:pre-trained*), while in other cases worsening it (e.g., ProphetNET:fine-tuned vs. ProphetNet:fine-tuned*).

However, for the popularity benchmark the pattern was completely different in a way that using the proposed approach always improves the result. Here, we present examples of the headlines generated by our models and analyze them, which are selected from the Newsroom dataset.

Article 1

"John Kerry 'hopeful' of Iran deal as nuclear talks reach decisive phase"

"John Kerry has said that nuclear talks with Iran in Vienna were "getting to some real decisions" and that he was hopeful of a deal.

The US secretary of state said, however, that the negotiators still had "a few tough things to do". With the Russian and Chinese foreign ministers expected back in the Austrian capital on Sunday evening to rejoin Kerry and their counterparts from the UK, France, Germany and Iran, there were rising hopes that the elusive, but potentially historic, deal could be struck on Sunday or Monday.

On his return early Sunday afternoon, the French foreign minister, Laurent Fabius, said: "I hope we're arriving finally at the last phase of these marathon negotiations. I believe we are."

The optimism was tempered by knowledge there have been three missed deadlines in the past two weeks, and the talks seem to be stuck on the intractable question of whether the UN arms embargo on Iran and restrictions on its missile program should be left in place as part of any agreement.

The basic bargain underlying the deal – Iran accepts strict curbs on its nuclear program in return for relief from economic and financial sanctions – appears to have been agreed and drafted in a text of 20 pages, and five annexes taking up another 60 pages. The arms and missile issue were left until the endgame in Vienna.

“The missile issue now seems to be the only key matter blocking the way to a deal,” according to Ariane Tabatabai, writing in the Bulletin of the Atomic Scientists.

Kerry attended mass at Vienna’s 14th century cathedral, where Mozart got married, before making his way to the Palais Coburg hotel, where the talks are taking place. He entered the venue through the back door on the crutches he has been using since a bicycle accident at the end of May.”

T5 pre-trained:

the talks with Iran in Vienna were “getting to some real decisions” and that he

T5 pre-trained*:

John Kerry has said nuclear talks with Iran in Vienna were “getting to some real decisions”

T5 fine-tuned:

John Kerry hopeful of Iran nuclear deal

T5 fine-tuned*:

John Kerry hopeful of Iran nuclear deal, says talks still have 'tough things to do

BART pre-trained:

Kerry says negotiators still have 'a few tough things to do' French foreign minister

BART pre-trained*:

Kerry says negotiators still have 'a few tough things to do' Russian and Chinese

BART fine-tuned:

John Kerry says Iran nuclear talks are ‘getting to some real decisions 

BART fine-tuned*:

John Kerry says Iran nuclear talks are 'getting to some real decisions

ProphetNet pre-trained:

us secretary of state said talks were'getting to some real decisions'[X_SEP] but he said ne

ProphetNet pre-trained*:

u. s. secretary of state john kerry said he was hopeful of a deal. [X_SEP] he

ProphetNet fine-tuned:

john kerry hopeful of iran nuclear deal as negotiators'have tough things to do '

ProphetNet fine-tuned*:

john kerry says iran nuclear talks are getting to some real decisions

Article 2:

'Cuddlers' help keep babies born addicted to heroin alive

“Cuddling babies is an experience that many love. But for some babies, cuddling could mean life or death. Babies born addicted to heroin are some of the most vulnerable newborns. And cuddling is an important treatment for them as they withdraw from heroin.

At Mercy Health - Anderson Hospital several volunteers have the title of ""cuddlers"" and spend time comforting and snuggling with the small babies.

""They come into this world at a big disadvantage and I just want to help them out,"" volunteer Ted Rohling told WLWT.

""They don't have the nervous systems yet to settle themselves,"" added volunteer Donna Mullins. ""And sometimes the only time they sleep is when they are in someone's arms and you can rock them.""

Both Rohling and Mullins have been volunteers for several years.

The ""cuddlers"" are crucial for the babies' survival the hospital's manager Carmen Bowling said. ""If the mom has been a drug user for a long time, the babies require almost 24-hour holding because they are inconsolable,"" Bowling said.”

T5 pre-trained:

we thought you'd like these... Cuddling babies is an experience that many love

T5 pre-trained*:

we thought you'd like these... Cuddling babies is an experience that many love

T5 fine-tuned:

Children born addicted to heroin are the most vulnerable newborns

T5 fine-tuned*:

'Cuddlers' help babies with heroin

BART pre-trained:

Babies born addicted to heroin are some of the most vulnerable newborns. Cudd

BART pre-trained*:

Babies born addicted to heroin are some of the most vulnerable newborns. Cudd

BART fine-tuned:

Hospital 'cuddlers' help babies born addicted to heroin withdraw from

BART fine-tuned*:

Hospital 'cuddlers' comfort babies born addicted to heroin

ProphetNet pre-trained:

cuddling babies is an important treatment for them as they withdraw from heroin. [X_SEP] the " cu

ProphetNet pre-trained*:

cuddling babies is an important treatment for them as they withdraw from heroin. [X_SEP] " cuddle

ProphetNet fine-tuned:

' cuddlers'are crucial for the babies'survival at mercy health - anderson hospital

ProphetNet fine-tuned*:

cuddling babies is crucial for their survival

Article 3:

Foreign aid is not a waste of money

As public spending cuts have an impact across European and North American countries, the debate over the value of overseas aid is reaching new levels of intensity, no more so than in the United Kingdom where the Coalition has pledged to ringfence and increase the aid budget to reach 0.7 per cent of its Gross National Income from 2013. The devastating earthquake and tsunami in Japan adds a new and urgent perspective, tragically showing how even a rich and developed nation can find itself in need of external humanitarian assistance.

The debate encourages us to think about the role of foreign aid in developing countries – its potential, its benefits, as well as its possible pitfalls. Is foreign aid affordable? Does the need for foreign aid outweigh domestic calls on public spending? Will it reduce poverty or instead be lost to corruption? Will it lead to a culture of dependency? These are critical questions for governments and the public they represent, and I would like to add my thoughts both from the perspective of President of Liberia and Goodwill Ambassador for Water and Sanitation in Africa.

Whether you measure poverty by the one and a half billion people living on less than a pound a day, the one billion who go hungry each day, or other indicators such as the eight million children who die each year from preventable diseases such as pneumonia, diarrhoea and malaria, poverty on the scale it exists today is an affront to our common humanity. It also carries a significance beyond national borders and is therefore of global importance. In an increasingly interlinked world, countries are more dependent on one another for their prosperity, security and safety, and to answer challenges such as climate change.

I have seen, at first–hand, how aid, effectively targeted and delivered, reduces poverty. In sub-Saharan Africa there have been major improvements in child health and in primary school enrolment over the last two decades. To choose one example, between 1999 and 2004, the continent achieved one of the largest reductions in measles' deaths ever seen. These positive results and outcomes would not have been possible without the support of donors such as the UK's Department for International Development (DFID), complementing the resources which low-income countries mobilize domestically. In my own country, Liberia, both humanitarian and development aid have helped us recover and rebuild from the devastation and trauma of civil war, improving the future for the millions directly involved and affected.

The British public is right to demand that their aid be corruption-free. It is the poor who bear the heaviest burden of corruption – it pushes them further into poverty, and deepens inequality and injustice. Corruption is a continual risk to progress, and combating it must be a joint endeavour pursued tirelessly by developed and developing countries. In Liberia, I declared a zero tolerance on corruption when I assumed office, and I continue to stand by this declaration. I believe passionately that all governments receiving aid have an enduring obligation to ensure that it is properly used. When they do so, and their anti-corruption efforts complement donor

transparency and independent monitoring such as adopted by DFID, I have no doubt that aid can and does achieve its essential objectives.

Citizens in both the North and South also have a right to demand that aid achieves value for money, and in this light I am pleased to see that DFID has identified increasing the number of people with access to clean water and sanitation as a top priority for Liberia as well as other African countries, including the Democratic Republic of the Congo, Malawi and Mozambique. There can be few more important and effective interventions, as diarrhoea, caused by lack of access to these basic human rights, is the biggest killer of children under five in Africa. With access to safe water and sanitation, lives are saved, health outcomes are improved, more children, especially girls, can go to school, and stability thrives. It is hardly surprising that the United Nations Development Programme estimates that for every £1 invested in water and sanitation, £8 is returned to the economy through increased productivity.

The UK has a long-sustained dedication to reducing poverty, inequality and inequity in the developing world, and for this the public should be proud. And whilst it can only be right in the current context that aid budgets in all donor countries undergo increased scrutiny, and that strong results and value for money are both demanded and expected, I believe it would be a fatal mistake for governments to adopt aid-sceptic or, worse still, anti-aid approaches.

Aid should, of course, never be an end in itself. Provided that it is delivered on the basis of being timely, temporary, and targeted, it can save lives and transform life chances in today's developing world, just as the Marshall Plan helped rebuild European economies after the long years of war, laying the platform for stability and prosperity. African entrepreneur Mo Ibrahim recently commented that the main objective of aid is to abolish the need for aid. Let this inform our approach as our vision for Africa and other developing nations going forward.

T5 pre-trained:

And dfid, the UK has a long-sustained dedication to

T5 pre-trained*:

The UK has a long-sustained dedication to reducing poverty, inequality and

T5 fine-tuned:

Is foreign aid affordable? would it be a fatal mistake for governments to adopt anti-

T5 fine-tuned*:

The UK is right to demand that aid be corruption-free

BART pre-trained:

The UK has pledged to increase its aid budget to 0.7 per cent of

BART pre-trained*:

The UK has pledged to increase its aid budget to 0.7 per cent of its

BART fine-tuned:

Liberia’s president Ellen Johnson sirleaf on foreign aid:

BART fine-tuned*:

Why foreign aid is essential for the world's poor countries

ProphetNet pre-trained:

President of liberia calls on african nations to do more to help their people. [x_sep] he calls on

ProphetNet pre-trained*:

Japan earthquake and tsunami shows even a rich and developed nation can find itself in need of external humanitarian

ProphetNet fine-tuned:

foreign aid is vital to tackling poverty in africa. here's why

ProphetNet fine-tuned*:

foreign aid is essential if we want to end global poverty

One important observation we can make from the above headlines is that the fine-tuned models are able to provide noticeably better headlines for the news articles compared to the pre-trained ones. Although the pre-trained models are state-of-the-art generative models that have demonstrated excellent performance in various downstream applications [62], [64], [85], they struggle to produce concise and complete headlines for the sample articles. However, when these models are trained specifically for the headline generation task, they show significant

improvement in performance. This serves as further evidence of the differences between summarization and headline generation tasks.

To compare the generated headlines using qualitative measures, we conduct a human evaluation in the next section.

5.4.3. **Human Evaluation**

We conducted a human evaluation on 50 articles using Amazon Mechanical Turk, employing five human evaluators per article. Each article and its model-generated headlines were presented to the evaluators, who rated each headline on a scale from 1 to 10, with 1 being the lowest score and 10 the highest, in terms of conciseness, attractiveness, and readability. The results of this experiment are presented in Table 5-5.

As anticipated, finetuning significantly enhances all three evaluation metrics (i.e., conciseness, attractiveness, and readability) across each model (T5, ProphetNet, and BART). This underscores the effectiveness of tailoring models specifically for the headline generation task, resulting in higher quality headlines from the perspective of human evaluators.

Moreover, our proposed approach, represented by the starred versions, demonstrates improvements in the attractiveness scores for all models. This indicates that our refinements positively impact the attractiveness of the generated headlines.

Table 5-5 The human evaluation result for the headline generator models

Model	Version	Conciseness	Attractiveness	Readability
BART	Fine-tuned	5.792	6.948	7.36
	Fine-tuned*	6.012	**6.976**	7.28
	Pre-trained	4.908	6.292	6.484
	Pre-trained*	5.26	6.54	6.796
T5	Fine-tuned	6.076	6.92	**7.516**
	Fine-tuned*	**6.092**	6.944	7.48
	Pre-trained	4.012	5.476	5.476
	Pre-trained*	4.036	5.596	5.484
ProphetNet	Fine-tuned	5.46	6.496	6.956
	Fine-tuned*	5.624	6.748	7.008
	Pre-trained	4.612	5.992	6.024
	Pre-trained*	4.516	5.996	6.024

5.5. Conclusion

We proposed a novel approach to generating popular headlines by combining transformer-based generative models with a headline popularity prediction model. To generate the headlines, we first fine-tuned state-of-the-art transformer generative models on the Newsroom dataset and used the fine-tuned models to generate multiple candidate headlines for an input article. Then, a

headline popularity prediction model is used to select the headline with the highest predicted popularity. The popularity prediction model consists of a several layers of neural network structures, including a transformer-based encoder, a fully connected layer and a regression layer. We created a new dataset, the Headline Popularity (HP) dataset, to train the headline popularity prediction model. To automatically evaluate the effectiveness of our proposed approach for generating popular headlines, we developed an approach by creating a popularity benchmark. Our results demonstrated that our proposed method improved the state-of-the-art text generation models in terms of the metrics for generation quality (that is, BLEU, ROUGE and METEOR) on the popularity benchmark consisting of popular articles, indicating the proposed method can better generate popular headlines.

Future research could investigate the potential of major language models like ChatGPT and GPT-4 in the field of headline generation. We expect these large language models might achieve better results than our models, as they not only have a significantly larger number of parameters compared to our models (T5, BART, and ProphetNet), but also trained on a massive amount of data, including news articles from various media sources. Additionally, these models benefit from human-generated input used to refine their behavior, such as reinforcement learning from human feedback (RLHF), which helps improve response quality and alignment with human preferences. However, evaluating these large language models presents another challenge, as it requires a benchmark dataset containing news articles that can be verified as unseen by these models during their training phase. Developing such a benchmark would be crucial for ensuring a fair and accurate evaluation of these models in headline generation task.

Chapter 6
Bringing Conversational Agents into the Loop

6.1. Introduction

The previous chapters focused on evaluating, predicting, and generating high-quality headlines. However, this chapter introduces a new approach to presenting headlines and article bodies. We propose a novel method based on news conversational agents that can enhance user experience. We predict in the near future, users instead of navigating through different pages of news portals, might communicate with a chatbot to stay informed about the news. The chatbot would answer their questions, provide them with a list of most relevant headlines of the articles, or provide updates on the latest events. With the emergence of powerful large language models (LLMs) such as ChatGPT, using news conversational agents to reshape the news consumption got closer.

Large Language models are based on transformer architecture [55], having huge amount of learnable parameters that are trained on the vast amount of data, enabling them to understand and generate human-like text. They excel in various downstream NLP applications such as question answering, text comprehension, summarization, and more. However, there are two major bottlenecks for using them as an AI assistant. First, each organization possesses valuable internal knowledge, such as FAQs, policies, regulations, etc., that can or should be used to resolve incoming inquiries. However, LLMs are trained based on public datasets and may not be aware of private knowledge sources that could help them resolve incoming inquiries more

accurately. Secondly, fine-tuning these LLMs on the organization's internal data is not an easy task due to factors such as the size of the LLMs, cost of training, frequent updates in the internal data, and data privacy. For example, in news media, news articles are published every day that LLMs are not aware of. If the news media decides to use an LLM as an agent, the agent would be unable to provide users with information about current events or answer their questions about what is happening now. On top of that, the fine-tuning option is not available for certain LLMs (e.g., GPT-4).

One possible solution to the mentioned problems is In-Context Learning (ICL), as it can enable the LLMs to perform well on the tasks or data that they have never seen before [86]. In ICL, a prompt containing an instruction, a few labeled samples, and an unlabeled sample is given to the LLM. Then, the LLM would be able to label the unlabeled sample without the need for any gradient-based training [90]. With this approach, the LLM can generate responses tailored to the specific input context they receive. By considering the context provided in the input text, the LLM can generate responses that are not only more informative but also more relevant to what the user asks.

However, it is infeasible to show all the available samples to the LLM due to the high cost of computation. Also, previous research shows that the format of the prompt, the selection of samples, the number, order, and structure of samples could have not only significant but also unforeseeable effects on LLMs' performance [90]–[93].

To solve the aforementioned problems, we propose Semantic In-Context Learning (S-ICL) [15] which utilizes a semantic search engine (i.e., an SBERT model [14]) and an LLM (i.e., ChatGPT with the gpt-3.5-turbo engine) to build a conversational agent. This agent not only benefits from the knowledge of an LLM but also utilizes available private knowledge sources to

provide the correct answer to the inquiries. We also propose a flexible architecture that allows experts to apply and compare different approaches to prompt engineering. The proposed model is developed and participated in the BEA 2023 Shared task [16]. The proposed model is flexible, and the agent can be used in other domains such as news media, customer service, etc.

Contributions of this work are as follows:

- We propose Semantic In-Context Learning, which enables Large Language Models (LLMs) to efficiently identify the most relevant samples from the in-domain corpus to answer inquiries.
- While fine-tuning on in-domain data could significantly enhance the performance of language models, our proposed architecture allows LLMs to learn from the data without the need for any gradient-based training [16].

The remaining portion of the chapter is structured as follows:

Subsection 6.3 elaborates on the proposed architecture, providing insights into its components. Subsection 6.4 entails a comparison of various configurations of the proposed model using the generated test set. Additionally, the model's performance is assessed using the private competition data in Subsection 6.5. Subsection 6.6 elaborates on the importance of headlines in news conversational agents. Finally, Subsection 6.7 presents the conclusion of this paper, summarizing the findings and implications of the study.

6.2. Problem Formulation

Given a conversation between two parties (such as a teacher and a student), which consists of a list of utterances, our goal is to generate an answer a by considering the last utterance as the

query q and the utterances before the last one in the conversation as the context c. We divide the problem into two sub-problems. First, given query q and the context c, find the top k most relevant contexts s_1, s_2, …, s_k from a training corpus D. Second, given q, c, and s_1, s_2, …, s_k, generate a using an LLM.

6.3. Proposed Model

In this section, we present our proposed approach to generating a response to the inquiry. Our proposed approach uses semantic search [14] to enable the agent to utilize private domain data. It also uses a large language model not only to provide higher quality answers but also to enable the agent to answer questions that are significantly different from past QAs in the private domain data.

6.3.1. **Overview**

As shown in Figure 6-1, the proposed architecture consists of five main components: Data Pre-Processor, Embedder, Retriever, Prompt Builder, and Answer Generator. The first three components are related to the semantic search part of the architecture, while the other two are related to the language model.

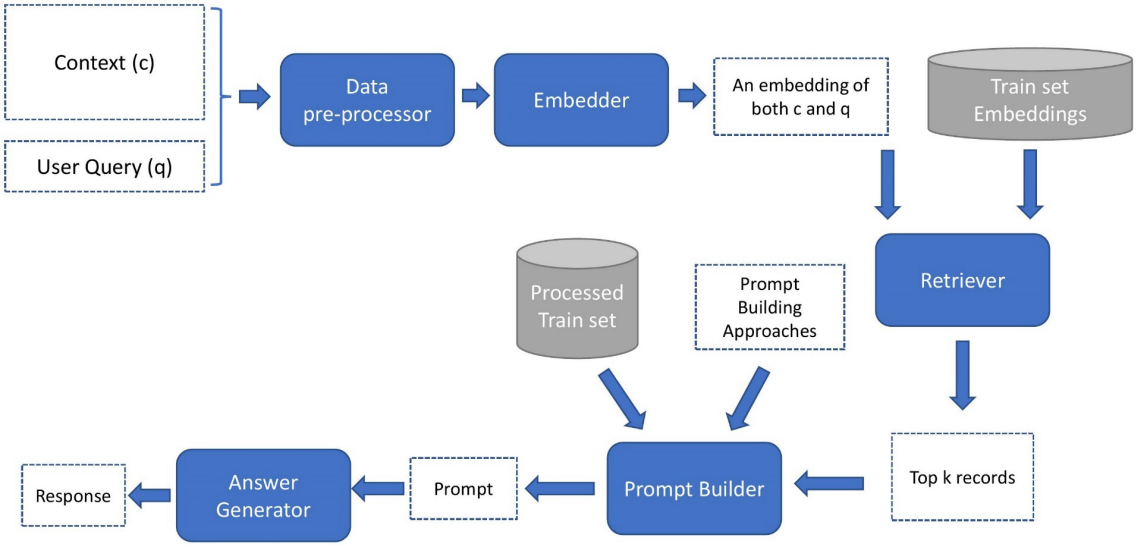

Figure 6-1 The proposed architecture for the conversational agent using both semantic search and a large language model.

6.3.2. **Data Pre-Processor**

The Data Pre-Processor receives utterances in the JSON format containing a context and a query (i.e., the last utterance). It extracts and transforms the JSON file into the followings:

Concatenation: it's a textual concatenation of all the utterances made by a student and a teacher. The main purpose of transforming data into this format is to enable its use in the semantic search part of the architecture.

Sample: It's a conversational flow between the student and the teacher. Based on who wrote the utterance, either "Teacher: " or "Student: " would be appended in the beginning of the utterance. This format is being used by the prompt builder component as it is more appropriate to be used by the language model.

6.3.3. **Embedder**

In this section, we use a state-of-the-art transformer encoder model to convert the concatenation of utterances (i.e., all the utterances in the provided chat between the student and the teacher), which is built by the Data Pre-Processor, into the embedding representation. The tokenizer first tokenizes the input text, and then the transformer encoder model infers an embedding vector with a size of 768 for each token of the input text. The embedding vector of the CLS token in the last layer is considered the embedding representation of the whole input text.

6.3.4. **Retriever**

The Retriever is responsible for finding the most similar records that exist in the training data to the incoming context. It calculates the cosine similarity between the embedding vector of the context and each embedding vector in the training set. Then, the results would be sorted in descending order based on the cosine similarity score, and the top *k* results would be passed on to the next step.

This process could be significantly sped up on large datasets by using approximate K-nearest neighbor methods, such as Facebook AI Similarity Search (FAISS) [94]. However, due to the small size of our data, we don't need to use any approximate K-NN methods.

6.3.5. **Prompt Builder**

The Prompt Builder component creates a prompt based on the selected prompt building approach. Figure 6-2 shows the structure of the prompt which consists of the following components in order:

Command: It's a first component of the prompt that informs the language model of what is expected to be done.

Sample(s): The retrieved sample(s) from the training set are included to assist the language model in answering the inquiry. This part of the prompt is optional because the number of samples to be used depends on the selected approach.

Inquiry: It contains the last utterance along with the previous utterances (i.e., Context) given to the system.

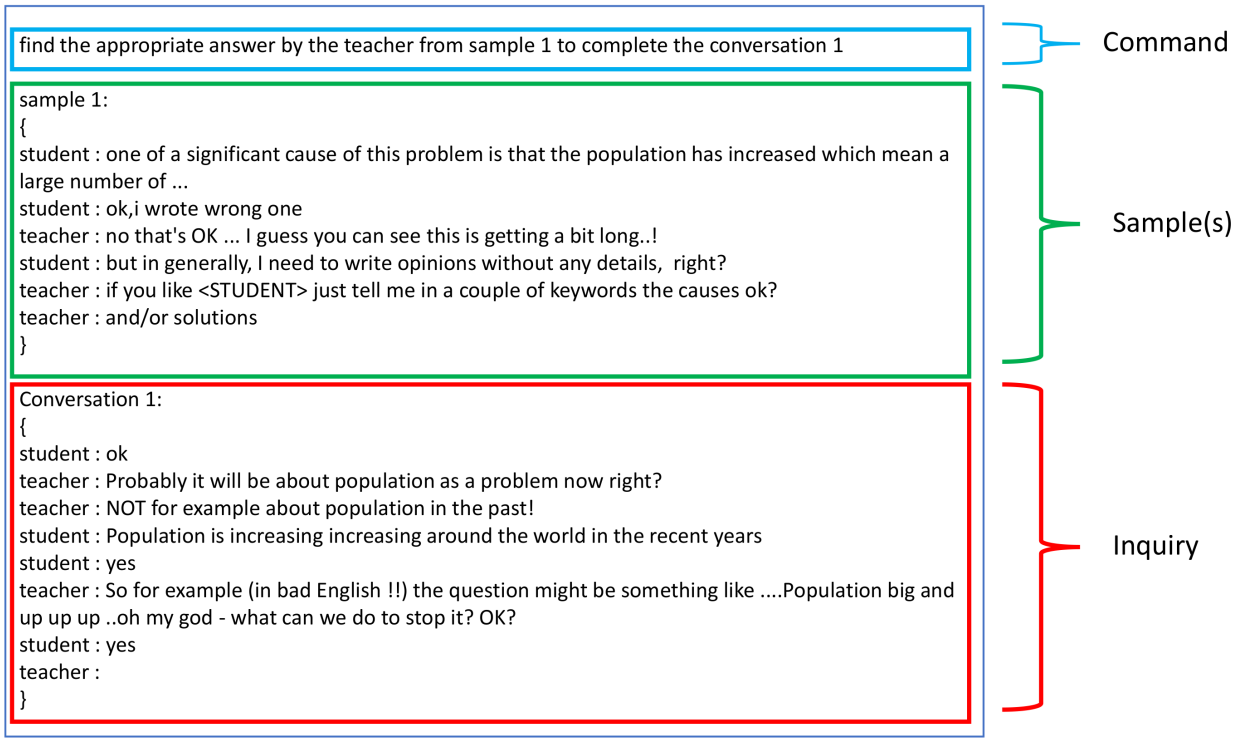

Figure 6-2 An example of a prompt structure with a single sample.

The command part of the prompt is written by humans, whereas the remaining parts are automatically generated depending on the chosen prompt building approach. So far, four prompt building approaches have been designed, as outlined in subsection 6.4.2. However, additional approaches could be defined to further improve the agent's performance or better tailor it to diverse data domains, such as news.

6.3.6. **Answer Generator**

In this part of the architecture, a large language model (LLM) is utilized. In our experiment, we leverage the ChatGPT API with the gpt-3.5-turbo engine, although other LLMs capable of understanding prompts could be employed as well. A prompt generated in the previous stage is sent to the language model, and the response is then returned to the end user. To ensure reproducibility of the results, we set the temperature parameter to zero. This action renders the model deterministic, as it favors the most probable token at each step. The language models utilize a variation of the softmax function that includes a temperature parameter. The softmax function and the softmax with temperature parameter are depicted in Formula (6-1) and Formula (6-2), respectively, where x is a vector with a size of N, and T represents the temperature parameter.

$$softmax(x)_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \tag{6-1}$$

$$softmax(x,T)_i = \frac{e^{x_i/T}}{\sum_{j=1}^{N} e^{x_j/T}} \tag{6-2}$$

In the modified version of softmax (Formula (6-2)), each element would be divided by the temperature parameter. The likelihood of tokens during the generation process is influenced by temperature. As we can see in Formula (6-2), the lower the value of the temperature parameter, the sharper the distribution would be, making the most probable element more dominant. A temperature of zero would render the model entirely deterministic, favoring the most probable token at each step.

By retrieving the most relevant samples and including them in the prompt, the language model can not only utilize its knowledge but also access relevant past responses from private domain knowledge to answer the question. Another advantage is that there's no need to fine-tune the large language model on private internal data, which may not be an option for many LLMs.

6.4. Experiment

This section has three subsections. In the first part, we introduce the dataset used, split the training portion of the data into our created train and test sets, and show how the pre-processing has been done. In the second subsection, we conduct experiments on the proposed architecture using the created test set (i.e., selected from the original training set) and compare the accuracy of the model using different prompt building approaches. In the third subsection, we will demonstrate the model's performance on the development and testing sets of the competition data.

6.4.1. **Data**

The data consists of a conversation between a student and a teacher provided by [95]. We choose this dataset because we require a conversational data for this section. Since such data is not available within the context of news, we opted to utilize this dataset. Nevertheless, our proposed architecture could also be applied to conversational data between a news media agent and a user.

The size (i.e., the number of conversations between students and teachers) of the provided data and their release dates in the competition are shown in Table 6-1.

Table 6-1 The statistics of the TSCC dataset.

Set	Size (number of conversations)	Release date
Train	2747	March 24, 2023
Dev	305	March 24, 2023
Test	273	May 1, 2023

We transform the training set using the pre-processor component (subsection 6.3.2). Then, we use the embedder component (subsection 6.3.3) to convert the concatenation of the utterances into their embedding representations (i.e., Figure 6-1).

Then, we split the training set into customized training and test sets with sizes of 2647 and 100, respectively. We use the customized training and test sets to compare the different prompt generation approaches in subsection 6.4.2. Since some of the records in the training set have similar utterances (i.e., they overlap), we select the test data in a way that none of the test conversations can be answered directly from the conversations in the training set (i.e., there is no overlap between the utterances of the training and test sets).

6.4.2. **Evaluation of Different Approaches**

We employ five different approaches to generate responses to incoming inquiries. In the first approach, we rely solely on semantic search, selecting the retrieved sample most similar to the last utterance as the response. We adopt this approach to assess how effectively our model performs when using only semantic search to find the most relevant result and considering it as the response. This entails utilizing exact phrases from our dataset without any modification or generation.

Next, we are curious to see how effective the language model is in completing the conversation without relying on any samples. The prompt command given to the language model is "*Complete the following conversation by providing an appropriate response as the teacher.*" In this scenario, we would evaluate the performance of the generative component of our architecture in generating a response without receiving any additional information about the data in the prompt. In this manner, the language model would answer the student's question solely based on its own knowledge.

In the third approach, we prompt the language model to "*Find the appropriate response from sample 1 to complete the conversation 1."* The provided sample, identified as "train_0063" (refer to Figure 6-3), was selected from the training set and is utilized for all queries. In this scenario, our objective is to evaluate whether providing a sample from our dataset, to acquaint the LLM with the format of a complete conversation, enhances its ability to generate a better response.

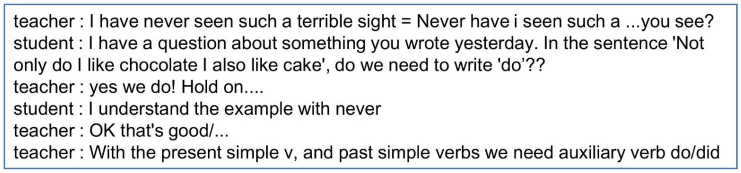

Figure 6-3 A sample from the training set used in the prompt with the ID "train_0063"

During the experiments, we observed that ChatGPT tends to generate longer responses than the ground truths. However, we discovered that by formulating our prompt command in a certain way (i.e., find the appropriate answer by the teacher from sample ...), ChatGPT can produce more concise and shorter responses. Therefore, we decided to write the command part of our prompt in this way. We also observed that for some inquiries, ChatGPT mentions "teacher :" in its response, so we wrote a rule to remove it.

The fourth approach involves incorporating the top 3 most similar samples into the prompt, with the command being "*Find the appropriate response from sample 1, sample 2, and sample 3 to complete the conversation 1.*" Lastly, the last approach mirrors the third one, but instead of utilizing a curated sample, the most similar sample from the training set is employed. Through these two approaches, we aim to explore whether presenting similar samples to the question aids the LLM in generating a better response. The last two approaches are based on S-ICL.

The results of the aforementioned approaches on the generated test dataset are presented in Table 6-2, evaluated in terms of BERTScore [12] and DialogRPT [96]. These metrics were selected by ACL to evaluate the responses generated by the models and compare them with the private ground truth data. It's worth noting that the private data was not disclosed, and access was limited to the BEA workshop.

DialogueRPT utilizes a set of pre-trained GPT-2 models to assess the quality of the generated responses based on the following measures. To calculate the value of each mentioned measure below, a model is employed where the input is a generated response, and the output is the corresponding value for that measure:

- Updown (U): the probability that a response receives upvotes.
- Human vs Random (HvR): the probability that the response is relevant to the given context.
- Human vs Machine (HvM): the probability that the response was written by a human rather than generated by a machine.

In Table 6-2 , P, R, F, U, HvR, HvM stand for precision, recall, f1-score, updown, human vs random, human vs machine, respectively. The first three measures belong to BERTScore, while the remaining measures belong to DialogRPT.

The pre-trained DialogRPT models for each measure are available on HuggingFace [16]. However, for BERTScore, there is no specified model as any pre-trained transformer encoder models could be utilized for this purpose. As BEA workshop did not specify which type of transformer encoder model they used to in BERTScore, we opeted to use the "roberta-large" model[17] as it was recommended in the GitHub account[18] of BERTScore for our evaluation in

[16] https://huggingface.co/microsoft/DialogRPT-depth

[17] https://huggingface.co/roberta-large

[18] https://github.com/Tiiiger/bert_score

Table 6-2. However, BEA workshop might use another type of transformer encoder model for the results represented in Table 6-3 and Table 6-4.

Table 6-2 The comparison between different approaches used on the created test set.

Approach	BERTScore			DialogRPT				
	P	R	F	U	HvR	HvM	Best	average
1	0.824	0.823	0.823	0.433	0.789	0.983	0.999	0.735
2	0.835	**0.837**	0.836	0.490	**0.922**	**0.999**	0.999	**0.804**
3	**0.839**	0.836	**0.837**	0.464	0.877	0.997	0.998	0.779
4	0.828	0.832	0.830	**0.574**	0.767	0.998	0.999	0.780
5	0.835	0.831	0.833	0.481	0.839	0.997	**0.999**	0.773

As we can observe in Table 6-2, utilizing the language model alone (approach 2) can outperform the semantic search approach (approach 1) in terms of all the metrics employed. This is an intriguing observation: despite semantic search leveraging the most pertinent available answers from analogous conversations in the past to address the inquiry, the language model, without access to previous data, does a commendable job by surpassing the performance of semantic search.

The model that uses the fixed sample for all the queries (the third approach) gained the best BERTscore in terms of F1-score. This observation is in line with the results of other studies such as [91] that they concluded replacing the sample labels randomly would barely hurts the performance of the LLMs.

In terms of the average of best scores for DialogRPT, the fifth approach is the best. However, by considering the average of the measures, the second approach obtained the best results. However, when we examined the generated answers, we found out the answers of the fifth approach are both more reasonable and preferable in comparison with the other approaches.

6.5. BEA Workshop's Evaluation

Our proposed approach ranked fourth both in development (Table 6-3) and evaluation (Table 6-4) phases of BEA workshop shared task 2023.

Table 6-3 BEA Workshop's development phase

	user	**BERTScore**			**DialogRPT**				
		P	**R**	**F**	**U**	**HvR**	**HvM**	**Avg**	**Best**
1	adaezeio	0.67	**0.71**	**0.69**	0.37	**0.98**	0.99	0.35	0.71
2	justinray-v	0.65	0.7	0.67	**0.42**	0.96	**1**	**0.4**	0.76
3	ignacio.sastre	0.68	0.69	0.68	0.37	0.95	0.98	0.33	0.72
4*	Our Approach	0.67	0.69	0.68	0.39	0.9	0.99	0.35	0.76
5	pykt-team	0.67	0.71	0.68	0.36	0.91	0.99	0.33	0.72
6	ThHuber	0.68	0.65	0.66	0.38	0.93	0.98	0.34	0.68
7	alexis.baladon	**0.7**	0.67	0.68	0.35	0.92	0.98	0.3	0.68
8	a2un	0.67	0.65	0.66	0.38	0.93	0.95	0.33	**0.81**
9	rbnjade1	0.62	0.62	0.62	0.37	0.84	0.99	0.31	0.7
10	tanaygahlot	0	0	0	0.35	0.88	0.97	0.3	0.6

Table 6-4 BEA Workshop's evaluation phase

	Team	BERTScore			DialogRPT				
		P	**R**	**F**	**U**	**HvR**	**HvM**	**AVG**	**Best**
1	NAIST	0.71	**0.71**	**0.71**	0.48	**0.98**	**1**	0.46	0.67
2	NBU	0.72	0.69	0.71	0.4	0.97	0.98	0.37	0.65
3	Cornell	0.71	0.69	0.7	**0.52**	0.86	0.98	**0.47**	**0.75**
4*	Our Approach	0.72	0.69	0.7	0.4	0.92	0.98	0.36	0.69
5	Retuyt-InCo	**0.74**	0.68	0.71	0.38	0.9	0.96	0.35	0.65
6	Cornell	0.72	0.69	0.7	0.4	0.93	0.98	0.36	0.64
7	Data	0.72	0.63	0.67	0.41	0.93	0.95	0.37	0.76
8	Retuyt-InCo	0.72	0.68	0.7	0.37	0.91	0.96	0.34	0.68
9	DT	0.67	0.62	0.64	0.36	0.75	0.96	0.29	0.62
10	TanTanLabs	0.71	0.6	0.65	0.32	0.85	0.96	0.29	0.65

We used our third approach (using the fixed sample) for the development phase as we noticed the majority of utterances in development data have overlap with the training set. If we use either the fifth or fourth approach, the model would recognize the similarity between the sample and the conversation and produce a response so similar to the existing utterance in the sample that it would inflate the performance of the system.

However, we discovered that the test data is different in a way that none of its conversations could have their responses directly obtained from any utterances in either the training or development sets. Therefore, for the evaluation set, we use the fifth approach. Another reason why we use the fifth approach in the evaluation phase is that the top three models

would be evaluated by the human evaluators, and we already noticed in subsection 6.4.2 that the results of the fifth approach are more desirable from humans' point of view.

The evaluation phase started on May 1st and ended on May 5th, with our model ultimately ranking fourth in the evaluation. We think that the proposed model has a high potential for improvement, especially if more efforts are put on the prompt engineering part of the architecture.

6.6. Importance of Headlines in Conversational News Agents

Although news conversational agents may reshape how people consume news in the future, we believe that headlines will remain an essential component of the news delivery process. Even when interacting with news conversational agents, users may still encounter and rely on headlines to understand and engage with news content. This suggests that the shift from traditional news portals to conversational news agents will not eliminate the need for well-crafted headlines.

One of the most significant aspects of headlines is their ability to provide a concise summary of an article. This property is especially important in a chatbot environment, where users expect quick and clear answers. In a conversation between a user and a chatbot, the chatbot may display a list of headlines from relevant articles in response to the user's query. In such cases, a headline serves as an anchor, giving users a clear indication of the news topic before they decide whether to request more details. Just as readers of online news portals read headlines before diving into an article, users of news chatbots would still want to see headlines before

asking for more content or details about an article. Moreover, displaying the actual headline rather than generating a summary via an LLM helps mitigate risks such as hallucination, where the LLM may produce a summary that misrepresents the facts or spreads misinformation.

Headlines can also play a pivotal role in semantic search within conversational agents. When a user queries a chatbot about a particular topic, the system can utilize headlines to assess the relevance of news articles by performing a semantic comparison between the user's query and the headlines. Since headlines are designed to be concise yet informative, they provide an efficient way to match user queries with relevant news articles.

6.7. Conclusion

We proposed a Semantic In-Context Learning (S-ICM) approach for conversational agents using the combination of a semantic search and a large language model (i.e., ChatGPT). We also implemented an architecture enabling users to apply and compare different approaches for prompt engineering. We applied our proposed method on the BEA 2023 shared task and our approach ended up ranking fourth in both the development and evaluation phases. For future work, the proposed approach could be applied to the news domain. To achieve this goal, prompt engineering would be conducted to customize the model for the news domain. Additionally, a dataset including users' conversations data with agent responses and user feedback should be collected. This dataset can then be used to compare different prompts, models, and more.

Chapter 7
Conclusions and Future Works

7.1. Summary of Contributions

In Chapter 3, we proposed four quality indicators to define the quality of headlines. Our approach can automatically determine the quality of published headlines using the news portal's log history, without the need for human annotation [5]. We labeled The Globe and Mail's dataset using the proposed approach to provide insights regarding the published headlines. Our proposed approach was well-received by The Globe and Mail and has also been adopted by others such as Kaleva Media [6], [7].

We then proposed a novel model for predicting headline quality, which includes a semantic similarity matrix, a transformer encoder, and a topic modeling component. We trained this model using the created labels from our proposed method. The trained model can be used to assess the quality of unpublished headlines and assist authors in selecting the appropriate headline for their article. Additionally, we enhanced the model's quality by incorporating semantic latent features through transfer learning techniques, resulting in improved accuracy [8].

Most of the previous headline popularity prediction models used users' log history on a news portal to calculate the popularity of headlines using measures such as the number of clicks, views, etc. However, the view or click count of headlines, in addition to the headline itself, depends on other hidden factors, such as the personalization algorithm, the position of the title,

the font size, and more, which are not shared in the dataset. Therefore, we created an HP dataset and defined the popularity of a headline as the number of likes it received on Twitter. We used our HP dataset to train a headline popularity prediction model and to create a headline popularity benchmark.

We developed a headline generation model based on the work of [9] and trained it on different metrics using Reinforcement Learning in Chapter 4. This approach optimizes the model not only based on a non-differentiable metric (e.g., ROUGE, METEOR, BERTScore) but also on two metrics simultaneously. We further improved the generated headlines by proposing a novel headline generator architecture in Chapter 5, employing state-of-the-art transformer models. Our model comprises two main components: generation and selection. In the generation part, we fine-tuned transformer models on the headline generation task using the Newsroom dataset. For the selection part, we trained a headline popularity model on the HP dataset. The generative component generates 10 different headlines for each news article using multinomial sampling technique, and the selection component chooses the most popular one. We created a popularity benchmark for evaluation purposes, which automatically assesses how the proposed model can assist in generating headlines with higher popularity.

We predict that conversational agents might soon revolutionize the way people consume news. Many individuals may prefer to stay updated on the latest news by interacting with their AI assistants rather than reading news articles themselves. However, despite the ability of LLMs to answer a wide range of questions, as mentioned in Chapter 6, there are two major obstacles preventing their use for this purpose. Firstly, LLMs are not aware of the internal private data of the news media. Secondly, news articles are published daily, making it difficult and costly to periodically fine-tune LLMs on new data. Moreover, certain LLMs, cannot be fine-tuned. To

address these issues, we proposed S-ICM [15], which combines a semantic search engine with an LLM. Our proposed approach participated in BEA's shared task [16] and achieved the fourth rank in both the development and evaluation phases. Furthermore, our approach is flexible and can be adapted to other domains, such as news media.

7.2. Future Directions

For future works on this thesis, the following directions can be explored:

- One of the limitations of our research in headline generation (Chapter 5) is that we only trained both the popularity prediction and headline generation models on English news articles. Future research can focus on training multilingual models that can be used for articles written in any language.
- Another limitation of our headline generation model (Chapter 5) is that we did not explore the effectiveness of large language models (LLMs) in the task. Future research should investigate the potential of prominent language models like GPT-4 for headline generation. Due to their significantly larger number of parameters, LLMs can learn much more from the data, allowing them to capture complex linguistic patterns and generate more fluent and contextually relevant headlines. Additionally, these models benefit greatly from extensive pretraining on massive amounts of data, which enhances their ability to generalize across different news topics and styles. However, a significant challenge in evaluating the effectiveness of LLMs in this task is ensuring that the models have not been exposed to the

evaluation dataset during their pre-training, as the news article benchmarks (e.g., Newsroom) are publicly accessible on the Internet.

- For future work, the proposed S-ICL method (Chapter 6) can be applied to the News Domain. To achieve this goal, prompt engineering would be conducted to customize the model for the news domain. Additionally, a dataset including users' conversations data with agent responses and user feedback should be collected. This dataset can then be used to compare different prompts, models, and more.

Bibliography

[1] S. Wang, E.-H. Han, and A. Rush, “Headliner: An integrated headline suggestion system,” in *Computation+ Journalism Symposium California. https://goo. gl/9fvdx4*, 2016.

[2] O. Vasilyev, T. Grek, and J. Bohannon, “Headline Generation: Learning from Decomposed Document Titles,” *arXiv Prepr. arXiv1904.08455*, 2019.

[3] K. Alnajjar, L. Leppänen, and H. Toivonen, “No Time Like the Present: Methods for Generating Colourful and Factual Multilingual News Headlines,” in *The 10th International Conference on Computational Creativity*, 2019, pp. 258–265.

[4] S. Chopra, M. Auli, and A. M. Rush, “Abstractive sentence summarization with attentive recurrent neural networks,” *2016 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. NAACL HLT 2016 - Proc. Conf.*, pp. 93–98, 2016.

[5] A. Omidvar., H. Pourmodheji., A. An., and G. Edall., “Learning to Determine the Quality of News Headlines,” in *Proceedings of the 12th International Conference on Agents and Artificial Intelligence - Volume 1: NLPinAI,* 2020, pp. 401–409.

[6] J. Tervonen, T. Sormunen, A. Lämsä, J. Peltola, H. Kananen, and S. Järvinen, “Predicting Headline Effectiveness in Online News Media using Transfer Learning with BERT.,” in *DeLTA*, 2021, pp. 29–37.

[7] G. Song, Y. Wang, X. Chen, H. Hu, and F. Liu, “Evaluating User Engagement in Online News: A Deep Learning Approach Based on Attractiveness and Multiple Features,” *Systems*, vol. 12, no. 8, p. 274, 2024.

[8] A. Omidvar, H. Pourmodheji, A. An, and G. Edall, “A Novel Approach to Determining the Quality of News Headlines,” *Nat. Lang. Process. Artif. Intell. 2020*, vol. 939, p. 227, 2021.

[9] R. Paulus, C. Xiong, and R. Socher, “A Deep Reinforced Model for Abstractive Summarization,” no. i, pp. 1–12, 2017.

[10] C. Y. Lin, “Rouge: A package for automatic evaluation of summaries,” *Proc. Work. text Summ. branches out (WAS 2004)*, no. 1, pp. 25–26, 2004.

[11] S. Banerjee and A. Lavie, “METEOR: An automatic metric for MT evaluation with improved correlation with human judgments,” in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72.

[12] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, “BERTSCORE: EVALUATING TEXT GENERATION WITH BERT,” in *8th International Conference on Learning Representations, ICLR 2020*, 2020.

[13] A. Omidvar and A. An, “Learning to Generate Popular Headlines,” *IEEE Access*, vol. 11, pp. 60904–60914, 2023.

[14] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using siamese BERT-networks,” in *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2019, pp. 3982–3992.

[15] A. Omidvar and A. An, “Empowering Conversational Agents using Semantic In-Context Learning,” in *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, 2023, pp. 766–771.

[16] A. Tack, E. Kochmar, Z. Yuan, S. Bibauw, and C. Piech, “The BEA 2023 Shared Task on Generating AI Teacher Responses in Educational Dialogues,” in *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications*, 2023, p. to appear.

[17] J. H. Kim, A. Mantrach, A. Jaimes, and A. Oh, “How to Compete Online for News Audience,” pp. 1645–1654, 2016.

[18] T. Szymanski, C. Orellana-Rodriguez, and M. T. Keane, “Helping news editors write better headlines: A recommender to improve the keyword contents \& shareability of news headlines,” *arXiv Prepr. arXiv1705.09656*, 2017.

[19] J. Reis, F. Benevenuto, P. O. S. V. de Melo, R. Prates, H. Kwak, and J. An, “Breaking the News: First Impressions Matter on Online News,” pp. 357–366, 2015.

[20] R. Bandari, S. Asur, and B. A. Huberman, “The pulse of news in social media: Forecasting popularity,” in *Sixth International AAAI Conference on Weblogs and Social Media*, 2012.

[21] A. Piotrkowicz, V. Dimitrova, J. Otterbacher, and K. Markert, “Headlines matter: Using headlines to predict the popularity of news articles on twitter and facebook,” in *Eleventh International AAAI Conference on Web and Social Media*, 2017.

[22] B. D. Horne and S. Adali, “The impact of crowds on news engagement: A reddit case study,” in *Eleventh International AAAI Conference on Web and Social Media*, 2017.

[23] Y. Mao, M. Chen, A. Wagle, J. Pan, M. Natkovich, and D. Matheson, “A Batched Multi-Armed Bandit Approach to News Headline Testing,” in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 1966–1973.

[24] A. Voronov, Y. Shen, and P. K. Mondal, “Forecasting popularity of news article by title analyzing with BN-LSTM network,” in *Proceedings of the 2019 International*

Conference on Data Mining and Machine Learning, 2019, pp. 19–27.

[25] W. Stokowiec, T. Trzciński, K. Wołk, K. Marasek, and P. Rokita, "Shallow reading with deep learning: Predicting popularity of online content using only its title," in *International Symposium on Methodologies for Intelligent Systems*, 2017, pp. 136–145.

[26] J. Wang, S. Yang, H. Zhao, and Y. Yang, "Social media popularity prediction with multimodal hierarchical fusion model," *Comput. Speech \& Lang.*, vol. 80, p. 101490, 2023.

[27] Y. L. Lopez, D. Grimaldi, S. Garcia, J. Ordoez, C. Carrasco-Farre, and A. A. Aristizabal, "Artificial intelligence model to predict the virality of press articles," in *2022 14th International Conference on Machine Learning and Computing (ICMLC)*, 2022, pp. 221–228.

[28] W. Wei and X. Wan, "Learning to identify ambiguous and misleading news headlines," *IJCAI Int. Jt. Conf. Artif. Intell.*, pp. 4172–4178, 2017.

[29] R. Mihalcea and P. Tarau, "TextRank: Bringing order into texts," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP 2004 - A meeting of SIGDAT, a Special Interest Group of the ACL held in conjunction with ACL 2004*, 2004, pp. 404–411.

[30] S.-Q. Shen *et al.*, "Recent advances on neural headline generation," *J. Comput. Sci. Technol.*, vol. 32, no. 4, pp. 768–784, 2017.

[31] T. Higurashi, H. Kobayashi, T. Masuyama, K. Murao, † Yahoo, and J. Corporation, "Extractive Headline Generation Based on Learning to Rank for Community Question Answering," *Proc. 27th Int. Conf. Comput. Linguist.*, pp. 1742–1753, 2018.

[32] S. Xu, S. Yang, and F. Lau, "Keyword extraction and headline generation using novel

word features," in *Proceedings of the AAAI conference on artificial intelligence*, 2010, vol. 24, no. 1, pp. 1461–1466.

[33] G. Klein, Y. Kim, Y. Deng, J. Crego, J. Senellart, and A. M. Rush, "OpenNMT: Open-source toolkit for neural machine translation," in *20th Annual Conference of the European Association for Machine Translation, EAMT 2017*, 2017, p. 22.

[34] S. Takase, J. Suzuki, N. Okazaki, T. Hirao, and M. Nagata, "Neural headline generation on abstract meaning representation," in *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 2016, pp. 1054–1059.

[35] R. Nallapati, B. Zhou, C. dos Santos, Ç. Gulçehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence RNNs and beyond," in *CoNLL 2016 - 20th SIGNLL Conference on Computational Natural Language Learning, Proceedings*, 2016, pp. 280–290.

[36] J. D. M.-W. C. Kenton and L. K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[37] X. Ao, X. Wang, L. Luo, Y. Qiao, Q. He, and X. Xie, "PENS: A Dataset and Generic Framework for Personalized News Headline Generation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 82–92.

[38] J. Zhang, Y. Zhao, M. Saleh, and P. Liu, "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," in *International Conference on Machine Learning*, 2020, pp. 11328–11339.

[39] M. Ailem, B. Zhang, and F. Sha, “Topic Augmented Generator for Abstractive Summarization,” *arXiv Prepr. arXiv1908.07026*, 2019.

[40] K. M. Hermann *et al.*, “Teaching machines to read and comprehend,” in *Advances in neural information processing systems*, 2015, pp. 1693–1701.

[41] C. Napoles, M. Gormley, and B. Van Durme, “Annotated gigaword,” in *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, 2012, pp. 95–100.

[42] L. Leppänen, M. Munezero, M. Granroth-Wilding, and H. Toivonen, “Data-driven news generation for automated journalism,” in *Proceedings of the 10th International Conference on Natural Language Generation*, 2017, pp. 188–197.

[43] J. Kuiken, A. Schuth, M. Spitters, and M. Marx, “Effective Headlines of Newspaper Articles in a Digital Environment,” *Digit. Journal.*, vol. 5, no. 10, pp. 1300–1314, 2017.

[44] D. Cer *et al.*, “Universal sentence encoder,” *arXiv Prepr. arXiv1803.11175*, 2018.

[45] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global vectors for word representation,” in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014.

[46] R. M. NA and S. Veni, “A stacked ensemble technique with glove embedding model for depression detection from tweets,” *Indian J. Comput. Sci. Eng.*, vol. 10, pp. 586–595, 2022.

[47] A. Onan, “Mining opinions from instructor evaluation reviews: a deep learning approach,” *Comput. Appl. Eng. Educ.*, vol. 28, no. 1, pp. 117–138, 2020.

[48] C. Ammer and L. Grüner, “UniTuebingenCL at SemEval-2020 task 7: Humor detection in news headlines,” in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*,

2020, pp. 1060–1065.

[49] A. Rafay, M. Suleman, and A. Alim, “Robust review rating prediction model based on machine and deep learning: Yelp dataset,” in *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*, 2020, pp. 8138–8143.

[50] M. Narasimhan, S. Lazebnik, and A. Schwing, “Out of the box: Reasoning with graph convolution nets for factual visual question answering,” *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.

[51] H. Yan, B. Deng, X. Li, and X. Qiu, “TENER: adapting transformer encoder for named entity recognition,” *arXiv Prepr. arXiv1911.04474*, 2019.

[52] K. Lopyrev, “Generating News Headlines with Recurrent Neural Networks,” pp. 1–9, 2015.

[53] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng, “Text Matching as Image Recognition,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016, vol. 30, no. 1.

[54] G. E. Dahl, T. N. Sainath, and G. E. Hinton, “Improving deep neural networks for LVCSR using rectified linear units and dropout,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2013, pp. 8609–8613.

[55] A. Vaswani *et al.*, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[56] J. Kim, Y. He, and H. Park, “Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework,” *J. Glob. Optim.*, vol. 58, no. 2, pp. 285–319, Feb. 2014.

[57] M. D. Hoffman, D. M. Blei, and F. Bach, “Online learning for Latent Dirichlet

Allocation," in *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010, NIPS 2010*, 2010.

[58] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative Study of CNN and RNN for Natural Language Processing," *arXiv Prepr. arXiv1702.01923*, 2017.

[59] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, 2014, pp. II–1188.

[60] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[61] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[62] C. Raffel *et al.*, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *J. Mach. Learn. Res.*, vol. 21, pp. 1–67, 2020.

[63] M. Zaheer *et al.*, "Big bird: Transformers for longer sequences," in *Advances in Neural Information Processing Systems*, 2020, vol. 2020-Decem.

[64] M. Lewis *et al.*, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7871–7880.

[65] J. Cheng, Z. Wang, and G. Pollastri, "A neural network approach to ordinal regression," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008, pp. 1279–1284.

[66] J. Yao, "Personalized News Headline Generation System with Fine-grained User

Modeling," in *2022 18th International Conference on Mobility, Sensing and Networking (MSN)*, 2022, pp. 581–585.

[67] Z. Li, J. Wu, J. Miao, and X. Yu, "News headline generation based on improved decoder from transformer," *Sci. Rep.*, vol. 12, no. 1, p. 11648, 2022.

[68] Z. Fang *et al.*, "How to generate popular post headlines on social media?," *AI Open*, vol. 5, pp. 1–9, 2024.

[69] S. Shalev-Shwartz and T. Zhang, "Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization," *Math. Program.*, vol. 155, no. 1–2, pp. 105–145, 2016.

[70] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *3rd International Conference on Learning Representations*, 2014.

[71] A. Radford *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[72] M. T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1412–1421.

[73] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7008–7024.

[74] D. Gavrilov, P. Kalaidin, and V. Malykh, "Self-Attentive Model for Headline Generation," in *European Conference on Information Retrieval*, 2019, pp. 87–93.

[75] C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio, "Pointing the unknown words," in *54th Annual Meeting of the Association for Computational Linguistics, ACL*

2016 - Long Papers, 2016, vol. 1, pp. 140–149.

[76] Y. Keneshloo, T. Shi, N. Ramakrishnan, and C. K. Reddy, "Deep reinforcement learning for sequence-to-sequence models," *IEEE Trans. neural networks Learn. Syst.*, vol. 31, no. 7, pp. 2469–2489, 2019.

[77] M. Grusky, M. Naaman, and Y. Artzi, "Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies," in *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 2018, vol. 1, pp. 708–719.

[78] Y. Keneshloo, S. Wang, E.-H. Han, and N. Ramakrishnan, "Predicting the popularity of news articles," in *Proceedings of the 2016 SIAM International Conference on Data Mining*, 2016, pp. 441–449.

[79] K. Fernandes, P. Vinagre, and P. Cortez, "A proactive intelligent decision support system for predicting the popularity of online news," in *Portuguese conference on artificial intelligence*, 2015, pp. 535–546.

[80] C. Liu, W. Wang, Y. Zhang, Y. Dong, F. He, and C. Wu, "Predicting the popularity of online news based on multivariate analysis," in *2017 IEEE international conference on computer and information technology (CIT)*, 2017, pp. 9–15.

[81] E. Hensinger, I. Flaounas, and N. Cristianini, "Modelling and predicting news popularity," *Pattern Anal. Appl.*, vol. 16, no. 4, pp. 623–635, 2013.

[82] S. Natarajan and M. Moh, "Recommending news based on hybrid user profile, popularity, trends, and location," in *2016 international conference on collaboration technologies and systems (CTS)*, 2016, pp. 204–211.

[83] F. Long, M. Xu, Y. Li, Z. Wu, and Q. Ling, "XiaoA: A robot editor for popularity

prediction of online news based on ensemble learning," in *International Conference on Intelligence Science*, 2018, pp. 340–350.

[84] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*, 2002, pp. 311–318.

[85] W. Qi *et al.*, "ProphetNet: Predicting Future N-gram for Sequence-to-SequencePre-training," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 2401–2410.

[86] T. Brown *et al.*, "Language models are few-shot learners," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 1877–1901, 2020.

[87] D. Jin, Z. Jin, J. T. Zhou, L. Orii, and P. Szolovits, "Hooks in the headline: Learning to generate headlines with controlled styles," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 5082–5093.

[88] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv Prepr. arXiv1711.05101*, 2017.

[89] L. E. Dor *et al.*, "Learning thematic similarity metric from article sections using triplet networks," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018, pp. 49–54.

[90] H. Liu *et al.*, "Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 1950–1965, 2022.

[91] S. Min *et al.*, "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?," *arXiv Prepr. arXiv2202.12837*, 2022.

[92] V. Sanh *et al.*, "Multitask Prompted Training Enables Zero-Shot Task Generalization," in

International Conference on Learning Representations, 2021.

[93] J. Wei *et al.*, "Larger language models do in-context learning differently," *arXiv Prepr. arXiv2303.03846*, 2023.

[94] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, 2019.

[95] A. Caines *et al.*, "The Teacher-Student Chatroom Corpus," *Proc. 9th Work. Nat. Lang. Process. Comput. Assist. Lang. Learn. (NLP4CALL 2020)*, vol. 175, pp. 10–20, 2020.

[96] X. Gao, Y. Zhang, M. Galley, C. Brockett, and B. Dolan, "Dialogue response ranking training with large-scale human feedback data," in *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2020, pp. 386–395.