Computational Analysis and Mitigation of Textual Media Bias

A dissertation presented by
Wei-Fan Chen

to the
Faculty of Computer Science, Electrical Engineering, and Mathematics
of Paderborn University

in partial fulfillment of the requirements for the academic degree of
Dr. rer. nat.

Paderborn, Germany
December 2023

Dissertation

Computational Analysis and Mitigation of Textual Media Bias

Wei-Fan Chen, Paderborn University

Paderborn, Germany, 2023

Reviewers

Prof. Dr. Henning Wachsmuth, Leibniz University Hannover

Prof. Dr. Benno Stein, Bauhaus Universität Weimar

Doctoral Committee

Prof. Dr. Henning Wachsmuth, Leibniz University Hannover

Prof. Dr. Benno Stein, Bauhaus Universität Weimar

Prof. Dr. Axel-Cyrille Ngonga Ngomo, Paderborn University

Jun.-Prof. Dr. Sebastian Peitz, Paderborn University

Prof. Dr. Stefan Sauer, Paderborn University

My last salutations are to them who knew me imperfect and loved me.

—Rabindranath Tagore.

Acknowledgments

I want to express my deepest appreciation to Professor Henning Wachsmuth, my supervisor, who always supports me and guides me including but not limited to research, but also my life in Germany. Additionally, my heartfelt thanks go to Professor Benno Stein, who supported me in my initial years in Weimar. Without the two professors, I could not achieve so many things here. Your opinions, suggestions, and research methodologies have significantly shaped my research career. I extend my appreciation to Prof. Axel-Cyrille Ngonga Ngomo, Prof. Sebastian Peitz, and Prof. Stefan Sauer for their invaluable contributions as members of my doctoral committee.

I am grateful for the unwavering support of my beloved, Mei-Hua, particularly during the final months of my PhD journey. Your encouragement consistently motivates me and enriches my life in a foreign land.

This thesis includes my publications when I was in Weimar and Paderborn. I would like to take the chance to express my gratitude to Weimar colleagues: Michael Völske, who is my beloved office mate in Weimar; Yamen Ajjour, with whom I have my first publication during my PhD; Khalid Al Khatib who contributed a lot to many research; and engaging discussion partners Johannes Kiesel, Shahbaz Syed, and Roxanne El Baff. Also I thank all my colleagues in Paderborn: Milad Alshomary, with whom I collaborated on numerous outstanding papers; Maximilian Spliethöver, who provided invaluable assistance with the German language, especially in translating the abstract of this thesis; Zahra Nouri, Timon Ziegenbein, Maja Stahl, and Meghdut Sengupta, and Gabriella Skitalinska, with whom I shared many enjoyable moments. My PhD journey would have been monotonous without the rich contributions of these colleagues.

Abstract

Computational Analysis and Mitigation of Textual Media Bias

Media plays a vital role in shaping public opinion, as it serves as a primary source of communication in modern society. However, *media bias*, occurring when media exhibit favoritism or discrimination in their reporting, may potentially influence people in undesirable directions by presenting misinformed or polarized views. In light of the recent advancements in machine learning, natural language processing (NLP) has become a powerful instrument for tackling media bias in terms of detecting or changing the bias in texts. With this in mind, this dissertation delves into the comprehensive examination of media bias, including corpora creation, computational model development, and both quantitative and qualitative analyses. In particular, we study three different kinds of media biases, namely *gatekeeping bias* (selection of what to report), *coverage bias* (visibility of each side), and *statement bias* (opinions of political sides).

To address media bias with the help of NLP, we study three main research topics.

(1) Media bias analysis. We create a media bias corpus focusing on bias in news articles. Leveraging this corpus, our initial step involves the examination of biased language within news articles. Subsequently, we employ neural NLP models to detect bias in the texts. Our study yields empirical evidence demonstrating that detecting article-level bias can be achieved by knowing lower-level (such as sentence-level) biases, and vice versa.

(2) Media bias mitigation. Once we have gained an understanding of the characteristics of media bias, our focus shifts to exploring strategies for mitigating the bias. For the three types of media biases previously mentioned, we propose our strategies to mitigate them. Specifically, we cast them as natural language generation tasks. To combat gatekeeping bias, our model learns to fluently integrate new information into existing texts. To tackle coverage bias, we propose to rewrite the text from the perspective of the other side. Lastly when confronting statement bias, our objective is to alter the political stance of the texts, thereby fostering diverse viewpoints.

(3) Generalization of the developed methods. The third research topic is centered on leveraging the insights gained from our studies on media bias

and applying them to other domains. One area we study is content transfer, which aims to modify the content of a text while preserving its writing style. We frame it as a natural language generation problem that can be solved by the gatekeeping bias mitigation model. Another task is generating biased snippets in web search results, viewing it as a special case of gatekeep bias mitigation. We use the learned knowledge we have acquired to generate these tailored search result snippets.

With the aforementioned research topics, our dissertation contributes to the computational detection and mitigation of media bias. We present empirical experiment findings, accompanied by in-depth discussions of our approaches and results. The models we have developed, alongside our research findings, collectively showcase the feasibility of detecting and reducing media bias through NLP techniques. By identifying biases in the texts, media consumers can have a better awareness of potential biases in the texts. At the same time, texts with mitigated bias offer different aspects of the news, which helps to capture the whole picture of the reported event. To the best of our knowledge, this thesis stands as the first work to comprehensively study media bias analysis and mitigation through the lens of natural language processing.

Abstract (in German)

Computational Analysis and Mitigation of Textual Media Bias

Medien haben einen entscheidenden Einfluss auf die öffentliche Meinung, da sie eine primäre Kommunikationsquelle der modernen Gesellschaft sind. Allerdings hat *Media Bias*, welcher Auftritt, wenn Medien einseitig bzw. voreingenommen berichten oder diskriminieren, das Potenzial, Menschen mit Fehlinformationen oder polarisierenden Ansichten in nicht wünschenswerten Hinsichten zu beeinflussen. Unter Berücksichtigung aktueller Fortschritte im Bereich des maschinellen Lernens ist die natürliche Sprachverarbeitung (NLP) zu einem starken Werkzeug für die Erkennung und Änderung von Media Bias geworden. Die vorliegende Dissertation stellt diesbzüglich eine umfassende Untersuchung von Media Bias dar, welche die Erstellung von Datensätzen, die Entwicklung von computergestützten Modellen und sowohl quantitative, als auch qualitative Analysen umfasst. Insbesondere untersuchen wir drei unterschiedliche Arten von Media Bias: Gatekeeping Bias (die Auswahl des zu Berichtenden), Coverage Bias (die Sichtbarkeit verschiedener Standpunkte), und Statement Bias (die Meinungen von politischen Lagern).

Wir untersuchen primär drei Forschungsthemen, die Media Bias mithilfe von NLP behandeln.

(1) Analyse von Media Bias. Wir erstellen einen Datensatz mit dem Fokus auf Bias in Nachrichtenartikeln. Mithilfe dieses Datensatzes starten wir mit einer Untersuchung von voreingenommener Sprache in Nachrichtenartikeln. Anschließend nutzen wir neuronale Modelle, welche diesen Bias in den Texten erkennen. Die Ergebnisse unserer Studie deuten darauf hin, dass die Erkennung von Bias auf der Artikel-Ebene mit Wissen über Bias auf untergeordneten Ebenen (z.B. auf Satz-Ebene), und umgekehrt, erreicht werden kann.

(2) Minderung von Media Bias. Anhand des gewonnenen Verständnisses über die Merkmale von Media Bias konzentrieren wir uns anschließend auf Strategien zu dessen Minderung. Für die drei bereits erwähnten Arten von Media Bias schlagen wir eigene Strategien zur Minderung vor. Genauer gesagt formulieren wir die Strategien als Aufgaben zur Generierung von

natürlicher Sprache. Um Gatekeeping Bias entgegenzuwirken, lernt unser Modell, neue Informationen in einen existierenden Text fließend zu integrieren. Für Coverage Bias schlagen wir vor, Texte aus Sicht entgegengesetzter Blickpunkte umzuschreiben. Und schließlich, um Statement Bias entgegenzuwirken, nehmen wir uns zum Ziel, den politischen Standpunkt des Textes zu ändern, um so diverse Ansichten zu fördern.

(3) Generalisierung der entwickelten Methoden. Das dritte Forschungsthema beschäftigt sich hauptsächlich damit, die aus unseren Untersuchungen gewonnenen Erkenntnisse über Media Bias zu nutzen und sie auf andere Domänen anzuwenden. Einen Bereich, den wir untersuchen, ist der Inhaltstransfer, welcher zum Ziel hat, den Inhalt eines Textes zu ändern, während der Stil erhalten bleibt. Wir formulieren diese Aufgabe als Generierung von natürlicher Sprache, welche von dem oben zur Minderung von Gatekeeping Bias genutzten Modells gelöst werden kann. Eine weitere Aufgabe ist die Generierung von kurzen, voreingenommenen Textausschnitten, die für Ergebnisse in einer Websuche angezeigt werden, welche wir als einen speziellen Fall der Minderung von Gatekeeping Bias betrachten. Wir nutzen das gewonnene Wissen zur Generierung dieser zugeschnittenen Textausschnitte für die Suchergebnisse.

Mit den erwähnten Forschungsthemen trägt diese Dissertation zur computergestützten Erkennung und Minderung von Media Bias bei. Wir präsentieren Ergebnisse aus empirischen Experimenten, welche mit ausführlichen Diskussionen unserer Ansätze und der Ergebnisse einhergehen. Die von uns entwickelten Modelle, zeigen zusammen mit unseren Forschungsergebnissen, dass die Erkennung und Minderung von Media Bias mithilfe von NLP-Techniken möglich ist. Die Identifizierung von Bias in Texten ermöglicht Medienkonsumenten, ein besseres Bewusstsein über die in den Texten vorhandenen politischen Biases zu erlangen. Gleichzeitig bieten Texte, in denen Bias gemindert wurde, verschiedene Ansichten auf die Nachrichten, was dabei hilft, einen umfänglichen Überblick über das berichtete Ereigniss zu schaffen. Nach unserem Wissen ist diese Dissertation die erste Arbeit, welche umfassend untersucht, wie Media Bias mithilfe von natürlicher Sprachverarbeitung analysiert und gemindert werden kann.

Contents

1 Introduction

> Nihil est in intellectu quod non sit prius in sensu (Nothing is in the intellect that was not first in the senses).
>
> St. Thomas Aquinas

In this chapter, we first motivate to study textual media bias in Section §1.1. In particular, we highlight our view of bias and the media. Afterward, in Section §1.2, we discuss the research questions regarding the textual media bias and how we are going to answer them. Next, we discuss the limitations of our studies in Section §1.3. Section §1.4 outlines the structure of the remaining chapters. Lastly, in Section §1.5 we summarize the publications included in this thesis.

1.1 Textual Media Bias

In contemporary society, media serves as an indispensable source of information in our daily lives, encompassing various information consumption activities such as watching or reading news reports, and browsing webpages on the internet. For instance, Yuan (2011) studied the penetration rates (percentage of people using a particular service) of media in Chinese major cities. The results show that both radio and television have 100% penetration rates, and the internet has penetration rates between 52.91% to 65.63%. The report also shows that 76% of the users consume news every day. Besides, a national survey conducted in Australia (Nguyen and West-

ern, 2006) indicates that a significant majority of internet users, specifically three-quarters, receive online news and information, and a quarter of them are frequent users. Traditional media remains popular as well, with 86% of the participants utilizing television while 66% prefer newspapers to other media.

The primary purpose of media is to facilitate the transfer of information from one location to another.[1] However, it is crucial to acknowledge that the information conveyed by media is not merely a neutral transmission but also impacted by bias. As Park et al. (2012) pointed out, "Bias of the news media is an inherent flaw of the news production process." Therefore, while media remains a vital source of information, it is necessary to be aware of the potential biases that may influence the information being transmitted.

Knowing that the media play a significant role in shaping our view of the world, biased media coverage of events and issues can influence public opinion (Gentzkow and Shapiro, 2010), voting (DellaVigna and Kaplan, 2007), and beliefs (Happer and Philo, 2013). By selectively reporting on specific events and issues, the biased media can shape the public's perception of reality, emphasizing particular narratives while downplaying or ignoring others (Iyengar and Hahn, 2009). Moreover, biased media can also influence our understanding of various social, political, and cultural issues by framing them in a particular way (De Vreese, 2005). By representing different groups of people and social issues using frames, the media can also reinforce or challenge stereotypes, discrimination, and social inequalities (Gorham, 1999).

Given the potential impact of the media on our perception of the world, the issue of bias in media has been a long-standing concern. In light of the recent development in natural language processing (NLP) especially in neural networks, researchers have been using NLP techniques in studying textual media bias (media bias that is presented in textual form) in the last decades (Mullainathan and Shleifer, 2002; Hamborg et al., 2019; Kim and Johnson, 2022). With the help of NLP, computational models can be built to *detect bias in the text*, which media consumers may not be aware of. Secondly, computational models with the ability to *mitigate bias in the text* also help to provide a more neutral view of the world.

Yet, most current works in NLP focus on identifying bias in the text but barely on trying to change/mitigate the bias (Park et al., 2012). Therefore, this thesis positions itself to be a comprehensive study of media bias including bias analysis and bias mitigation using NLP techniques. To achieve the

[1] As defined by the Merriam-Webster dictionary, *medium* refers to "a channel or system of communication, information, or entertainment."

goal of the study, we focus on two NLP tasks in the lens of tackling media bias: (1) text classification and (2) text generation.

For text classification, the task is to classify a given text into different classes, for example, to know whether a news article is biased or not. Also, it can be used to detect whether a certain location of a text is biased, e.g., detecting the biased word in a sentence. In this thesis, we cast the bias detection task as a typical text classification problem in NLP. As a result, we can train machine-learning models to complete the tasks.

After detecting the bias in the text, the next step is trying to mitigate such bias. Specifically, mitigating the bias means rewriting the text in order to remove/change the bias, and such a rewriting task is a special case of text generation. For example in a bias-changing case, given the text containing bias, the target type of bias, and context information, a text generation model generates a new piece of text with a different bias. In the thesis, we train text generation models to mitigate media bias, and models vary depending on different bias mitigation scenarios.

1.1.1 Biases Studied in the Thesis

As mentioned, we aim to comprehensively explore textual media bias with the help of NLP. In this subsection, we discuss the type of biases we study in the thesis. In fact, there have been many definitions of media bias. Among them, we follow D'Alessio and Allen (2000) to focus on studying three kinds of media bias, namely *gatekeeping bias*, *coverage bias*, and *statement bias*. These three distinct types of bias provide a comprehensive framework to understand the different ways in which bias can manifest.

Gatekeeping Bias. Gatekeeping bias, or known as selection bias, focuses on the selection of what to report and what not to report in media. Considering that there exist all stories related to one event, gatekeeping bias occurs when selecting a set of perspectives to produce a news article (D'Alessio and Allen, 2000).

For example, the sentence

Six million undocumented workers in this country cause security issues.

discusses the security impact of the event of "undocumented workers in this country." If an article containing the above sentence only elaborates on the security issues of undocumented workers, this article is heavily gatekeeping biased because only one kind of perspective is selected in the article. To mitigate such gatekeeping bias, one can consider adding other perspectives of undocumented workers. For example, the sentence

Undocumented workers are paying their taxes, too.

discusses the contribution of these workers. If an article manages to discuss these two different perspectives of undocumented workers, the article is considered to have lower gatekeeping bias compared to articles discussing only one of the perspectives.

Coverage Bias. Coverage bias focuses on the visibility of each side of an issue. Taking political news as an example, an article without coverage bias should discuss the points of view from each side (left or right) equally (D'Alessio and Allen, 2000).

Within the idea of coverage bias in political sides, we consider framing as the proxy of political sides, where framing means to emphasize a certain perspective of an issue. Accordingly, there are right-oriented frames (e.g., *economic* and *capacity*) which are preferred by right-oriented people, and left-oriented frames (e.g., *security* and *health*) which left-biased people favor (Mendelsohn et al., 2021).

For instance, the following sentence from the Media Frame Corpus uses the economic framing bias:

Implicit in the debate and the stalemate that left the bill to die when Congress adjourned was a recognition that the cost of immigration reform would be high, although no one knew how high.

The word usage can identify the frame, for example, the *cost* in the sentence above indicates the text containing an economy frame. Also, a sentence can have multiple frames. Accordingly, the above example also uses the legality frame because it discusses the bill issue in Congress.

In general, mitigating the coverage bias suggests having comparable visibility of each political side. Take the above text as an example, given that it contains a right-oriented frame (economy), the strategy to mitigate its coverage bias could be having another text using a left-oriented frame (such as security) together in one article.

Statement Bias. Statement bias is concerned with how the media interject their own opinion. Taking political news as an example, showing positive or negative sentiment toward one party suggests a statement bias (D'Alessio and Allen, 2000).

For example, the sentence

Trump is making a huge mistake on Jerusalem.

has a statement bias it contains the writer's criticism against Donald Trump. Also, this sentence contains political left bias because the writer opposes Donald Trump, who is known as a right-wing politician.

To mitigate the statement bias, we consider rewriting the sentence to have an opposite political bias. For example,

Why Trump is right in recognizing Jerusalem as Israel's capital.

With the above sentences in mind, an article discussing both positive and negative opinions on Trump has less statement bias compared to an article showing only positive or negative opinions.

To sum up the discussion of bias mitigation, we need different rewriting strategies for different types of media bias. For gatekeep bias, the rewriting has to insert new information. For coverage bias, the rewriting aims to change the frame. Finally for statement bias, the rewriting seeks to change the opinion. In the following Chapters, we discuss how to build computational models for each bias mitigation strategy.

1.1.2 Media Studied in the Thesis

Many types of media we are using contain the biases we just discussed. In this thesis, we focus on the most commonly used medium: news, as it is the most important source people rely on to receive information (Yuan, 2011; Nguyen and Western, 2006). Besides, our studies focus on textual media. In other words, this thesis does not include other types of media, such as audio media (e.g., broadcasting) and video media (e.g., television).

Among all media, the news could be the most well-studied one regarding media bias in the past decades (Groseclose and Milyo, 2005). Researchers in journalism have been aware of the existence of news bias for a long time (Groseclose and Milyo, 2005). In news media, the reporter can draw the readers' attention to particular entities or events while ignoring others. In particular, the selection of what to report (e.g., positive or negative facets of the reported event), and strategy of how to report (e.g., by phasing to emphasize a positive or a negative impression of the mentioned entities) both play an important role in introducing bias in the reports.

In this thesis, news articles are the material we use to study. Reporters (journalists) write news articles to present events that readers would be interested in. News also covers a wide range of topics, including government announcements, business, economy, politics, sport, etc. Among them, bias is much visible in news articles about political events, and most of the studies on based on political news articles (D'Alessio and Allen, 2000). However, bias also exists in other topics.

In our studies, we leverage NLP techniques to construct computational models aimed at analyzing and mitigating media bias within news articles. With the power of NLP, we create algorithms that not only detect bias but also offer insights into how bias is presented in the text. This allows us to contribute to digital journalism and computation social science. As previously outlined, our research focuses on the three primary types of media bias. Our computational findings also support the findings from traditional journalism. Such synergy between computational analysis and journalistic insights contributes to a better understanding of media bias in news reporting.

1.2 Research Questions and Contributions

This section discusses three umbrella research questions and our contributions to these topics regarding textual media bias. (1) How to analyze textual media bias? (2) How to mitigate textual media bias? And (3) how to apply textual media bias knowledge to other domains of tasks? For each umbrella research question, we outline the subquestions and our contributions in detail.

1.2.1 RQ1. How to Analyze Textual Media Bias?

The awareness of media bias is the very first step in fighting against media bias. Therefore, the first research question in tackling media bias is analyzing media bias, particularly news media. According to Groseclose and Milyo (2005), bias in news media actually *"has nothing to do with the honesty or accuracy"*, but rather means *"taste or preference"* of the reporter. As such, journalists may (1) report partial facts in favor of one particular side and thus (2) conclude with their own opinion as the product of media bias. To begin with, we first prepare the dataset to study media bias. Afterward, we build computational models to analyze media bias. In the following subquestions, we ask how these biases manifest in news media from different perspectives.

Subquestion 1.1 *How is media bias manifested in news articles at lexical level?*

Our first research question aims to know how reporters write biased news articles. Especially, we are interested in whether biased articles have special word usage compared to unbiased articles. To answer it comprehensively, we first crawled a news bias dataset from `allsides.com` including 7,775 news articles with their political bias labels. Based on this corpus,

we developed a metric to find out words that are discriminative in distinguishing different political biases. From the results, we found a general word usage tendency: sentiment words are equally used in politically left or right-biased articles. However, left and right-biased articles would support or oppose different entities.

Subquestion 1.2 *What are bias features other than word usage?*

Our second research question would like to know more features to distinguish biased articles rather than word usage. In this regard, we study the correlation between article-level and sentence-level bias. In other words, if an article is biased, how to know where the biased sentences or words are? Also, if we know the locations and the number of biased sentences in an article, how to know if the article is biased?

To study it, we used the corpus mentioned above and the BASIL (*B*ias *A*nnotation *S*pans on the *I*nformational *L*evel) corpus (Fan et al., 2019) containing another 300 labeled news articles. We developed two machine-learning models. One detects article-level bias using sentence-level bias features, and the other one detects lower-level (such as paragraph and sentence-level) bias using article-level bias. From the experiment results, we provide empirical evidence that there is a strong correlation between article-level and lower-level bias. Therefore, computational models can detect one level of bias given the other. Also, we discuss the location of biased sentences and words in biased articles. The results help the understanding of how reporters embed bias in news articles.

Contributions. Our contributions to media bias analysis are thus threefold. (1) We create a new news bias corpus with articles and corresponding bias labels. We discuss the most distinguished words in different biases. The corpus benefits not only our own research topics but also other research topics related to media bias. (2) We propose novel bias detection models for article-level bias. Our models utilize semantic features such as words and structural features like the locations of the biased sentences. Lastly, (3) our analyses enlighten the following works in bias mitigation. For example, the most distinguished keywords shed some light on the strategies that should be considered in mitigating the biases. Also, the models in detecting bias can be used to automatically evaluate whether the bias mitigation is successful or not.

1.2.2 How to Mitigate Textual Media Bias?

After knowing how media bias is used in the texts, our second step is to mitigate it. Conceptually, mitigating the bias in the text can be done in two ways: (1) rewriting the text and removing its bias, and (2) rewriting the text and adding an opposite bias (if there is an opposite bias). From the natural language processing perspective, this means a text-to-text generation task, also known as a sequence-to-sequence task. Additionally, we would like to keep the original semantics as much as possible after rewriting. In our study, we experiment with different types of biases in news media, to get a full picture of mitigating media bias.

Subquestion 2.1 *How to mitigate gatekeeping bias in news articles?*

In the first subquestion of bias mitigation, we study the first type of bias: gatekeeping bias. As discussed, the gatekeeping bias can be mitigated by introducing other perspectives into the texts. In terms of a natural language generation task, it suggests inserting a new piece of text into existing ones. To do so, the generated text has to both (1) contain the new perspective and (2) fit the surrounding sentences and read naturally. Therefore, we developed a multi-task learning model to simultaneously accomplish the two objectives. Specifically, in the model one task is the generation task and the other tasks are to make sure the generated text fulfills the requirements (i.e., containing the story, fitting the surrounding sentences, and reading naturally).

Our analysis of the generated texts shows the limitation of current approaches, where they tend to add a general description in the scenario of adding a new story. For the same input, our approach can successfully generate a sentence containing correct and detailed information thanks to the multi-task learning model.

Subquestion 2.2 *How to mitigate coverage bias in news articles?*

As discussed above, one way to mitigate the coverage bias is to mention the favored frames from different sides of one issue. The frames come from the media frame corpus (Card et al., 2015). Similarly, from the natural language generation's perspective, our task is to rewrite a text into different frames. To do so, we apply three training strategies in our approach, namely (a) framed-language pretraining, to learn the word usages in framed texts, (a) named-entity preservation, to support the model in maintaining important entities, and (c) adversarial learning, to provide negative samples in order to avoid undesired outputs.

Our analyses include the ablation study of interpreting the importance of the three strategies, and also an in-depth analysis in different reframing directions (changing from one frame to another frame). For instance, we find that changing from crime to economic frames is more difficult than other directions because of the low relationship between the two frames.

Subquestion 2.3 *How to mitigate statement bias in news articles?*

As discussed, we would like to have sentences from opposite sides to mitigate the statement bias. In subquestion 1.1 we acquire the knowledge of word usage on different political sides, and we would like to use the knowledge to change the text. Based on the same corpora we have in subquestion 1.1, we build a text generation model. To rewrite the given text, the text generation model is a conditional generator (Hu et al., 2017) that aims to generate a text given a set of conditions. In our task, the two conditions are keeping the content, and changing from political left to right and vice versa. Our study involves automatic and manual evaluations to judge whether the change is successful or not. The results show that the automatic evaluation is limited in evaluating the changing task because the evaluation requires an in-depth understanding of politics in terms of left-wing or right-wing ideologies. However, the manual evaluation shows that our model successfully changes the text in most cases while there is still room for improvement.

Contributions. In terms of media bias mitigation, we made three important contributions. First of all, (1) we are the first to work on mitigating the three media biases in news articles. (2) For all three biases, we introduce the bias mitigation task. (3) For these three bias mitigation tasks, we propose approaches and provide empirical evidence of where our approaches can best change one bias to the other.

1.2.3 How to Apply Textual Media Bias Knowledge to Other Domains of Tasks?

After analyzing and mitigating the media biases we discussed, we examine the generalization of our approaches by studying other domains using the knowledge we learned. In the following, we study two NLP tasks: content transfer, which changes the content of a text but keeps the writing style unchanged, and biased snippet generation, which generates biased snippets for web search engines.

Subquestion 3.1 *How to apply media bias knowledge on the content transfer task?*

The content transfer task is a new NLP research direction. The goal is to rewrite the text in order to insert new content while keeping the writing style unchanged. In our study, we first cast it as a special case of our gatekeeping bias mitigation problem, where the new content to be added is the topic bias. After that, we train a topic bias mitigation model following the architecture in subquestion 2.1.

Our main findings show that our approach achieves a higher measurement in terms of transferring the content. It suggests that the content transfer task can be approached by our media bias mitigation models.

Subquestion 3.2 *How to generate biased snippets for search engines?*

Another NLP task we are interested in is biased snippet generation. This study focuses on generating a snippet of a search result, a short text summarizing a web page. Specifically, we are interested in generating the snippets given different biases (presented as queries in web search). The first task of this study is to perform a user study to verify if the generated biased snippets can be used to find the desired webpages by users. Secondly, we prepare a dataset with snippets, web pages, and bias tuples, where we propose to extra the anchor text (text containing hyperlinks in web pages) and its surrounding sentences (details in Chapter 5). Afterward, we developed a new text generator, a bi-directional generator. The bi-directional generator first generates the query words and then generates the rest of the texts from two directions (from the query word to the beginning of the sentence and from the query word to the end of the sentence).

The study includes the details of our method for acquiring 10 million such training tuples. Also, we show that the proposed approach can mostly guarantee the existence of the desired query words compared to other approaches.

Contributions. We make significant contributions to further research topics about media bias. These two topics are related to how we can use our knowledge of analyzing and mitigating media bias in other domains. Our contributions are three-fold. (1) We cast the content transfer task as a topic bias mitigation task, and we apply our model to it. (2) We create a dataset for studying topic bias in snippets. (3) Lastly, we propose an approach for generating biased snippets based on the idea of topic bias mitigation.

1.3 Overall Contribution and Limitations

In this thesis, we try to comprehensively study textual media bias, including analyzing, mitigating, and expanding to other domains. Nevertheless, our studies still have some limitations.

First, we are aware of the limited selection of media, and in fact, we only focus on textual news media. On one hand, such a limitation comes from the lack of suitable annotated datasets. On the other hand, we are missing the required knowledge to work on them. For example, knowing speech processing is required to tackle bias in broadcasting, but speech processing and other needed domain knowledge are out of the scope of this thesis. However, we expect future works in studying media bias can try to study the missing media and provide results beyond our findings.

Secondly, for all research questions, we have to narrow them down to solving specific tasks. There could be other tasks that can be introduced using the same set of media bias corpora and research questions. Still, our study provides representative examples of NLP tasks for answering the questions. We believe casting other NLP tasks is yet possible, but we have provided a useful and thoughtful case for answering the questions. In particular, we do not evaluate whether the three biases can be mitigated or reduced by our approaches. Yet, we expect future researchers to apply our methods in practice.

Lastly, there may be other types of biases that cannot be found within our definition. Or, there will be new media emerging in the future beyond our view of media in this thesis. However, the goal of this thesis is to try to have a comprehensive study within our definitions as much as possible. We still expect that biases or media in the new era can still be studied based on this work.

Nevertheless, we contribute to the research of media bias in many regards. (1) First of all, our media bias analysis reveals the linguistic phenomenon of media bias. The findings contribute to the future study of media bias analysis from NLP and journalism perspectives. We provide insights that media bias can be effectively detected, and it should be detected to increase the awareness of media bias for media consumers. (2) Secondly, our media bias mitigation demonstrates the possibility of mitigating different kinds of media bias with the help of NLP. In the era of AI-powered text generation, we shed some light on how to use the technique ethically to remove bias in the texts.

1.4 Structure of this Thesis

The remaining chapters of this dissertation are organized as follows: Chapter 2 reviews the background knowledge and necessary previous work for the subjects related to the later chapters. For Chapters 3 through 5, we introduce the research topic and detail the approach for each topic. In the end, a summary is given to summarize the findings of each chapter. In detail, Chapter 3 analyzes bias in media in order to answer Research Question 1. Equipped with knowledge from the previous chapter, in Chapter 4 we try to mitigate three types of media biases corresponding to Research Question 2. Chapter 5 contributes to further research topics for Research Question 3, where we use the knowledge from the above two chapters to study other domains of NLP tasks. Finally, Chapter 6 summarizes the dissertation, reviews the main findings, and discusses research topics for future works in studying textual media bias.

1.5 Publications Included in this Thesis

This thesis includes publications from major natural language and information retrieval conferences, workshops, and student theses. Table 1.1 summarizes these publications and how they are used in this thesis. In particular, we contribute the user study to the paper of Potthast et al. (2018), which partially motivates our works in Chapter 5. In the remaining, Chapter 3 uses the content from Chen et al. (2018b, 2020a,b). Chapter 4 involves the content from Chen et al. (2018b, 2020b, -). Lastly in Chapter 5, it use the content from Chen et al. (-, 2018a, 2020c)

Table 1.1: A selection of publications by the author and their usage within this dissertation.

Year	Venue	Type	Pages	Used in
2018	INLG	conference	10	Chapter 3, 4
	***Wei-Fan Chen**, Henning Wachsmuth, Khalid Al-Khatib and Benno Stein.*			
	Learning to Flip the Bias of News Headlines.			
2018	NewsIR	workshop	3	Chapter 5
	*Martin Potthast, **Wei-Fan Chen**, Matthias Hagen and Benno Stein.*			
	A Plan for Ancillary Copyright: Original Snippets.			
2018	SIGIR	conference	4	Chapter 5
	***Wei-Fan Chen**, Matthias Hagen, Benno Stein and Martin Potthast.*			
	A User Study on Snippet Generation: Text Reuse vs. Paraphrases.			
2019	NLP4IF	workshop	7	-
	***Wei-Fan Chen**, Khalid Al-Khatib, Matthias Hagen, Henning Wachsmuth and Benno Stein.*			
	Unraveling the Search Space of Abusive Language in Wikipedia with Dynamic Lexicon Acquisition.			
2020	NLP&CSS	workshop	6	Chapter 3
	***Wei-Fan Chen**, Khalid Al-Khatib and Benno Stein, Henning Wachsmuth.*			
	Analyzing Political Bias and Unfairness in News Articles at Different Levels of Granularity.			
2020	EMNLP Findings	conference	11	Chapter 3
	***Wei-Fan Chen**, Khalid Al-Khatib and Benno Stein, Henning Wachsmuth.*			
	Detecting Media Bias in News Articles using Gaussian Bias Distributions.			
2020	WebConf	conference	10	Chapter 5
	***Wei-Fan Chen**, Shahbaz Syed, Benno Stein, Matthias Hagen and Martin Potthast.*			
	Abstractive Snippet Generation.			
2021	EMNLP Findings	conference	10	Chapter 4
	***Wei-Fan Chen**, Henning Wachsmuth, Khalid Al-Khatib and Benno Stein.*			
	Controlled Neural Sentence-Level Reframing of News Articles.			
2022	ArgMining	workshop	11	-
	***Wei-Fan Chen**, Mei-hua Chen, Garima Mudgal and Henning Wachsmuth.*			
	Analyzing Culture-Specific Argument Structures in Learner Essays.			
-	under review	conference	11	Chapter 4, 5
	***Wei-Fan Chen**, Milad Alshomary, Maja Stahl, Khalid Al-Khatib, Benno Stein and Henning Wachsmuth.*			
	Towards Factual Correctness in Conditional Text Generation via Knowledge Distillation.			

2

Background and Related Work

> No one is so brave that he is not disturbed by something unexpected.
>
> Julius Caesar

This Chapter introduces the background and related work of this thesis. Section §2.1 discusses the foundations of textual media bias from the perspective of journalism. Later in Section §2.2, we discuss textual media bias from a natural language processing perspective. In Section §2.3, we review the related work about textual media bias analysis and textual media bias mitigation. We summarize the tasks and the approaches in the introduced work. Lastly in Section §2.4, we conclude this Chapter.

2.1 Textual Media Bias and Journalism

Journalism is a discipline that includes gathering, composition, and reporting of news. In a broad sense, it also includes the process of editing and presenting news. Journalism plays an important role in various media such as newspapers, television, and radio. A famous description by former Washington Post president and publisher Philip L. Graham said journalism is "the first rough draft of history." In the following, we briefly summarize the key topics in journalism, particularly focusing on textual media bias.

2.1.1 Presence of Media Bias

While journalistic objectivity is one of the core journalistic ethics (Ryan, 2001), bias in journalism still exists. Patterson and Donsbagh (1996) performed a survey on journalists from five different Western democracies and concluded that their reports are biased. Especially, Patterson and Donsbagh (1996) found out, "When they move from facts to analysis, their decisions are subject to errors of judgment and selectivity of perception. " An early study by Johnstone et al. (1972) also concluded that even "the most highly trained and perhaps best educated journalistic practitioners thus tend to embrace participant ideologies of the press."

The news portals themselves are also aware that they are biased. Many news portals have ombudsmen or public editors to supervise the implementation of proper journalism ethics at that publication. As an example, the New York Times had the position of public editor devoted "to receiving, investigating, and answering outsiders' concerns about our coverage."[1]

For news consumers, they are aware of bias in the reports. However, they do not view that bias as a major hindrance to using the news (Baron, 2006). Even, news consumers prefer biased news than (Mullainathan and Shleifer, 2005). The so-called *confirmation bias* suggests that readers "hold beliefs which they like to see confirmed, and that newspapers can slant stories toward these beliefs" (Mullainathan and Shleifer, 2005).

2.1.2 Impact of Media Bias

Media bias can have a strong impact on society and influence public opinion. One of the sources of media bias comes from the management of governement (Weis, 1997). By adding bias in the reports, the biased reports can be used as a powerful tool for the government to control or guide public opinion Watanabe (2017). During the Ukraine crisis, the Russian government used the state-owned news agency for international propaganda in its hybrid war Watanabe (2017).

Media bias can be used to change readers' opinions. Silverman et al. (2011) studied the impact of reading news articles related to the Middle East conflict published on Reuters. The author found that readers significantly shifted their sentiments when they read articles in favor of the Arabs/Palestinians.

[1] `https://www.nytimes.com/column/the-public-editor`

2.2 Textual Media Bias and NLP

Natural Language Processing (NLP) is a field of computer science that involves developing algorithms and computational models to understand (known as natural language understanding (Allen, 1995)), analyze (for example sentiment analysis (Medhat et al., 2014), and generate (known as natural language generation (Gatt and Krahmer, 2018)) human language. This thesis focuses on studying textual media bias using NLP techniques, in particular, analyzing the text to detect the bias and modifying the text to mitigate the bias. In the following, we briefly introduce the important NLP topics strongly related to our thesis.

2.2.1 Language Models

Language Models are trained on large amounts of text data to learn the patterns and structure of language. Traditionally, language models use probabilities to model human language (Bellegarda, 2004). One common type of traditional language model is the n-gram model, which is based on the frequency of n consecutive words or tokens in a text. N-gram models estimate the probability of a word given its context. For example, the following equation approximates a Bigram model (N=2).

$$p(w_n) = p(w_1) \cdot p(w_2|w_1) \cdot ... \cdot p(w_n|w_{n-1}) \tag{2.1}$$

where $p(w_x)$ refers to the probability of having the word w_x given the previous words w_1 to w_{n-1}. In bigram, the basic assumption is that any words are only conditioned in the previous word.

While traditional language models have been widely used in NLP for many years and have been effective in certain applications, they often have limitations in handling complex language patterns, capturing long-range dependencies, and adapting to new data. One obvious drawback of traditional language models is that the size of a language model is increased exponentially. Also, we need a tremendous amount of data to train reliable probabilities for all combinations of words.

2.2.2 Neural NLP Models

A recent popular direction of NLP is using neural networks. Powered by the increasing computational powers of graphic cards, neural NLP models significantly improve the performance of almost all NLP tasks. One of the key advantages of neural NLP models is their ability to learn from data without the need for handcrafted features or explicit rules, making them highly

flexible and adaptable to different domains and languages. They can also handle complex language patterns, capture long-range dependencies, and generate coherent text.

For neural NLP models, neural language models are one of the key components, such as BERT (Devlin et al., 2019) and GPT (Brown et al., 2020). These models are capable of processing sequences of words or tokens, allowing them to capture the contextual relationships and dependencies among words in a sentence or a document. Such Large Language Models (LLMs) are a recent breakthrough in NLP that have led to significant advances in various applications such as language translation, language generation, and question answering (like the recent popular ChatGPT.)[2]

In this thesis, most of our works are based on neural NLP models. We either develop our neural architecture built on existing models or reuse the existing neural frameworks.

2.2.3 Natural Language Generation

Natural Language Generation (NLG) is a subfield of NLP that focuses on generating human-like texts (Gatt and Krahmer, 2018). NLG systems use computational algorithms to automatically produce text that is coherent, fluent, and contextually appropriate.

NLG models can be rule-based, template-based, or data-driven. Rule-based and template-based NLG systems use predefined rules or templates to generate text based on specific patterns or structures. Data-driven NLG models, on the other hand, leverage machine learning techniques, such as neural networks, to generate text based on patterns learned from large amounts of data.

NLG has a wide range of applications, including generating reports (Huang et al., 2020), summaries (Allahyari et al., 2017), chatbot responses (like ChatGPT), and more. NLG has seen significant advancements in recent years, particularly with the use of deep learning techniques. These models have improved the quality and fluency of generated text, making NLG systems more sophisticated and capable of producing high-quality outputs.

In this thesis, textual media bias mitigation is precisely one of the applications of NLG. In particular, we use NLG to modify a piece of text while mitigating the bias in the text.

[2] https://chat.openai.com/

2.3 Related Work

In this section, two main research lines are discussed: textual media bias analysis and textual media bias mitigation. We select those works mostly close to the thesis.

2.3.1 Textual Media Bias Analysis

Textual media bias analysis has been studied for decades under different names, including *perspective* (Lin et al., 2006; Greene and Resnik, 2009), *ideology* (Iyyer et al., 2014), *truthfulness* (Rashkin et al., 2017), and *hyperpartisanship* (Kiesel et al., 2019). These terms define media bias differently, while some of them overlap with each other. One of the earliest works was (Lin et al., 2006), where they classified documents into Palestinian and Israeli perspectives. A similar term is an ideology (or political ideology) (Iyyer et al., 2014), where the documents are classified into conservative and liberal. Truthfulness (Rashkin et al., 2017), on the other hand, classifies statements into six trustfulness labels (true, mostly true, half true, mostly false, false, and pants-on-fire).

To detect the mentioned bias in the above forms, early approaches relied on lexical information. For example, Greene and Resnik (2009) used *kill* verbs and *domain-relevant* verbs to detect articles being pro-Israeli or Palestinian perspectives. Recasens et al. (2013) relied on linguistic cues, such as factoid verbs and implicatives, in order to assess whether a Wikipedia sentence conveys a neutral point of view or not. Besides the NLP community, also researchers in journalism have approached the measurement of media bias. E.g., Gentzkow and Shapiro (2010) used the preferences of phrases at each side (such as "war on terror" for Republicans but "war in Iraq" for Democratic). Groseclose and Milyo (2005) used the counts of think-tank citations to estimate the bias. In particular, they explored the bias in a sample of 20 news sources in the US. The bias was quantified based on the number of citations that were used by the think tanks and policy groups. Their work is one of the first that provided clear evidence of the presence of bias in media. Furthermore, Lin et al. (2011) proposed a scheme for bias categorization. The scheme includes the political party, frequently mentioned legislators, region, ideology, and gender. In a comparison study between the bias in news and blogs, the authors found blogs to be more sensitive to bursting events. In another related work, Yano et al. (2010) focused on liberal and conservative bias. Most notably, they conducted a manual annotation of the bias at the sentence level. Their study showed that bias indicators usually include named entities of opposing bias. As for our work, we

deal with right and left biases, e.g., the Democrats' and Republicans' bias, or conservative and liberal bias. Also, we analyzed to find the terms that frequently indicate left or right bias.

With the rise of deep learning, NLP researchers have also used neural-based approaches for bias detection. Iyyer et al. (2014) used RNNs to aggregate the polarity of each word to predict sentence-level bias based on parse trees. Gangula et al. (2019) made use of headline attention to classify article bias. Li and Goldwasser (2019) encoded social information in their Graph-CNN.

2.3.2 Textual Media Bias Mitigation

Over the few last years, several deep neural network models have been proposed for text generation. In these models, a variational autoencoder (VAE) has often been used to impose a prior distribution on the hidden vector (Kingma and Welling, 2013; Rezende et al., 2014; Bowman et al., 2016; Yang et al., 2017).

A related research line that addresses rewriting texts is *controlled generation* (Guu et al., 2017; Mueller et al., 2017; Zhou and Neubig, 2017). Controlled generation studies how to rewrite a text with a given attribute. Examples of controlled models include the multi-space VAE of Zhou and Neubig (2017), which modifies a word for a given tense and a part-of-speech tag, and the model of Guu et al. (2017), which generates a sentence given a template vector and an edit vector. This model is shown to be able to paraphrase a given template instead of re-generating a sentence entirely.

Specifically in mitigating one of the biases we are interested in, the only existing reframing approach that we are aware of is the one of Chakrabarty et al. (2021). In that work, a new model for reframing is developed by identifying phrases indicative for specific frames, and then replacing phrases that belong to the source frame with some that belong to the target one. As such, most of the content of the reframed text is kept, and only a few words are replaced. In contrast, we deal with reframing at the sentence level, and we do not require parallel training pairs or a dictionary to correlate words and frames.

In principle, textual media bias mitigation can be seen as a style transfer task (Shardlow, 2014; Shen et al., 2017; Chen et al., 2018b). Research on text style transfer focuses on the areas of sentiment transfer (e.g., replacing 'gross' by 'awesome') (Shen et al., 2017) and text simplification (e.g., replacing 'perched' by 'sat') (Shardlow, 2014).

2.4 Summary

This chapter has introduced the background and previous works related to this thesis. We first discuss media bias in journalism, and later shift our focus to natural language processing techniques including neural models and natural language generation for tackling media bias.

In the second half of this chapter, we discussed the related work from two perspectives: textual media bias analysis and textual media bias mitigation. In particular, we briefly introduced how media bias was defined in different works and how the researchers dealt with media bias.

In this Chapter, we have learned the presence of media bias and its potential impact on society. Subsequently, we discuss the needed knowledge of NLP within the context of this thesis. Building upon this foundation, we move on to discussing NLP techniques for dealing with media bias. In the later chapters, we elaborate on our methodologies for investigating textual media bias.

3

Textual Media Bias Analysis

> He (the high-minded man) must care for truth more than for what men will think of him, and speak and act openly.
>
> Aristotle

News media bear great responsibility because of their considerable influence on shaping the beliefs and positions of our society. Biased media can influence media users in undesirable directions and hence should be unmasked as such. In this chapter, we are interested in whether and how media bias is manifested in the news articles we read every day. Especially, we focus on statement bias in the text. Accordingly, we lay out the results obtained in our previously published papers in analyzing these biases (Chen et al., 2018b, 2020a,b).

Section §3.1 discusses how we created the bias corpus and how we analyzed the political news articles accordingly to the published paper (Chen et al., 2018b). To study how bias is manifested in the texts, we first create the bias corpus, utilizing the political bias labels found on `allsides.com`. This platform collects news reporting on the same event while conveying different political biases (left-oriented, neutral, or right-oriented). Using the corpus, we analyze the statement bias by extracting the most discriminative words. Besides, we study statement bias at different levels of granularity, including sentence level, paragraph level, and article level in Section §3.2 based on the publish paper (Chen et al., 2020a). In addition to political bias, we introduce unfairness as a second dimension of statement bias. We

developed a new approach to detect biased sentences and paragraphs based on only article-level biases. In Section §3.3, we study bias detection from another perspective: identifying article-level bias from sentence-level bias features based on the published paper (Chen et al., 2020b) . We discuss three features using sentence-level bias. In particular, we use the probability distributions of the frequency, positions, and sequential order of sentence-level bias. Finally, Section 3.4 summarizes the contributions we made in textual media bias analysis.

3.1 Media Bias Corpus

This section introduces our statement bias dataset of news articles with different political biases, based on existing bias labels from a news aggregator. The corpus is freely available at `https://webis.de/data/corpus-webis-bias-flipper-18`. After creating the corpus, we perform a discriminativeness analysis in order to find keywords discriminating different political biases.

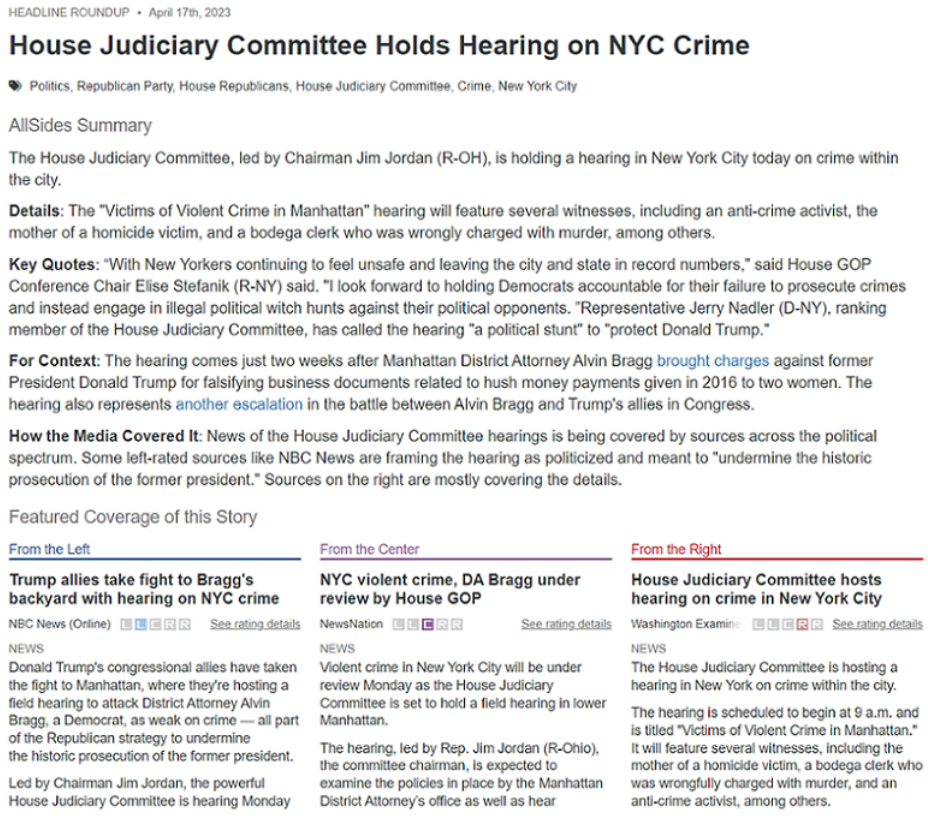
3.1.1 The News Aggregator allsides.com

The news aggregation platform `allsides.com` lists news events as of June 1st, 2012; about two to three events per day, focusing on political events in the US. Each event comes with a title and a summary, providing information to readers to understand the event from different perspectives at the same time. In addition, one selected news article is given for each of three biases: *left*, *center*, *right* (occasionally, only two articles are available).

In addition, the provided bias labels are not article-specific but portal-specific.[1] At the time we collected the data, `allsides.com` assigned 247 news portals to one out of six labels each: *left*, *lean left*, *center*, *lean right*, *right*, and *mixed*. In this study, we see both the left and the lean left portals as left-oriented news sources, and both the right and lean right portals as right-oriented news sources. The center and mixed portals are preserved for future applications.

As the labels are portal-specific, news articles with a particular bias are selected from all portals that have the respective labels. Conversely, no portal contains articles with different biases.

[1] `https://www.allsides.com/media-bias/media-bias-ratings`


HEADLINE ROUNDUP • April 17th, 2023

House Judiciary Committee Holds Hearing on NYC Crime

Politics, Republican Party, House Republicans, House Judiciary Committee, Crime, New York City

AllSides Summary

The House Judiciary Committee, led by Chairman Jim Jordan (R-OH), is holding a hearing in New York City today on crime within the city.

Details: The "Victims of Violent Crime in Manhattan" hearing will feature several witnesses, including an anti-crime activist, the mother of a homicide victim, and a bodega clerk who was wrongly charged with murder, among others.

Key Quotes: "With New Yorkers continuing to feel unsafe and leaving the city and state in record numbers," said House GOP Conference Chair Elise Stefanik (R-NY) said. "I look forward to holding Democrats accountable for their failure to prosecute crimes and instead engage in illegal political witch hunts against their political opponents. "Representative Jerry Nadler (D-NY), ranking member of the House Judiciary Committee, has called the hearing "a political stunt" to "protect Donald Trump."

For Context: The hearing comes just two weeks after Manhattan District Attorney Alvin Bragg brought charges against former President Donald Trump for falsifying business documents related to hush money payments given in 2016 to two women. The hearing also represents another escalation in the battle between Alvin Bragg and Trump's allies in Congress.

How the Media Covered It: News of the House Judiciary Committee hearings is being covered by sources across the political spectrum. Some left-rated sources like NBC News are framing the hearing as politicized and meant to "undermine the historic prosecution of the former president." Sources on the right are mostly covering the details.

Featured Coverage of this Story

From the Left

Trump allies take fight to Bragg's backyard with hearing on NYC crime

NBC News (Online) See rating details

NEWS

Donald Trump's congressional allies have taken the fight to Manhattan, where they're hosting a field hearing to attack District Attorney Alvin Bragg, a Democrat, as weak on crime — all part of the Republican strategy to undermine the historic prosecution of the former president.

Led by Chairman Jim Jordan, the powerful House Judiciary Committee is hearing Monday

From the Center

NYC violent crime, DA Bragg under review by House GOP

NewsNation See rating details

NEWS

Violent crime in New York City will be under review Monday as the House Judiciary Committee is set to hold a field hearing in lower Manhattan.

The hearing, led by Rep. Jim Jordan (R-Ohio), the committee chairman, is expected to examine the policies in place by the Manhattan District Attorney's office as well as hear

From the Right

House Judiciary Committee hosts hearing on crime in New York City

Washington Examiner See rating details

NEWS

The House Judiciary Committee is hosting a hearing in New York on crime within the city.

The hearing is scheduled to begin at 9 a.m. and is titled "Victims of Violent Crime in Manhattan." It will feature several witnesses, including the mother of a homicide victim, a bodega clerk who was wrongfully charged with murder, and an anti-crime activist, among others.

Figure 3.1: An example from allsides.com. At the top, the website shows the title of the event and summarizes the event by discussing how the media covered it. At the bottom, three news outlets (NBC News, NewsNation, and Washington Examiner were shown to present the three different perspectives.

3.1.2 Corpus Construction

We first collected all 2781 events available on the aggregator on February 10th, 2018 (spanning about five and a half years).[2] For each event, the title, the summary, all news portals belonging to the event, and the links to the news portals with respective biases were collected. After that, we crawled the news portals with the given links to retrieve their headlines and the content of all articles, because the content is not provided on the webpages in allsides.com. Metadata such as an article's author and its publication time were also collected for future applications. We retrieved 6,447 news articles in the end since some news articles were not available anymore.

The distribution of news portals and articles in our corpus is shown in Table 3.1. To validate the accuracy of the by-portal bias, we hired one edit-

[2] `https://www.allsides.com/story-list`

	News Portals		News Articles	
Bias	**Most Common**	**Total**	**Most Common**	**Total**
Left	Huffington Post	21	479	641
Lean left	New York Times	18	688	1747
Center	CNN (web)	24	776	1517
Lean right	Fox News	6	1061	1616
Right	Townhall	28	279	926

Table 3.1: News portals and articles in our corpus for each bias in total and in the most common portal.

ing expert on upwork [3] to label the bias of all headlines from major left-oriented (New York Times and Huffington Post) and right-oriented portals (Fox News and Townhall). The expert is familiar with American politics and he works as a news editor in the US. His labels are based on the headline only, and the judgments follow the notion of political bias from an American's point of view.

The expert assigned *left* to the headlines of left-oriented portals 3.4 times more than *right*, while the headlines from right-oriented portals have 1.9 times *right* more than *left*. Given the results, we conclude that the by-portal labels from the aggregator are trustable in general.

In detail, the portal labels on allsides.com are created based on different methods including blind surveys, academic research, feedback from the community, and in-depth editorial reviews from allsides.com editors[4]. The final portal labels consider the strength and consistency of the labels from the different methods. The most common portal contributes at least 30 percent of articles of each bias. The total number of right-oriented news slightly exceeds the number of left-oriented (2542 vs. 2388).

According to the community feedback on the website, the provided labels are agreed upon by the website's users in general. Thus, we argue that the labeling can be seen as being of high quality to be used as our ground-truth labels.

3.1.3 Discriminativeness Analysis

To gain knowledge of the difference between left and right biases, one way is to capture the words being used differently in different biases. In this regard, we define the discriminativeness of a word w can be measured in

[3] upwork.com

[4] https://www.allsides.com/media-bias/media-bias-rating-methods

Word	Ratio
Chad	9.52
Maduro	5.56
purportedly	7.81
Chechnya	6.80
Bethlehem	6.04
...	...
victorious	1.01
oppressive	1.01
tragedy	0.99
...	...
Shawn	0.04
incarceration	0.04
album	0.03
valuable	0.03
N.S.A	0.02

Table 3.2: The five words each with the highest and lowest discriminativeness ratio, and words with a ratio close to one in biased text.

terms of the *discriminativeness ratio*

$$\frac{occ(w, D_t)}{occ(w, D_{\bar{t}})}, \tag{3.1}$$

where $occ(w, D)$ is the frequency of w in text D and t and $\bar{t}$ are the types of text. In our case, t and $\bar{t}$ correspond to *right* and *left*. We normalize the occurrence by the total number of words of the respective type of text.

The discriminativeness ratio will make function words and type-unrelated words have values close to one, because these words are expected to occur similarly often in both types. On the other hand, words that often appear in one type but rarely in the other will have a high value (in case of type t) or a low value (type $\bar{t}$). To demonstrate the differences in discriminativeness ratios, we compute the ratio for all the words from the corpus we created where the articles are right or left-biased.

In Table 3.2, we list the words having the highest and the lowest discriminativeness ratio in the biased texts. The first observation is that both positive and negative sentiment words have a frequency ratio close to one. This is expected, because we observe that both sides use positive (negative) words to support (oppose) some entities. Moreover, many of the top-5 and the bottom-5 words are named entities, such as *Maduro* and *N.S.A* (*National Security Agency*). This indicates that articles with either bias tend to criticize or approve different entities, but that they do not use different sentiment

words to do so. In line with this, a previous analysis of biased language showed that many bias indicators include named entities (Yano et al., 2010).

The created corpus provides the material for our statement bias analysis and mitigation studies. The analysis of the corpus shows that some keywords are important to distinguish different biases. In the following, we are going deep in this direction by studying statement bias at different levels other than just word level.

3.2 Media Bias Analysis at Finer Granularities

In this section, we study statement bias at different granularity levels, from single words, to sentences and paragraphs, to the entire discourse to get a full picture of how statement bias manifests in an article. Also, we would like to extend the corpus we had by including a fairness label from `adfontesmedia.com`, where a fair article means the article focuses on original facts rather than on analyses or opinion statements based on false premises. In the end, the new corpus consists of 6,964 news articles, each of which is labeled for its topic, political bias, and unfairness. Based on this corpus, we develop a recurrent neural network (see Section §2.2) architecture to learn classification knowledge for bias detection. We choose this network class because of its proven ability to capture semantic information at multiple levels: taking the model output for whole texts, we conduct an in-depth reverse feature analysis to explore media bias at the word, the sentence, the paragraph, and the discourse level. At the word level, we correlate the most biased sentences with LIWC categories. At the sentence and paragraph level, we reveal what parts of an article are typically most politically biased and unfair. At the discourse level, we reveal common sequential media bias patterns.

3.2.1 Corpus Extension

We started with the creation of a corpus for this study. We extend the corpus collected from allsides.com in the previous section and add the fairness labels provided by `adfontesmedia.com`. The new corpus is available at `https://github.com/webis-de/NLPCSS-20`.

For this study, we extended the corpus in the previous section by integrating all articles until March 15, 2019, resulting in a total of 7,775 articles. In addition to the political bias labels, we crawled the topic tags of each article.

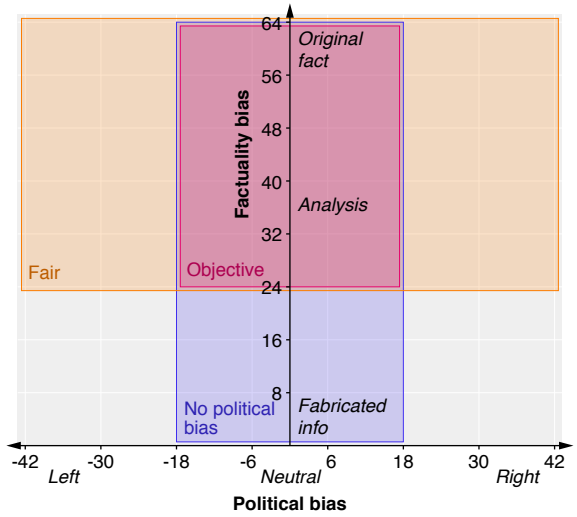


Figure 3.2: The bias chart from `adfontesmedia.com`(visually adjusted) in our paper (Chen et al., 2020a). The three rectangles represent the positive counterparts of the regions of the three bias types in our definition (*political bias*, *unfairness*, and *non-objectivity*).

Since `allsides.com` focuses on political bias, we exploit `adfontesmedia.com` as another source for additional bias types. This portal maintains a "bias scale" quantifying the media bias of a broad set of US news portals. The bias assessments stem from media experts who annotate each portal with bias and fairness labels. As supporting evidence of the label quality, Bentley et al. (2019) show that the portals' labels are highly correlated to findings from social scientists.

Figure 3.2 gives an overview of the labels from `adfontesmedia.com`; it is based on the bias chart at the website: The political bias focuses on the x-axis of the chart, while the unfairness focuses on the y-axis of the chart. The non-objectivity is the combination of the two kinds of bias.

Based on the labels from `adfontesmedia.com`, we define three media bias types for news portals:

1. **Political Bias.** A portal is *neutral* if it is labeled as "skew left/right" or "neutral". It is *politically biased* if it is labeled with "most extreme left/right" or "hyperpartisan left/right".

2. **Unfairness.** A portal is considered *fair* if it is labeled as "original fact reporting", "fact reporting", "mix of fact reporting and analysis", "analysis", or "opinion". The portal is considered *unfair* if it is labeled as "selective story", "propaganda", or "fabricated info".

Portal		Topic	
Name	Count	Name	Count
CNN	1021	presidential election	914
Fox News	1002	politics	525
New York Times	781	White House	515
...		...	
NPR News	1	domestic policy	2
The Nation	1	EPA	1
Vice	1	women's issues	1

Table 3.3: The top three and the bottom three portals along with the topics in our corpus.

3. **Non-Objectivity.** A portal is considered *objective* if it is politically unbiased and fair. Otherwise, it is considered as *non-objective*.

We label the collected articles according to this scheme. Since adfontesmedia.com does not cover all portals from allsides.com, the final corpus contains 41 portals with 6,964 articles. The three largest portals are CNN (1021 articles), Fox News (1002 articles), and the New York Times (781 articles). Altogether, we count 111 different topics such as, "presidential election" (914 articles), "politics" (525 articles), and "white house" (515 articles). Table 3.3 lists the top three and the bottom three portals along with topics in our corpus.

3.2.2 Statement Bias Classification

For the detection of bias in a text, we develop classifiers with RNN that serve as the classical model for sequential inputs, where a cell is a GRU with a recurrent state size of 32. On top of the final hidden vector of GRUs is a prediction layer whose activation function is a softmax and the size is 2. We use the pre-trained word embedding of GloVe (Pennington et al., 2014) with a word embedding dimension of 50, the optimizer Adam, and a learning rate of 0.001. We train classifiers until no improvement in the development set is observed anymore; all classifiers are of the same structure and have the same hyperparameters.

To minimize the mnemonic information induced by the article topic, we split the dataset controlling the topics as an independent variable: we group the articles by their topic and select some groups to be in the test set, some to be in the development set, and the rest to be in the training set. We ensure that either the development set or the test set has at least 10% of the articles

Media Bias	Training	Development	Test
Political Bias	39.25%	40.00%	42.82%
Unfairness	18.59%	17.48%	18.16%
Non-objectivity	39.84%	40.66%	43.31%

Table 3.4: The percentage of articles with each considered media bias type in the three datasets of our corpus.

	Political Bias	Unfairness	Non-objectivity
Majority	36.38%	45.01%	36.18%
RNN	75.60%	83.42%	75.42%
- Biased	69.41%	72.09%	69.57%
- Unbiased	81.80%	94.75%	81.13%

Table 3.5: The F_1 scores of RNN, majority baseline, and by-class performance of the three bias types.

in the whole dataset, obtaining 5394 articles in the training set, 755 articles in the development set, and 815 articles in the test set.

Table 3.4 shows the distribution of the labels in the corpus. To avoid the exploitation of portal-specific features, each article is thoroughly checked and all information regarding the portal it was taken from (e.g., "CNN's Clare Foran and Phil Mattingly contributed to this report") is removed.

Table 3.5 summarizes the performance of the developed RNNs in the three media bias classes. All classifiers outperform the majority baseline, achieving 75.60% for political bias, 83.42% for unfairness, and 75.42% for non-objectivity. Such a performance demonstrates the capability of the classifiers to detect topic-independent media bias features.

Looking closely at individual bias classes, we find that the RNN is good at predicting the absence of bias rather than bias. We interpret this because of the uneven distribution of the classes, especially in the unfairness (see Table 3.4).

3.2.3 Bias at Different Levels of Granularity

One key contribution of this chapter is to analyze the bias at different levels. To do so, we use the developed classifiers to output the predicted bias probability p_{art} of the test articles, i.e., the probability of being *politically biased*, *unfair*, or *non-objective*. We iteratively remove text segments from the article and use the classifier to again predict the bias probability p_{art-i}, where i denotes the index of the text segment in the article. The media bias strength

of a text segment t_i is estimated as $p_{art} - p_{art-i}$. If a text segment is relevant for prediction, we expect to see a significant decrease from p_{art} to p_{art-i}.

Based on this estimation of bias strength, we design three experiments to analyze and interpret the classifiers' predictions at the following levels of text granularity:

Word level (LIWC correlations) Related research suggests that bias is manifested at a larger granularity level, including the paragraph level as we showed (Chen et al., 2018b) and the clause level (Iyyer et al., 2014). To validate this, we use the LIWC categories to check the word level bias, because they have been used in Iyyer et al. (2014) to sample a set of sentences that may contain ideology bias.

In detail, for each sentence s_i, we compute its LIWC score of the category j as $|\{w_{i,k} \in c_j, k \in K\}| \,/\, |\{w_{i,k}, k \in K\}|$, where c_j denotes the words in LIWC category j, K denotes the bag-of-words in s_i, and $w_{i,k}$ denotes the k-th word in K. The Pearson correlation coefficient is used to measure the correlation between LIWC categories and media bias strength.

According to the Pearson correlations, most of the LIWC categories are not correlated with a high coefficient (neither positively nor negatively). However, the highest correlated categories are different among the three types of media bias. The categories that have the highest correlation with political bias are *negative emotion, anger*, and *affect*. This shows that politically biased articles tend to use emotional and opinionated words such as "disappoint", "trust", and "angry". For unfair articles, we see a higher correlation in *focus present*. Examples in this category are "admit", "become", and "determine". For non-objective articles, the bias is related to *percept* words such as "feel", "gloom", and "depict".

Sentence and paragraph-level (locations of media bias) After seeing bias at the word level, we further analyze to find biased sentences and paragraphs. The results tell us where are the biased parts of an article. To do so, we analyze the distribution of the media bias strength in the sentences and paragraphs (approximated as three continuous sentences). These values indicate which segment of a text mostly contains media bias.

Figure 3.3 visualizes the estimated media bias strength at the sentence and paragraph levels. As an example, we chose an article from Daily Kos, which is labeled as politically biased. In this article, we see a strong tendency to criticize Trump's claim, especially at the end of the article. At the paragraph level, our strength analysis of media bias successfully identifies the last paragraph as the most biased text segment. While at the sentence

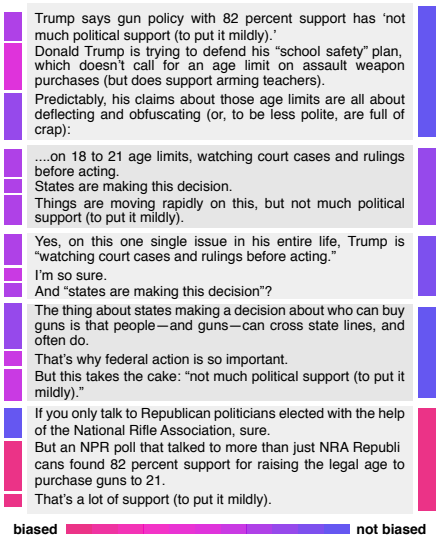


Figure 3.3: The media bias strength on sentence (left) and paragraph (right) level in an excerpt of one news article from the given corpus. The figure is reused from our paper (Chen et al., 2020a). Biased text segments are shown in red and unbiased text segments in blue.

level, we see that the last two sentences are most biased in the last paragraph. The second sentence seems to be a bit biased, perhaps because of the word usage of "trying to defend". However, we see that the analysis fails to identify the third sentence as politically biased. Still, given that the sentence or paragraph-level analysis is fully unsupervised, the reverse feature analysis seems to perform quite well.

Discourse level (media bias patterns) On top of sentences and paragraphs, we would like to see bias at a higher level. Here we analyze the patterns of the media bias strength across the different parts of an article's discourse. In particular, we split an article into four equally sized parts and computed the average media bias strength of the sentences for each part. The splitting is comparable to the so-called "inverted pyramid" structure in journalism, where a news article starts with the summary, important details, general and background info (Pöttker, 2003).

Figure 3.4 shows the identified media sequential patterns for the three bias types. We can notice that the media bias strengths for all articles in the second quarter are somewhat close. This is, in our opinion, because the second quarter of news articles usually contains some background information, which does not tend to be biased. We also see that all biased articles start with a neutral tone (close to mean) in the beginning and then emphasize the bias in the latter parts. Among the three media bias types,

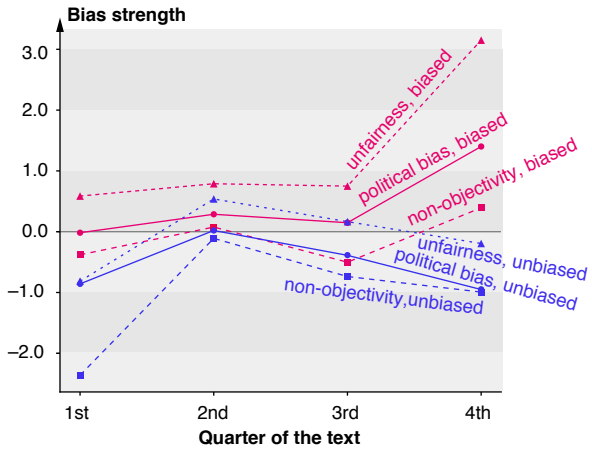


Figure 3.4: Patterns of the types of media bias as well as biased and unbiased text from our paper (Chen et al., 2020a). Values are normalized to have a mean of zero and a standard deviation of one. Positive values indicate a stronger bias and negative values indicate that text has a lower bias or is unbiased.

unfairness has the highest bias strength. Observing our corpus, we find that one typical way to be unfair is to report selected facts in favor of some entity, which leads to completely different word usage. On the other hand, for political bias, describing facts with positive or negative expressions is a common indicator of bias. Such a difference might be the reason why the classifiers can better discover unfair texts.

In this section, we contribute to present our method to analyze statement bias at different levels. The results show how the writer puts bias at different locations to make an article read biased (or unbiased). In the next section, we are going to use these bias locations as features to build a statement bias classification model.

3.3 Media Bias Analysis at Article-level

In the previous section, we study how to identify lower-level bias given the article-level bias, and the results show that a biased article has different bias locations compared to an unbiased article. In this section, we then use the bias locations as features to study how to identify article-level bias given lower-level bias.

First, we are aware that using word-level bias makes it difficult to capture article-level bias. Considering the following sentences from `allsides.com` reporting on the event "Trump asks if disinfectant, sunlight can treat coronavirus" demonstrate bias in different news portals:

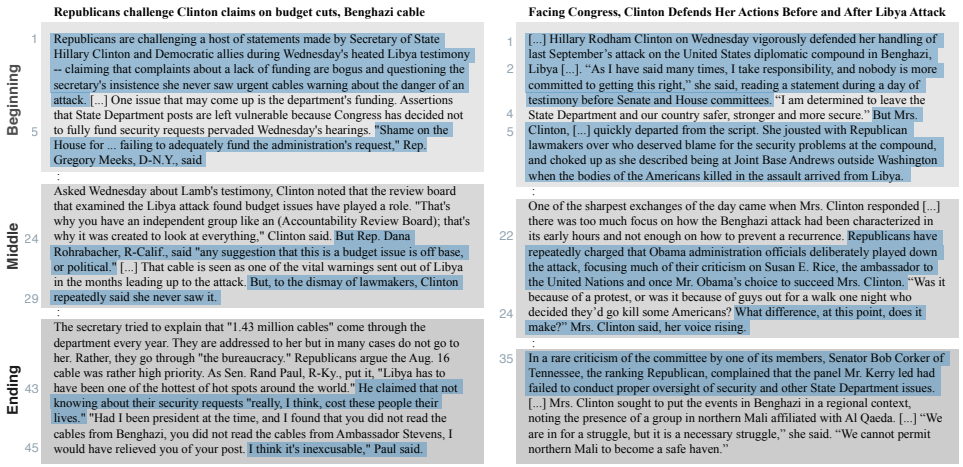


Figure 3.5: Excerpts of a biased article (left) and a neutral article (right) from the used dataset, reused from our paper (Chen et al., 2020b). All sentences labeled as having lexical or informational bias are highlighted; their position can be read from the numbers next to them.

The activists falsely claimed that Trump "urged Americans to inject themselves with disinfectant" and "told people to drink bleach."
— The Daily Wire, right-oriented

Lysol maker issues warning against injections of disinfectant after Trump comments
— The Hill, center-oriented

"This notion of injecting or ingesting any type of cleansing product into the body is irresponsible and it's dangerous," said Gupta.
— NBC News, left-oriented

From an NLP perspective, bias in the example sentences could be detected by capturing sentiment words, such as "falsely" or "irresponsible". However, bias detection becomes harder at the article level. To start with, we review what the locations of biased sentences look like in articles as shown in Figure 3.5. The figure shows two articles and their sentence-level bias from the used dataset. It becomes clear that the actual words in the biased sentences are not always indicative enough to distinguish biased from neutral articles, nor is the count of the biased sentences: Bias assessments on sentence level do not "add up". In this regard, the *position* of biased sentences is a better feature here.

The existing approaches to bias detection are transferred from other, less intricate text classification tasks. They largely model *low-level lexical information*, either explicitly, e.g. by using bag-of-words (Gerrish and Blei, 2011), or implicitly via neural networks (Gangula et al., 2019). Such approaches

tend to fail at the article level, particularly for articles on events not covered in the training data. The reason is that bias clues are subtle and rare in articles, especially event-*in*dependent clues. Altogether, modeling low-level information at the article level is insufficient to detect article-level bias, as we will later stress in experiments.

We study article-level bias detection both with and without allowing to learn event-specific information. The latter scenario is more challenging, but it is closer to the real world, because we cannot expect that the information in future articles always relates to past events. Inspired by ideas from modeling local and global polarities in sentiment analysis (Wachsmuth, 2015), we hypothesize that using *second-order bias information* in terms of lexical and informational bias at the sentence level is key to detecting article-level bias. To the best of our knowledge, no bias detection approach so far uses such information.

3.3.1 Dataset

To test the hypothesis that sentence-level bias is an important feature for article-level bias detection, we need data that is annotated for both bias levels. Fan et al. (2019) released a dataset on media bias, *Bias Annotation Spans on the Informational Level* (*BASIL*). The dataset contains 300 news articles on 100 events, three each per event. These three articles were taken from Fox News, New York Times, and Huffington Post, which have been selected as a representative of right-oriented, neutral, and left-oriented portals respectively.

On the article level, the dataset comes with manually annotated media bias labels (right, center, or left). While we noticed that more Fox News articles are right (50) than Huffington Post articles (10), the labels do not only rely on the source of the articles. Since we target bias in general rather than a specific orientation, we merged right and left to the label *bias*, and see center as *neutral*. Because both biased and unbiased articles include all three portals, we can be confident that the task is not detecting the source, but detecting the bias.

On the sentence level, each sentence has been manually labeled as having *lexical bias*, *informational bias*, or *none*. According to Fan et al. (2019), lexical bias refers to "how things are said", i.e., the author used polarized or otherwise sentimental words showing bias. On the other hand, sentences with informational bias "convey information tangential or speculative". In our experiments, we consider both settings where we separate the two types of bias and settings where we merge them.

3.3.2 Second-Order Bias Information

As mentioned, we study the correlation between sentence-level and article-level bias. Specifically, we examine whether article-level bias correlates with (a) the frequency of biased sentences, (b) their position in an article, and (c) their sequential order. For each correlation, we extract features and then train a respective machine-learning model. The code is available at `https://github.com/webis-de/EMNLP-20`.

Bias Frequency A straightforward way of leveraging sentence-level bias information is counting. Let an article with sentence-level bias labels $\{b_1, b_2, ..., b_n\}$ be given, where n is the number of sentences in the article and b_i is the label of the i-th sentence. Assuming that b_i is binary with $b_i = 1$ being bias, the *absolute bias frequency*, f_{abs}, is defined as:

$$f_{abs} = \sum_{i=1}^{n} b_i \tag{3.2}$$

Accordingly, the *relative bias frequency*, f_{rel}, is defined based on the length of the article as:

$$f_{rel} = \frac{\sum_{i=1}^{n} b_i}{n} \tag{3.3}$$

Bias Position We consider the positions of biased sentences as second-order features. Given a target number of positions, k, we first normalize the sentence-level bias annotations $\{b_1, b_2, ..., b_n\}$ into $\{\bar{b}_1, \bar{b}_2, ..., \bar{b}_k\}$, with $\bar{b}_i \in [0, 1]$. The higher $\bar{b}_i$, the more likely position i is biased. In detail, we first normalize $\{b_1, b_2, ..., b_n\}$ to $\{b'_1, b'_2, ..., b'_m\}$ by linear interpolation, where m (here set to 100) is larger than the largest n (and also larger than k). After the interpolation, b'_i is in the range of $[0, 1]$. Secondly, we "sample" from the b'_i to make the final sentence-level bias having length k. There are three "sampling" methods we explore: (1) average (take the average of the data points, (2) maximum (take the maximum value in the range, and (3) last (take the last data points). We treat this as a hyperparameter and find the best one by the validation set. We use this two-step normalization (upsampling and then downsampling) to avoid instability during sampling when n/k is not an integer.

Our goal is to predict the most likely article-level bias label, a^*, given the sentence-level bias. Formally, assuming that an article can be seen as a combination of its sentences, we have

$$a^* = \arg\max_a p(a \mid \bar{b}_1, \bar{b}_2, ..., \bar{b}_k), \tag{3.4}$$

where a is any possible bias label (0 for neutral and 1 for bias), and $p(a \mid \cdot)$ is the conditional probability of a, given a sentence-level bias sequence. According to Bayes' rule and given that $p(\bar{b}_1, \bar{b}_2, ..., \bar{b}_k)$ is irrelevant to the arg max, we can rewrite it as:

$$a^* = \arg\max_a p(\bar{b}_1, \bar{b}_2, ..., \bar{b}_k \mid a) \cdot p(a) \tag{3.5}$$

Assuming that each $\bar{b}_i$ is independent of other positions, we further simplify this as

$$a^* = \arg\max_a \prod_{i=1}^{k} p(\bar{b}_i \mid a) \cdot p(a), \tag{3.6}$$

which is a Naïve Bayes classifier, and each $p(\bar{b}_i \mid a)$ is the bias position feature we are interested in.

In the remainder, we simplify the notation $p(\bar{b}_i \mid a)$ to $p(\bar{b} \mid a)$. Estimating $p(\bar{b} \mid a)$ in each position for each a is difficult, since $\bar{b} \in [0, 1]$ and we cannot observe enough data points in that range on realistic text corpora. Instead, we therefore estimate $p(a \mid \bar{b})/p(a)$, where $p(a)$ can be properly estimated by the distribution of the labels, and $p(a \mid \bar{b})$ can be estimated well using a Gaussian Mixture Model.

Gaussian Mixture Model Given a set of m articles along with their bias labels, $\{a_1, a_2, ..., a_m\}$, we first retrieve the interpolated bias value in each position $b_{i,j}$ where i is the index of the position and j is the index of the article. $b_{i,j}, 1 \leq j \leq m$ can be seen as a distribution of the bias strength in one position i. For example, the distribution in Figure 3.6 shows the bias in the second position if we normalize the articles into 10 positions.

To model the distribution, we employ a Gaussian mixture model (GMM) (Reynolds, 2009). The assumption behind GMMs is that a distribution can be seen as a combination of Gaussian distributions, where each distribution is represented by its mean μ, its variance σ^2, and a weight w, the sum of all weights being 1. Modeling a GMM is unsupervised; we only need to set the number of mixtures we would like to have.

After applying GMM on $b_{i,j}, 1 \leq j \leq m$, the distribution of a bias position i is represented by a set of Gaussian mixtures, $\mathcal{N}_l(\mu_l, \sigma_l^2, w_l)$, where l is the index of mixtures. For each mixture, we can then learn its bias distribution by:

$$p(a = 1 \mid \mathcal{N}_l) = \frac{occur(\bar{b}_{i,j} \in \mathcal{N}_l, a_j = 1)}{occur(b_{i,j} \in \mathcal{N}_l)} \tag{3.7}$$

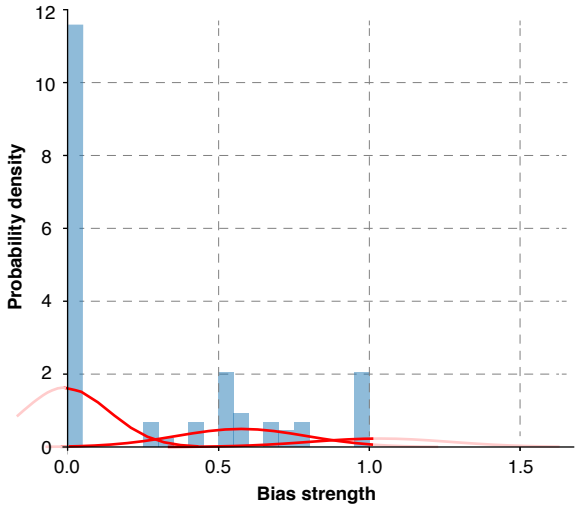


Figure 3.6: Bias strength in one position and the fitted Gaussian mixtures of it, reused from our paper (Chen et al., 2020b). The bias strength is the value of $\bar{b}_i$. Note that the y-axis is the probability density, i.e., the sum of all areas in bins or the sum of all areas under Gaussian mixtures is one.

To avoid zero probability in some mixtures, we also apply add-one smoothing. Then, the bias probability $p(\bar{b} \mid a = 1)$ in one position is:

$$p(\bar{b} \mid a = 1) \propto \frac{p(a = 1|\bar{b})}{p(a = 1)} \sim \frac{p(a = 1|\mathcal{N}_{\bar{b}})}{p(a = 1)}, \tag{3.8}$$

where $\mathcal{N}_{\bar{b}}$ is the mixture most likely generating $\bar{b}$.

Bias Sequence The Naïve Bayes classifier in Equation 3.6 assumes that each position is independent of other positions. We can also consider a position to depend on the previous positions. For example, under the assumption that each position depends on the one before, we can rewrite Equation 3.6 as:

$$a = \arg\max_{a} \prod_{i=1}^{k} p(\bar{b}_i \mid \bar{b}_{i-1}, a) \cdot p(a) \tag{3.9}$$

Then, we can further rewrite $p(\bar{b}_i \mid \bar{b}_{i-1}, a)$ as:

$$p(\bar{b}_i \mid \bar{b}_{i-1}, a) = \frac{p(a \mid \bar{b}_i, \bar{b}_{i-1})}{p(\bar{b}_{i-1} \mid a) \cdot p(a)} \tag{3.10}$$

In this equation, $p(\bar{b}_{i-1} \mid a)$ can be approached by the GMM as described, and the numerator of the equation can be seen as the transition probability in a Markov process. In particular, after finding the mixtures most likely

Training		Validation		Test	
Neutral	Bias	Neutral	Bias	Neutral	Bias
84	96	31	29	29	31

Table 3.6: Bias distribution of articles in the experiment setting.

generating $\bar{b}_i$, and $\bar{b}_{i-1}$, we estimate the transition probability $p(a|\bar{b}_i, \bar{b}_{i-1})$ as:

$$p(a \mid \bar{b}_i, \bar{b}_{i-1}) \sim p(a \mid \mathcal{N}_i, \mathcal{N}_{i-1}), \tag{3.11}$$

where $\mathcal{N}_i$ and $\mathcal{N}_{i-1}$ are the mixtures most likely generating $\bar{b}_i$ and $\bar{b}_{i-1}$ respectively. Again, we apply add-one smoothing when estimating the transition probabilities.

The previous equations can be easily extended to the case that each position is dependent on more than one position. However, longer dependencies imply fewer observations of each possible transition. As a result, we only test the first and the second-order Markov process below (i.e., dependence on the previous one or two positions).

3.3.3 Impact of Sentence-Level Bias

We first use the ground-truth sentence-level bias from the dataset. Thereby, we investigate the ideal case where the sentence-level bias can be detected perfectly (assuming the manual annotations are correct). The different types of sentence-level bias are also tested to understand if article-level bias is more correlated to a certain type.

In the experiment setting, the size of the training, validation, and test sets are 180, 60, and 60 articles, respectively. Additionally, we control the events in the sets to ensure there is no event overlap between the sets. The distribution of labels in each set and setting can be found in Table 3.6. As can be seen, the article-level labels are almost balanced, with some more biased than neutral articles. According to the distribution in the training set, we chose all-bias as the majority baseline in the later experiments.

As standard feature-based approaches, we employ an SVM and a logistic regression classifier based on word n-grams with $n \in \{1, 2, 3\}$. The considered n-grams are learned on the training set and lowercase. Hyperparameters such as cost and class balance are optimized on the validation set.

As a standard neural approach, we employ a pre-trained uncased BERT model using word embeddings as "features".[5] We fine-tuned the approach

[5]Cased and uncased BERT performed similarly in tests.

Feature	Classifier	Accuracy
–	All-bias baseline	0.52
n-grams (1–3)	SVM	0.52 (+0.00)
n-grams (1–3)	Logistic Regression	0.53 (+0.01)
Word embeddings	BERT	0.53 (+0.01)

Table 3.7: Accuracy of the three standard approaches and the all-bias baseline in article-level bias detection. The numbers in parentheses indicate the difference compared to the baseline.

and optimized the number of epochs for fine-tuning the training and validation set. Only the first 256 and the last 256 words of an article are used for bias prediction because the maximum sequence length of the BERT model is 512 tokens.

Tables 3.7 show the results of the basic article-level bias detection experiments, which address the effectiveness of standard classification approaches in article-level bias detection. With a maximum of 0.55, the accuracy of all classifiers is generally low for a two-class classification task. Given that the event information is not available, the classifiers seem to learn almost nothing: In the absence of event features, the classifiers are more forced to learn style or structural features. Yet, they turn out not to be able to do so without a proper design of such features. These results suggest that standard approaches are insufficient for article-level bias detection.

We prepare three types of sentence-level bias features, according to the descriptions in Section 3.3.2: For *bias frequency*, we consider a single feature SVM. We use a linear kernel and optimize its cost hyperparameter on the validation set. For *bias positions*, we compute the bias probability in each position and then apply either Naïve Bayes, in line with Equation 3.6, or an SVM. For *bias* sequences, we use the Markov process from Equation 3.9 to predict an article-level bias label. Besides, we use the probabilities $p(\bar{b}_i \mid \bar{b}_{i-1}, a)$ as features for an SVM. Finally, we also test *stacking* models. To test the effectiveness of each feature, we stack all three SVMs of each bias feature, as well as any two of the three SVMs as an ablation test.

The column $Acc(GT)$ of Table 3.8 shows the accuracy of employing ground-truth sentence-level bias features in predicting article-level bias. The SVM stacking classifier with bias frequency and sequence (F+S) performs best with an accuracy of 0.67. Stacking all features (F+P+S) achieves the same accuracy. In general, all feature and classifier combinations outperform all approaches found in Table 3.7.

Bias	Feature	Classifier	Acc (GT)	Acc (Pr)
Lex.	f_{abs}	SVM	**0.65**	**0.52**
	f_{rel}	SVM	0.63	0.48
	Bias Position	Naïve Bayes	0.55	0.48
		SVM	0.57	0.48
	Bias Sequence	Markov Process	0.50	0.50
		SVM	0.53	0.50
	F + P	SVM Stacking	**0.65**	**0.52**
	F + S	SVM Stacking	**0.65**	**0.52**
	P + S	SVM Stacking	0.52	**0.52**
	F + P + S	SVM Stacking	**0.65**	**0.52**
Info.	f_{abs}	SVM	0.57	0.52
	f_{rel}	SVM	0.52	0.52
	Bias Position	Naïve Bayes	0.55	0.50
		SVM	0.55	0.50
	Bias Sequence	Markov Process	0.48	0.48
		SVM	0.47	0.48
	F + P	SVM Stacking	0.55	0.52
	F + S	SVM Stacking	**0.58**	0.52
	P + S	SVM Stacking	**0.58**	0.52
	F + P + S	SVM Stacking	**0.58**	**0.57**
Any	f_{abs}	SVM	0.65	***0.67**
	f_{rel}	SVM	0.65	0.65
	Bias Position	Naïve Bayes	0.57	0.58
		SVM	0.52	0.52
	Bias Sequence	Markov Process	0.58	0.58
		SVM	0.42	0.42
	F + P	SVM Stacking	0.63	0.65
	F + S	SVM Stacking	***0.67**	0.62
	P + S	SVM Stacking	0.50	0.50
	F + P + S	SVM Stacking	***0.67**	0.62

Table 3.8: Accuracy of all evaluated combinations of features and classifiers in article-level bias detection based on ground-truth (GT) and predicted (Pr) sentence-level bias. F combines absolute (f_{abs}) and relative (f_{rel}) bias frequency, P stands for for bias position, and S for bias sequence. The best value for each bias type is marked bold. The best values overall are marked with *.

Among the features of sentence-level bias, bias frequency, and bias position can be exploited best by the SVM. While the bias sequence does not perform as well as the others, the stacking classifier using it yields the highest effectiveness. The bias sequence appears to be the weakest and sometimes brings a negative impact on the performance. However, there may be

	Training		Validation		Test	
	Neutral	Bias	Neutral	Bias	Neutral	Bias
Lexical bias	4 611	263	1 558	85	1 382	78
Informational bias	4 102	772	1 404	239	1 272	188
Any bias	3 839	1035	1 319	324	1 194	266

Table 3.9: Distribution of the different types of sentence-level bias in the settings for research question Q1. In the *Any bias* setting, a sentence is considered biased if it contains lexical and/or informational bias.

several reasons behind it. For example, the sequential features may be too subtle, such that our models (SVM and Markov process) are too sensitive to the tiny changes in the features. But, it may also be that a smarter combination strategy for the three different types of features is required; to keep the models simple, we tested only stacking. On the single features, the results show that an SVM is not always the best choice to utilize the features. In particular, Naïve Bayes and Markov process work better when dealing with informational bias and any bias.

Next, we take a closer look at the stacking part of Table 3.8, to analyze the feature's effectiveness. While using lexically biased sentences as features, the frequency features contribute more (combinations in stacking with F achieve the best results). On the other hand, while using informationally biased sentences as features, the sequential features are more important. In other words, to detect article bias, it is important to know the number of lexically biased sentences as well as the order of informationally biased sentences. Our interpretation is that the existence of lexical bias is already a strong clue for presenting bias, whereas informational bias has to be conveyed in a certain order or writing strategy (and thus is more difficult to capture).

Regarding the two types of sentence-level bias, the best results are observed for *any* bias. Using only informational bias leads to the lowest effectiveness. While there is more informational than lexical bias, as shown in Table 3.9, the classifiers seem to rely more on the lexical bias. The reason could be that lexical bias is easier to capture (by word usage), while informational bias clues, if any, are subtle. Still, including both types of bias (but not distinguishing them) works best.

To study how sentence-level bias detection can be utilized for article-level bias detection, we first present the results of applying the standard approaches to sentence-level bias detection in Table 3.10. Besides accuracy, we also show precision, since a high precision boosts the confidence in pre-

Bias	Feature	Classifier	Acc.	Prec.
Lex.	–	All-bias baseline	0.05	0.05
	n-grams (1–3)	SVM	0.13	0.13
	n-grams (1–3)	Logistic Regression	0.07	0.05
	Word embeddings	BERT	**0.95**	**0.38**
Info.	–	All-bias baseline	0.13	0.13
	n-grams (1–3)	SVM	0.13	0.13
	n-grams (1–3)	Logistic Regression	0.47	0.14
	Word embeddings	BERT	**0.86**	**0.37**
Any	–	All-bias baseline	0.18	0.18
	n-grams (1–3)	SVM	0.38	0.18
	n-grams (1–3)	Logistic Regression	0.69	0.23
	Word embeddings	BERT	**0.79**	**0.58**

Table 3.10: Accuracy (Acc.) and precision (Prec.) of the three standard approaches and the all-bias baseline in sentence-level bias detection. The highest accuracy and precision values for each bias type are marked bold.

Bias	Classifier	Precision	Recall	F_1
Lex.	Fan et al. (2019)	29.13	38.57	31.49
	Reimplementation	37.50	13.64	20.00
Info.	Fan et al. (2019)	43.87	42.19	43.27
	Reimplementation	58.62	32.08	41.46

Table 3.11: Classification results of Fan et al. (2019) and our reimplementation. Both use pre-trained BERT, but the exact dataset split of Fan et al. (2019) is unclear.

dicting sentence-level bias. We expect precision to be more important than recall since we use the predicted bias for computing the article-level bias features. We find that fine-tuned BERT is strongest in effectiveness. Matching intuition, and predicting lexical bias seems much easier than predicting informational bias.

3.3.4 Impact of Predicted Sentence-Level Bias

We first present the results of applying the standard approaches to sentence-level bias detection in Table 3.10. Besides accuracy, we also show precision, since a high precision boosts the confidence in predicting sentence-level bias. We expect precision to be more important than recall since we use the predicted bias for computing the article-level bias features. We find that fine-tuned BERT is strongest in effectiveness. Matching intuition, and predicting lexical bias seems much easier than predicting informational bias.

Since Fan et al. (2019) provides their results of using BERT on sentence-level bias classification, we also used BERT for comparison. To this end, we split the dataset into sets of the same *size* as Fan et al. (randomly with 6819 training, 758 validation, and 400 test instances). However, the actual distribution of labels is not provided by the authors. As shown in Table 3.11, the results of our reimplementation for predicting informational bias are comparable to their results (in terms of F_1-score), but it is much worse for predicting lexical bias. Note that lexical bias in the dataset is rather rare ($478/7984 \approx 6\%$). We thus assume that the difference between our and the original test set caused the difference.

We used the predictions of the best sentence-level bias classifier (i.e., BERT) to compute the bias features. The resulting effectiveness in article-level bias detection can be found in column $Acc(Pr)$ of Table 3.8. Comparing these results to those obtained from giving ground-truth sentence-level bias, we see a clear drop in the effectiveness, when using only lexical bias or only informational bias. Interestingly, however, the best configuration—with absolute bias frequency (f_{abs}) and SVM on any bias—is as good as the best one for ground-truth sentence-level bias. This means that using the predicted bias can sometimes be better than using ground-truth bias. We explain this by the fact that sentence-level bias classifiers are deterministic while human annotators may not, which can help our approaches learn more stable patterns in the features.

Overall, our approaches with sentence-level bias information outperform the standard approaches, underlining the impact of our approach. With an accuracy of 0.67, we outperform the standard approaches (0.53) by 14 points and the all-bias baseline (0.52) by 15 points. Regarding the different types of bias, the bias frequency is still the best feature, while the bias position and the bias sequence are weaker. The stacking model is the most effective in general.

3.4 Summary

This chapter has introduced a new media bias corpus containing news articles and their political bias labels in Section 3.1. We discussed the creation of the corpus and showed the characteristics of biased text via our discriminativeness analysis.

Based on the created corpus, in Section 3.2 we have studied political bias and unfairness in news articles. We have trained sequential models for bias detection and have applied a reverse feature analysis to demonstrate that it is possible to reveal at what granularity level and how sequential patterns

media bias is manifested. Specifically, we find that the last quarter of an article seems to be the most biased part. A significant "by-product" of our research is a new corpus for bias analysis. We believe this corpus can help, for example, investigate how journalists convey bias in a news article.

Lastly in Section 3.3 we have given evidence that the exploitation of low-level lexical information is insufficient to detect article-level bias — especially, if the dataset is small. To provide a complete picture, we have formulated three research questions related to article-level bias detection, in order (1) to assess the state of the art of event-dependent and event-independent bias prediction, (2) to learn about the relation between sentence-level and article-level bias, and (3) to study whether sentence-level bias can be leveraged to predict article-level bias. To tackle the detection of article-level bias, we have proposed and analyzed derived (second-order) bias features, including bias frequency, bias position, and bias sequence. As a main result of our research, we have shown that this new approach outperforms the best approaches existing so far. If bias detection can be done sufficiently robustly on the article level, we envisage, as a line of future research, the development of "reformulation" strategies and algorithms for the task of neutralizing biased articles (Pryzant et al., 2020).

In this chapter, we have equipped ourselves with the knowledge of media bias in news articles using NLP techniques. Specifically, we have explored the ways in which media bias manifests through computational analysis. As emphasized earlier, the awareness of media bias serves as the initial and critical step toward tackling it. Armed with this knowledge, the subsequent chapter will study using NLP to mitigate media bias.

4

Textual Media Bias Mitigation

> There are no facts, only interpretations.
>
> Friedrich Nietzsche

In the previous chapter, we have learned the characteristics of news media bias. With this in mind, the next step of this thesis is to study how to mitigate the three kinds of media bias: gatekeeping bias, coverage bias, and statement bias. From the results of our previously published papers (Chen et al., 2018b, 2021) and a paper under review (Chen et al., -), we discuss our bias mitigation strategies.

Section §4.1 studies the topic of mitigating gatekeeping bias. We discuss how to cast the gatekeeping bias mitigation problem into an NLP task, and we propose a neural network model for it (Chen et al., -). Moving on to the coverage bias mitigation in Section §4.2, we also discuss how to frame the text differently (reframe) in order to mitigate the bias (Chen et al., 2021). Finally in Section §4.3, we use the statement bias dataset created in Section §3.1 to develop a neural network-based model to transfer a left-oriented sentence to a right-oriented sentence and vice versa (Chen et al., 2018b). For all mitigation models, we perform both automatic and manual evaluations to provide empirical results that we can mitigate the three types of media bias.

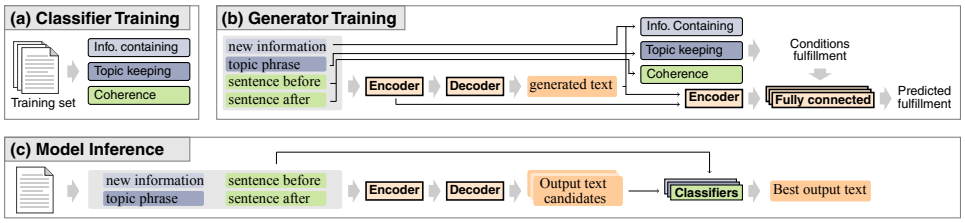


Figure 4.1: The three stages of our approach: (a) Training classifiers for each condition. (b) Training the multi-task learning model with the knowledge from the trained classifiers. Three fully-connected layers (one for each condition) are used to predict the condition fulfillment. The model learns to generate the text while satisfying the conditions. (c) Selecting the output text best fulfilling the conditions using the classifiers. Different conditions require different inputs.

4.1 Gatekeeping Bias Mitigation

Gatekeeping bias means the reporter selects what to report and what not to report. To mitigate such bias, we propose to focus on how to insert a piece of new information in the text. By doing so, the gatekeeping bias is reduced because more information from different angles is included.

From the perspective of natural language generation, such a mitigation strategy suggests a text-filling problem given its context. Therefore, we propose a multi-task learning problem. In particular, the first task is the text generation task while the generated text has to fulfill three conditions: (1) It contains the desired new information. (2) It is coherent with the context. And (3) it keeps the topic unchanged. In the following subsections, we first present a multi-task learning approach for bias mitigation, and then we introduce the corpus containing such bias. Later, we discuss the results of our model.

4.1.1 Gatekeeping Bias Mitigation Approach

In this subsection, we define the gatekeeping bias mitigation task that we consider, and then we present our approach to the task using knowledge distillation. Figure 4.1 gives an overview of the approach.

As discussed, we model gatekeeping bias mitigation as a sentence generation task:

> *Given new information, a topic should be kept, and a surrounding context, generates a text that contains the new information, keeps the topic, and is coherent with the context.*

We assume the new information is given a sentence, and the topic to be specified as a phrase. For context, we assume it is given as one sentence

before and one sentence after the text to be generated. Examples are shown in Figure 4.2.

Knowledge Distillation We approach the three conditions of the given task (information containing, topic keeping, and coherence) using a multi-task paradigm, where the primary task is text generation while the fulfillment of each condition serves as one auxiliary task. Most multi-task learning methods produce all output at once (Collobert and Weston, 2008), generating the text and predicting the conditions' fulfillment at the same time. However, we argue that the prediction should be made *after* generation, so the misfulfillment of the conditions can be backpropagated to guide the generation step.

We realize our approach in three stages, as illustrated in Figure 4.1:

(a) *Classifier training.* Given the three conditions, we train one classifier for each.
(b) *Generator training.* The classifiers are used as teacher models to guide the training of the generation model.
(c) *Model inference.* Given output text candidates, the classifiers find out the text best fulfills the conditions.

Inspired by research on knowledge distillation (Hinton et al., 2015; Liu et al., 2021), we use the classifiers as the teacher models and distill their knowledge into our approach. This could also be extended by further conditions, simply by using further classifiers and concatenating the embeddings with different inputs, if necessary. The main difference between Liu et al. (2021) and ours is that we add (c) to further improve the performance.

Classifier Training For topic bias, the classifier predicts the relation between the generated text, $\hat{y}$, and the desired information, f. Borrowing ideas from natural language inference (MacCartney, 2009), we distinguish three cases: (1) *information containing*, i.e., one text entails the other text, (2) *opposite information containing*, i.e., the two texts contradict each other, and (3) *neural* that the texts are unrelated to each other.

For topic keeping, a binary classifier predicts whether the generated text covers the desired topic t. Lastly, a binary classifier predicts whether the three sentences (sentence before $sent_{before}$, generated text, and sentence after $sent_{after}$) are coherent in sequence.

Generator Training For generation, we use an encoder-decoder architecture that learns to generate the target output text $\hat{y}$ given the inputs (topic

Training	Validation	Test
4,064	533	1,457

Table 4.1: The number of instances in training set, validation set, and test set in the topic bias mitigation dataset.

bias, entity, sentence before, and sentence after), as shown in Figure 4.1(b). After generation, we pass $\hat{y}$, t, e, $sent_{before}$, and $sent_{after}$ to the trained classifiers to get the predicted labels L, one label per classifier. We also feed $\hat{y}$ and all other inputs to the encoder again and concatenate their embeddings. Three fully connected layers (one for each condition) are used to predict labels $\hat{L}$, where the target labels L are given by the classifiers. Note that, unlike most multi-task learning approaches, the labels of the conditions are given by the teacher models on the fly and are not known before training.

Multiple losses have to be optimized: $\mathcal{L}_g$, the generation loss from comparing y with $\hat{y}$, and one condition loss $\mathcal{L}_i$ each from comparing $l_i \in L_i$ with $\hat{l_i} \in \hat{L}_i$ for condition i. The overall loss is as follows, where n is the number of conditions:

$$\mathcal{L} = \alpha_g \cdot \mathcal{L}_g + \sum_{i=1}^{n} \alpha_i \cdot \mathcal{L}_i, \tag{4.1}$$

where $\alpha_g \geq 0$, $\alpha_i \geq 0$ and $\alpha_g + \sum \alpha_i = 1$.

Model Inference During inference, we first generate a set of candidate texts $\hat{y}_j$ and then use the classifiers to predict the probabilities of condition fulfillment, as shown in Figure 4.1(c). The text with the highest aggregated probability is selected as the output. In particular, given all predicted probabilities $p_{i,j}$ of condition i and candidate j, the best output text is computed as follows, with the same weights as in Equation 4.1:

$$\hat{y} = \arg\max_{\hat{y}_j} \sum_{i=1}^{n} \alpha_i \cdot p_{i,j}, \tag{4.2}$$

4.1.2 Experiments

In this section, we report on the experiments we conducted to investigate the extent to which the proposed approach can perform topic bias mitigation. We present the setup, the dataset preparation, and the considered baselines.

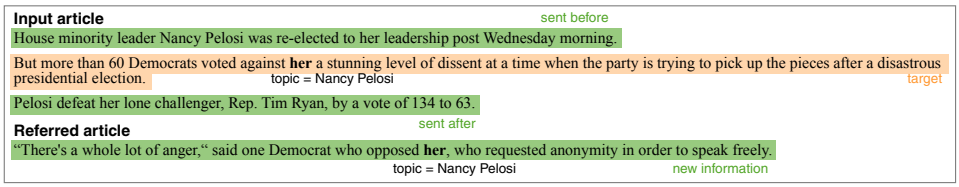


Figure 4.2: Examples from the BASIL dataset. Note that the original annotation does not include the location of the topic in this dataset. We label "her" here for better understanding. Similarly, both the *target* and the *new information* have negative sentiments toward the *topic*, "Nancy Pelosi". In this figure, the inputs ($sent_{before}$, $sent_{after}$, and *new information*) are in green and the output (*target*) is in orange.

Training			Validation			Test		
Info.	**Neutral**	**Oppo.**	**Info.**	**Neutral**	**Oppo.**	**Info.**	**Neutral**	**Oppo.**
4,064	3,182	3,466	533	528	333	1,457	1,039	917

Table 4.2: The number of instances in the training, validation, and test set for the information containing bias classifier training. We randomly selected neutral and opposite information containing (Oppo.) samples to have equal number as information containing (Info.) ones as much as possible.

Datasets We considered BASIL news corpus (Fan et al., 2019) as the gatekeeping bias mitigation corpus, which consists of three news articles each for 100 different events. The distribution of the training, validation, and test sets can be seen in Table 4.1.

We randomly selected 70 events as the training set, 10 events as the validation set, and 20 events as the test set. We only use the sentences which have bias annotated. An example of the dataset can be seen in Figure 4.2. A biased sentence is annotated with an entity (so-called *target* in the BASIL paper), such as a person or an event, and the sentiment toward the topic.

Classifiers and Model Here we detail the three training of the three classifiers outside the generation model. As a base model, we use `facebook/bart-base` from the Huggingface library (Wolf et al., 2019). Besides, we have the following three teacher classifiers.

Information Containing Classifier Given information to be added, f, and generated text, $\hat{y}$, this classifier predicts the probability of being contained. It is based on the pre-trained natural language inference model from `microsoft/deberta-base-mnli`. For a given sentence having a sentiment toward an entity (e.g., a politician), we took all sentences from other texts with the same reported event and sentiment as those as *information containing* samples. To finetune the classifier on the training data, we also needed

Training		Validation		Test	
Positive	**Negative**	**Positive**	**Negative**	**Positive**	**Negative**
960	960	124	124	315	315

Table 4.3: The number of instances in the training, validation, and test set for the topic adherence and the coherence classifiers. The positive label means topic keeping or coherence, respectively; the negative label means no topic keeping or incoherence.

samples with *opposite information containing* and *neutral* labels. For *opposite information containing*, we selected sentences from other news for the same event mentioning the same entity but opposite sentiment. For neutral, we randomly selected sentences from other news mentioning a different entity. Table 4.2 shows the distribution of the labels in the dataset. The performance of the classifiers is limited only, with a macro-average F_1-score 0.47 and an F_1-score of 0.61 for the information containing the label.

Topic Keeping Classifier Given a topic, t, and a generated text, $\hat{y}$, this classifier predicts whether $\hat{y}$ is relevant to t. For the negative instances, we randomly chose a sentence with a different annotated topic. We used the pre-trained `bart-base`(Lewis et al., 2020) and finetuned it on our datasets. The data distribution can be seen in Table 4.3. The accuracy of the model is 0.74.

Coherence Classifier Given $sent_{before}$, $\hat{y}$, and $sent_{after}$, this classifier predicts if $\hat{y}$ is coherent in between the others. We also used the pretrained `bart-base` and finetuned it on the given data. As negative instances, we randomly chose sentences to replace the sentence in the middle. The data distribution is the same in Table 4.3. The accuracy of the model is 0.84.

At inference time, we use the classifiers' output probabilities of *information containing*, *topic adherence*, and *coherence* for the three conditions in Equation 4.2. To train the classifiers, the positive and negative training instances are generated from the training instances in the content transfer experiment, and so for the validation and the test instances. As a result, we make sure the classifiers do not learn any information from the validation and test sets.

Baselines As baselines for our approach, we select the following two models that can be considered state-of-the-art to the best of our knowledge. We trained both baselines and optimized their hyperparameters on the validation set to create strong baselines.

Sequence-to-Sequence Model We compare to the closely related sequence-to-sequence training strategy in our previous paper (Chen et al., 2021). As a conditional text generator, the inputs are the sentence before, $sent_{before}$, the sentence after, $sent_{after}$, the topic, t, and the new information, f. The target output is the sentence in the middle. Similar to the setting in the previous section, the four inputs are concatenated together using special tokens as

$$\texttt{[SB]}\ sent_{before}\ \texttt{[/SB][SA]}\ sent_{after}\ \texttt{[/SA]}$$
$$\texttt{[F]}\ f\ \texttt{[/F]}\ \ \texttt{[T]}\ t\ \texttt{[/T]},$$

where the bracketed symbols are special tokens. As for our approach, the base model is `bart-base`.

Error Correction Model On the other hand, we consider the architecture proposed by Thorne and Vlachos (2021). Given an input claim, the model conditionally generates a corrected version of the claim based on retrieved evidence from Wikipedia. In our case, instead of using the retrieval component, we directly provide the ground-truth evidence to the input. In particular, we train a sequence-to-sequence model whose input consists of the sentence before, $sent_{before}$, the sentence after, $sent_{after}$, as well as the topic, t, concatenated with special tokens. The target then is the sentence in the middle. We used the trained model to generate the first draft output. Then, we trained an error correction model with the draft output and the new information, f, as input, and the target, y, as output. In other words, we have an ideal error correction case here: we exactly know the best f to guide the draft output.

4.1.3 Results and Discussion

This subsection discusses the automatic and manual evaluation results of our approach and the baselines. We analyze selected examples qualitatively and discuss the hyperparameters of the model.

Automatic Evaluation We first evaluate the generated texts and the fulfillment of the input conditions using ROGUE F_1-scores and available automatic metrics:

ROUGE F_1-Scores Table 4.4 shows that *our approach* outperforms both baselines. However, the differences between the scores of the three approaches are small.

Approach	Rouge-1	Rouge-2	Rouge-L
Chen et al. (2021)	17.17	3.20	13.18
Error correction	16.52	2.69	12.64
Our approach	**17.44**	**3.22**	**13.37**

Table 4.4: Rouge-{1, 2, L} F_1-scores of the two baselines and our approach. The best score in each column is marked bold.

Approach	Information containing↑	Topic keeping↑	Coherence↓
Chen et al. (2021)	.590	.412	18.19
Error correction	**.615**	**.571**	18.67
Our approach	.591	.479	**17.88**
w/o selection	.586	.391	18.08

Table 4.5: Automatic evaluation: Proportion of texts fulfilling the information containing, topic adherence and coherence conditions. *w/o selection* denotes the proportion before candidate selection. The best score per column is marked bold.

Condition Fulfillment We consider the following automatic metrics to evaluate the condition fulfillment. [1] For information-containing, we used the BERTScore (Zhang et al., 2020) with its best model `deberta-xlarge-mnli` from Microsoft to predict the similarity between the generated text and the new information. For topic keeping, we follow Yin et al. (2019) to use a vanilla `bart-large-mnli` model as the zero-shot topic classifier to predict the probability that the generated text has the desired entity. Finally, for coherence, we concatenated $sent_{before}$, generated text, and $sent_{after}$ as a single string and then computed the perplexity based on GPT-2 (Radford et al., 2019).

Table 4.5 shows that *our approach* has the lowest perplexity while it has the second-best performance in information containing and topic keeping. Such unstable results illustrate the limitation of these two baselines. We also see that the candidate selection (see Figure 4.1c) improves fulfillment of the three conditions, especially topic-keeping. However, the two baselines and the two variations of our approach are all very close to each other using automatic evaluations.

Manual Evaluation Since the automatic evaluation only approximates the actual quality, we also carried out a manual study where humans judged the information containing, topic keeping, and coherence of the generated

[1] We refrain from using the teacher classifiers as evaluators since they are integrated into our approach.

Approach	Information containing↑	Topic keeping↑	Coherence↓
Chen et al. (2021)	0.70	0.73	0.98
Error correction	0.72	0.69	**0.99**
Our approach	**0.76**†	**0.76**†	**0.99**

Table 4.6: Manual evaluation (main results): Mean scores of information containing, topic adherence, and coherence on each dataset. The best score in each column is marked bold. The †symbols denoting a significance ($p < 0.05$) comparing to the second best.

sentences. For the study, we randomly selected 100 instances from each dataset (200 in total). We showed the participants the sentence before and after, the topic, and the new information. On this basis, we asked them three questions regarding the generated text:

Q1. What is the relationship between the sentence and the new information?
{ *The sentence entails the new information* (*1*)
| *The new information entails the sentence* (*1*)
| *The sentence partly entails the new information* (*0.5*)
| *The new information partly entails the sentence* (*0.5*)
| *Neither entailment nor contradiction* (*0*)
| *They partly contradict each other* (*-0.5*)
| *They contradict each other* (*-1*) }

Q2. Does the sentence cover the given topic?
{*yes* (*1*) | *partially* (*0.5*) | *no* (*0*) }

Q3. Is the sentence coherent with the others?
{*yes* (*1*) | *partially* (*0.5*) | *no* (*0*) }

The numbers in parentheses indicate the scores of each answer. For information containing(Q1), entailment from either direction has a positive score while contradiction has a negative score.

We used Amazon Mechanical Turk to acquire four annotations per instance, at a cost of $0.33 per instance. To get more reliable results, we required the workers to have at least a 95% approval rate and at least 10,000 approved HITs. Furthermore, we applied the multi-annotator competence estimation (MACE) measure (Hovy et al., 2013) to estimate the final label. MACE achieved around 90% accuracy in predicting the labels.

Table 4.6 presents the manual evaluation results. For *information-containing*, our approach has the best score compared to the other two baselines. In terms of topic-keeping, our approach is still the best while error

Topic. Donald Trump *New information.* In this case, Trump is sidestepping Congress so that he can gain access to certain federal funds without congressional approval.
Chen et al. (2021). Ms. Coulter fired back shortly afterward. **"This is a power grab by a president who has disregarded the will of the American people," Mr. Trump said.** White House officials rejected criticism from across the ideological spectrum that Mr. Trump was creating a precedent that future presidents could use to ignore the will of Congress. *Error correction.* Ms. Coulter fired back shortly afterward. **Mr. Trump, who has repeatedly called for the repeal of the Affordable Care Act, called the bill "unconstitutional" and called it a "disaster" for the country.** White House officials rejected criticism from across the ideological spectrum that Mr. Trump was creating a precedent that future presidents could use to ignore the will of Congress. *Our approach.* Ms. Coulter fired back shortly afterward. **"It's a dangerous precedent" Mr. White House officials said in a brief statement.** White House officials rejected criticism from across the ideological spectrum that Mr. Trump was creating a precedent that future presidents could use to ignore the will of Congress.

Table 4.7: Sample from a news article Fan et al. (2019). The bold sentences are generated by the baselines and our approach, given topic and reference fact.

correction is the worst. The reason for this could be the two-step process harms topic-keeping. For the *coherence* condition, all approaches perform almost equally well, only Chen et al. (2021) is slightly worse.

Comparing the two baselines, we found that Chen et al. (2021) is better in generating topic-keeping texts, while the error correction model is better in generating texts containing the desired information. Conceptually, Chen et al. (2021) was developed to capture the frame in the texts as mentioned in Section §4.2, while the error correction model was developed to fix the *error* in the texts. The evaluation results also reflect such designed advantages of the two models.

Qualitative Analysis Exemplarily, we look at one sample of generated texts of the three approaches in Table 4.7.

As shown in Table 4.7, the new information to be inserted is about the comment that Trump was sidestepping Congress to access certain federal funds. Here, the model of Chen et al. (2021) generates an ironic sentence where Trump criticizes himself. The error correction model also talks about Trump, but "the repeal of the Affordable Care Act" has nothing to do with the fact. Our approach negatively comments that "It's a dangerous precedent" with respect to both the topic bias and the subsequent sentence on the White House officials.

In summary, the example suggests that the model of Chen et al. (2021) tends to generate text without much detail. The error correction model can provide more precise details, but part of the details tend to go wrong. Overall, our approach generates the most reasonable texts.

Hyperparameters To optimize our approach and the baseline models, we tuned their hyperparameters to maximize performance on the validation set. In particular, we considered the number of training steps and weights in Equation 4.1. We logged the results in every half epoch, with a maximum of 5 epochs of training. For the weights, we set the minimum value to be 0.1 and the maximum to be 0.7 (since $0.1 + 0.1 + 0.1 + 0.7 = 1$). We validated the model in every combination of the weights with a grid size of 0.1. This gives a total of 84 combinations.

In the end, all models were saturated between the second and the third epoch. For the weights, we found the best combination is 0.7 for generation loss and 0.1 for all conditions. The difference between the best combination and the worst one was about five points in terms of the ROUGE score. The high generation loss suggests that it is harder to generate news texts, so the models have to learn more from the generation loss.

4.2 Coverage Bias Mitigation

In this section, we study how to mitigate the other kind of bias: coverage bias. As has been said, coverage bias focuses on the visibility of each side of an issue. Therefore, our mitigation strategy is to change such visibilities by framing the text differently.

Framing is frequently used in media, to reorient how audiences think (Chong and Druckman, 2007), or to promote a decided interpretation. For example, when talking about a certain law one may emphasize its *economic* impact or its consequences regarding *crime*. With this in mind, we study coverage bias mitigation in terms of changing the frame, in order to change the perspective of the issue.

In detail, changing the frame can be a strategy to communicate with opposing camps of audiences, and, sometimes, just replacing specific terms can be enough to reach a reframing effect. Consider in this regard a reporter who may prefer to use "undocumented worker" instead of "illegal aliens" in left-leaning news (Webson et al., 2020). While still referring to the same people, the former can provoke a discussion of the economic impact of hiring them; the latter may raise issues of crime and possible de-

(a) **Economic Frame** (original text)
Key Congressional backers of the measure, sponsored by Senator Alan K. Simpson, Republican of Arizona, and Romano L. Mazzoli, Democrat of Kentucky, wanted a flexible spending limit. **Implicit in the debate and the stalemate that left the bill to die when Congress adjourned was a recognition that the cost of immigration reform would be high, although no one knew how high.** Without reform, though, the presence of what may be six million illegal aliens in this country exacts an economic and social toll.
(b) **Legality Frame** (reframed text)
Key Congressional backers of the measure, sponsored by Senator Alan K. Simpson, Republican of Arizona, and Romano L. Mazzoli, Democrat of Kentucky, wanted a flexible spending limit. **"It's time for Congress to take action," says a spokesman for the bill's sponsors, who want a flexible spending limit.** Without reform, though, the presence of what may be six million illegal aliens in this country exacts an economic and social toll.
(c) **Crime Frame** (reframed text)
Key Congressional backers of the measure, sponsored by Senator Alan K. Simpson, Republican of Arizona, and Romano L. Mazzoli, Democrat of Kentucky, wanted a flexible spending limit. **"Illegal aliens' is a growing problem in the country," says a spokesman for the measure's sponsors.** Without reform, though, the presence of what may be six million illegal aliens in this country exacts an economic and social toll.

Table 4.8: (a) Sample text from the media frames corpus Card et al. (2015). The bold sentence is labeled with the *economic* frame. Having reframed the sentence with the proposed approach, the text remains largely coherent and topic-consistent while showing the *legality* frame (b) and *crime* frame (c), respectively.

portation. Such low-level style reframing has been studied in recent work (Chakrabarty et al., 2021).

Usually, reframing requires rewriting entire sentences rather than single words or phrases. Table 4.8 illustrates the change of a sentence from the economic frame (a) to the legality frame (b) and the crime frame (c). While the original text emphasizes the cost of immigration reform, the legality-framed text quotes that "It's time for Congress to take action," and the crime-framed text includes the notion of "illegal aliens". The terms "bill" and "measure" in the respective reframed versions ensure the topical coherence of the texts. Two facts become clear from the example, namely that reframing needs (1) notable rewriting to shift the focus, and (2) overlapped entities to ensure topic consistency.

To work in the real world, a computational reframing model needs to be able to rewrite sentences completely. At the same time, the model has to

preserve the context, by maintaining coherence and topic consistency. Towards these goals, we propose to treat reframing as a *sentence-level fill-in-the-blank* task: Given three consecutive sentences plus a target frame, mask the middle sentence and generate a sentence that connects the preceding and the succeeding sentence in a natural way and that conveys the target frame. This task implies three research questions: (1) How to tackle a sentence-level fill-in-the-blank task in general? (2) How to generate a sentence with a specific frame? (3) How to make the sequence of sentences coherent?

The sentence-level blank filling is a new and unsolved task. We approach this task via controlled text generation, that is, by tweaking the input and output of a sequence-to-sequence model where the masked sentence is the target output, and the preceding and the succeeding sentences are the inputs (Section §4.2.1). For the second and third research questions, we propose three training strategies: (a) *framed-language pretraining*, to finetune the model on all framed texts to learn the framed "language", (b) *named-entity preservation*, to support the model in maintaining important entities extracted from the masked sentence, and (c) *adversarial learning*, to show the model undesired output texts in order to learn to avoid them.

Based on the corpus of Card et al. (2015) with annotated sentence-level frames, we empirically evaluate the pros and cons of each strategy and combinations thereof. The results reveal that our approach changes sentences properly from the original to the target frame in most cases. Some "reframing directions" remain challenging, such as from crime to economic. We find that obtaining high scores for all assessed dimensions at the same time is hard to achieve; for example, the adversarial learning strategy gives a strong signal toward the target frame at the expense of lower coherence. The implied trade-offs suggest that reframing technology should be configurable when applying it in a real-world scenario to put different stress on each sentence.

4.2.1 Reframing Model

We now present our approach to sentence-level reframing. We discuss how we tackle the reframing problem as a fill-in-the-blank task, and we propose three training strategies to generate a sentence that is framed as desired and that fits the surrounding text. Figure 4.3 illustrates our approach.

As discussed, reframing implies two problems: (1) To rewrite entire sentences from a text as much as needed in order to encode a given target frame, and (2) to maintain coherence and topic consistency concerning the context

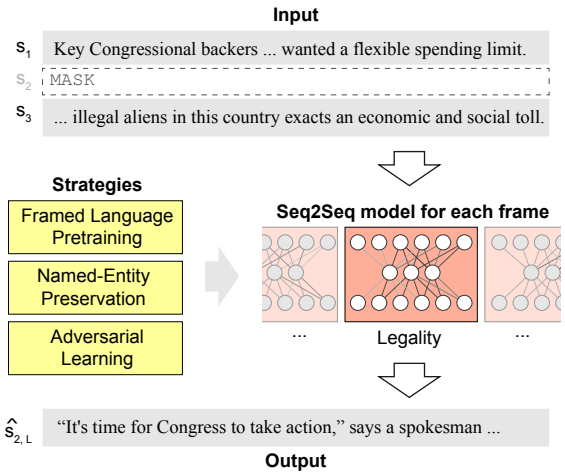


Figure 4.3: Illustration of our approach, reused from our paper (Chen et al., 2021). The sequence-to-sequence model trained on the desired target frame (here, *Legality*) takes the context sentences (s_1, s_3) as input and s_2 as target output. After applying the three training strategies, the model learns to decode [MASK] to the text addressing "It's time for Congress to take action".

given in the text. To tackle both problems simultaneously, we propose to treat reframing as a specific type of sentence-level fill-in-the-blank task.

In particular, let a sequence of three contiguous sentences, $\langle s_1, s_2, s_3 \rangle$, be given along with a target frame, f. The middle sentence, s_2, is the sentence to be reframed, and the other two sentences define the context taken into account for s_2. The fill-in-the-blank idea is to mask s_2, such that we have $\langle s_1, \texttt{[MASK]}, s_3 \rangle$. The task, in turn, is to then decode the masked token `[MASK]` to $\hat{s}_{2,f}$, a variation of the sentence s_2 that is reframed to f and both coherent and topic-consistent to s_1 and s_3.

No proper solution exists for this task yet, and only little prior work has addressed closely related problems. To approach the task, we propose a sequence-to-sequence model $r(\cdot)$ where the input is the two context sentences, $\langle s_1, s_3 \rangle$, and the output to be generated is s_2. To consider frame information in rewriting, we train one individual frame-specific model $r_f(\cdot)$ for each frame f from a given set of target frames, F, such that

$$\forall f \in F : r_f(s_1, s_3) \sim \hat{s}_{2,f} \tag{4.3}$$

4.2.2 Training Strategies

To better control the text generated by the model, we further guide the training process, by additionally considering the following three complementary training strategies. All three aim to provide extra information to the reframe-

ing model. In Section §4.2.4, we experiment with variations of the models to test each strategy and their combinations thoroughly.

Framed-Language Pretraining ($\mathcal{S}_{\mathrm{F}}$) Due to the complexity of manual annotation, we can expect only a limited number of task instances for each frame $f \in F$ in practice, so the models may have insufficient knowledge about how to generate framed language. To mitigate this problem, the first strategy we propose is to *pretrain the reframing model on all available text of any frame $f \in F$*. After that, this pretrained model will be further fine-tuned using instances from one particular frame.

Named-Entity Preservation ($\mathcal{S}_{\mathrm{N}}$) Given that a complete sentence is to be generated, a reframing model may mistakenly generate off-topic and incoherent text, if not controlled for. To avoid this, the second strategy is *to encode knowledge about the named entities to be discussed*. In particular, the set of named entities, N, can be extracted from s_2 and added to the input of the model.[2] Then, the input of the model can be extended to s_1 `[NE]` N `[/NE]` s_3, where `[NE]` and `[/NE]` are special tokens to indicate the start and ending of named entities.

Adversarial Learning ($\mathcal{S}_{\mathrm{A}}$) During training, the instances fed to the default model are all "positive" samples where the output s_2 comes from the same sentences $\langle s_1, s_2, s_3 \rangle$ the input sentences s_1 and s_3 are from. While this helps learning to generate coherent text, it impedes learning reframing. For example, if the goal is to encode the crime frame in $\hat{s}_{2,f}$, but s_1 and s_3 are from the economic frame, the model is likely to generate economic text, because it learns to reuse frame information encoded in s_1 and/or s_3 based on its experience. Inspired by adversarial learning, our third strategy is thus *to add "negative" training instances where the output sentence $\bar{s}_{2,f}$ is from the target frame*, but possibly incoherent and/or topic inconsistent to the input.

In the given example, $\bar{s}_{2,f}$ would be a sentence with the crime frame. In case we combine adversarial learning with named-entity preservation, $\bar{s}_{2,f}$ is chosen from all sentences s_2 in a given training set, such that the named entities of $\bar{s}_{2,f}$ and s_2 are as similar as possible. In case not, we choose a random sentence s_2 as $\bar{s}_{2,f}$. Conceptually, we thereby force the model to discard any possible input frame features. We note that this learning strategy likely harms the coherence and topic consistency of the generated

[2] We use the pretrained model *en_core_web_lg* from spaCy for named entity recognition in our experiments.

text, as $\bar{s}_{2,f}$ will often not fit to s_1 and s_3. We can control this effect, though, through careful use of the strategy, training only a few epochs.

4.2.3 Reframing Dataset

In this section, we describe how we prepare the corpus we use to create training and test instances for the sentence-level fill-in-the-blank task.

The Media Frames Corpus To analyze media framing across different social issues, Card et al. (2015) built a corpus that comprises 35,701 news articles (published between 1990 and 2012 in 13 news portals) in the US, addressing the topics of the death penalty, gun control, immigration, same-sex marriage, and tobacco.[3] Each article is annotated at span level for 15 general frames of the *Policy Frames Codebook* (Boydstun et al., 2013) in terms of the primary frame, the title's frame, and the span-level frame. Card et al. (2015) truncated articles to have at most 225 words.

Data Preprocessing Following several works in frame analysis (Naderi and Hirst, 2017; Hartmann et al., 2019), we focus on the five most frequently labeled frames in the corpus, accounting for about 60% of all labels. Examining these frames, we observed that two of them are hard to distinguish in various cases, namely *6: Policy prescription and evaluation* and *13: Political*.[4] Hence, we merge those two, ending up with a set $F = \{e, l, p, c\}$ of four frames:

- e. **Economic.** Costs, benefits, or other financial implications;
- l. **Legality, constitutionality, and jurisprudence.** Rights, freedoms, and authority of individuals, corporations, and government;
- p. **Policy prescription and evaluation + Political.** Discussion of specific policies aimed at addressing problems, or considerations related to politics and politicians, including lobbying, elections, and attempts to sway voters;
- c. **Crime and punishment.** Effectiveness and implications of laws and their enforcement.

For the sentence-level fill-in-the-blank task, we split the corpus articles into a training, a validation, and a test set. Each of the latter two comprises 3000

[3] We use the updated version from the authors' repository, `https://github.com/dallascard/media_frames_corpus`. Thus, the data distribution differs from the one of Card et al. (2015).

[4] Naderi and Hirst (2017) reported similar observations.

#	Frame	Training	Validation	Test
e	Economic	6 605	883	888
l	Legality c.a.j.	15 313	1 568	1 656
p	Policy p.a.e. + Political	20 903	2 169	2 109
c	Crime	10 726	1 144	1 257
	All four frames	53 547	5 764	5 910

Table 4.9: The number of fill-in-the-blank instances in the training, validation, and test set for each frame. Note that the four frames are not evenly distributed.

pseudo-randomly selected articles, 600 for each of the five given topics. The training set includes the remaining 29,701 articles. For each set, we collected all sentences from the respective articles that are labeled with one of the four considered frames. A sentence is considered to be labeled, if any part of the sentence is labeled. In case a sentence has more than one frame label, the sentence is associated with all the labels. For each of these framed sentences, s_2, we obtain its predecessor, s_1, and its successor s_3. Together, they form one data instance, as in subsection 4.2.1, where the input is the tuple of $\langle s_1, s_3 \rangle$ and the output is s_2.

To avoid those outliers misleading the learning process, we do not take all instances, but we filter instances by sentence length as follows. We consider only sentences s_1, s_2, and s_3 with at least five and at most 50 tokens each, and include only instances where s_2 has a similar length to the mean length of s_1 and s_3, with a tolerance of $\pm$ 50%. About 62% of the instances remain after this step.

The distribution of the framed sentences among the training, validation, and test sets is shown in Table 4.9. Note that the test set here is the one built for automatic evaluation. The test set for the manual evaluation is discussed in Section 4.2.5.

4.2.4 Experiments

We present the results of the pilot study for the different reframing approaches, the metrics for automatic evaluation, and the design of crowdsourcing tasks for manual evaluation.

Operationalizing Reframing We rely on transformers (Wolf et al., 2020) as the basis for reframing. The pretrained weights of the sequence-to-sequence model are from *T5-base* (Raffel et al., 2020). The three strategies from subsection 4.2.1 require pretraining on framed language ($\mathcal{S}_{\mathrm{F}}$) or a fine-tuning of the reframing model ($\mathcal{S}_N$ and $\mathcal{S}_{\mathrm{A}}$) respectively. For $\mathcal{S}_{\mathrm{F}}$ and $\mathcal{S}_{\mathrm{N}}$, the

models were optimized on the validation set; for the adversarial learning strategy, $\mathcal{S}_{\text{A}}$, we trained for three epochs in order not to harm the coherence of the output too much. Since each strategy can be applied independently, we considered eight reframing model variations, ranging from applying no strategy ($\mathcal{S}_{\emptyset}$) to applying all three strategies ($\mathcal{S}_{\text{FNA}}$).

Baselines The variant without any strategy, $\mathcal{S}_{\emptyset}$, can be considered as a baseline. Few other models exist so far that are suitable baselines for tackling the reframing task, but one is *GPT-2* (Radford et al., 2019). Specifically, we finetuned GPT-2 on all text available for each frame to have four framed versions of GPT-2. During application, we used s_1, the sentence before the target sentence, as the prompt and generated $s_{2,f}$ with the finetuned GPT-2. We also tested framed-language pretraining, $\mathcal{S}_{\text{F}}$, with GPT-2. To obtain *GPT-2* + $\mathcal{S}_{\text{F}}$, we first finetuned GPT-2 on all framed text and then further finetuned it on the text of the respective frame.

Pilot Study In our manual evaluation below, we focus on three of the eight variations of our approach, for budget reasons and to keep the evaluation manageable:

1. *B.Coherence*. The model variation generates the most coherent sentences.
2. *B.Framing*. The model variation generates the most accurately framed sentences.
3. *B.Balance*. The model variation achieves the best balance between coherence and framing.

We ranked the models in a pilot study where we randomly selected 10 instances $\langle s_1, s_2, s_3 \rangle$ from the test set for each of the four frames in F, 40 instances in total. We used the respective variation to reframe all sentences s_2 to the economic frame. Then, we judge each reframed sentence by assigning scores in response to the following questions:

Q1. Is the sentence coherent with other sentences?
{*yes* (*2*) | *partially* (*1*) | *no* (*0*)}

Q2. Does the sentence cover economic aspects?
{*yes* (*2*) | *partially* (*1*) | *no* (*0*)}

Table 4.10 shows the averaged scores. Pearson's correlation r for the two questions was 0.90 and 0.66 respectively, suggesting that the judges agreed substantially in the rankings. Based on the average scores, we made the following choices:

	Q1 (Coherence)			Q2 (Framing)			Balance
Strategy	**A1**	**A2**	**Avg.**	**A1**	**A2**	**Avg.**	**H. Mean**
$\mathcal{S}_{\emptyset}$	4	6	0.96	5	7	0.49	0.65
$\mathcal{S}_{\mathrm{F}}$	1	2	1.30	6	6	0.50	0.72
$\mathcal{S}_{\mathrm{N}}$	3	3	1.10	4	5	0.58	0.76
$\mathcal{S}_{\mathrm{A}}$	7	7	0.50	1	2	**0.89**	0.64
$\mathcal{S}_{\mathrm{FN}}$	2	1	**1.35**	7	2	0.57	0.80
$\mathcal{S}_{\mathrm{FA}}$	8	8	0.16	8	8	0.27	0.20
$\mathcal{S}_{\mathrm{NA}}$	5	4	0.99	2	1	0.88	**0.93**
$\mathcal{S}_{\mathrm{FNA}}$	6	5	0.90	3	2	0.70	0.79

Table 4.10: The pilot study rankings by the two annotators (A1, A2) along with the average of their scores from the eight model variations, resulting from the three training strategies $\mathcal{S}_F$, $\mathcal{S}_N$, and $\mathcal{S}_A$. Three framing variations are ranked second for A2 due to identical average scores. The right-most column shows the harmonic mean of the two average scores of both questions.

1. *B.Coherence*. $\mathcal{S}_{\mathrm{FN}}$ (coherence score 1.35)
2. *B.Framing*. $\mathcal{S}_{\mathrm{A}}$ (framing score 0.89)
3. *B.Balance*. $\mathcal{S}_{\mathrm{NA}}$ (harmonic mean 0.93)

We chose $\mathcal{S}_{\mathrm{NA}}$ in the latter case since it showed the maximum harmonic mean of the two scores. In addition, we manually evaluated $\mathcal{S}_{\emptyset}$, the baseline model without any training strategies.

4.2.5 Evaluation Metrics

To answer the research questions, we considered three dimensions for the different approaches: coherence, correct framing, and topic consistency, both in automatic and manual evaluation.

Automatic Evaluation We used ROUGE scores to approximate the overall quality of the generated texts. As ROUGE requires ground-truth information, we considered only those cases where the target frame matches the frame where the test instance stems from. To quantify the effect of reframing, we compiled a vocabulary for each frame by taking the 100 words with the highest TF-IDF values, where each sentence of a frame was seen as one document. By counting the number of words occurring in the respective vocabulary, we could get a rough idea of the reframing impact.

Manual Evaluation For the manual evaluation, we randomly selected 15 instances for each frame from the test set, 60 instances in total. For each instance, we applied the reframing models along with baselines to reframe it to the four frames in F. Among the reframed cases one was of type *intra-frame generation* (i.e., it had the frame from the original sentence); the other cases were of the *inter-frame generation* type. These two types will be discussed separately.

We used Amazon Mechanical Turk to evaluate the selected test set, where each instance was annotated by five workers (for \$0.80 per instance). For reliability, we employed only master workers with more than 95% approval rate and more than 10k approved HITs. The percentage of agreement with the majority is 73% on average in our experiments. The workers were provided three continuous sentences and were asked to judge the middle one (the one generated) by answering six questions:

Q1. Is the sentence coherent with other sentences?
{*yes* (2) | *partially* (1) | *no* (0)}

Q2. Does the sentence match the topic in the first and the last sentence?
{*Same or close related topic* (2) | *related or no topic* (1) | *unrelated topic* (0)}

Q3. Does the sentence cover economic aspects?
{*yes* (2) | *partially* (1) | *no* (0)}

Q4. Does the sentence cover legality-related aspects?
{*yes* (2) | *partially* (1) | *no* (0)}

Q5. Does the sentence cover policy-related aspects?
{*yes* (2) | *partially* (1) | *no* (0)}

Q6. Does the sentence cover crime-related aspects?
{*yes* (2) | *partially* (1) | *no* (0)}

The first two questions asked for coherence and topic consistency, respectively. The latter four assessed the reframing effect. For the computation of the framing scores presented below, only the question asking for the target frame was taken into account. Since a sentence may serve multiple frames, the four framing questions were asked individually. We believe this scoring method is better than only asking whether a text has a desired frame, to avoid making the question suggestive. Along with this questionnaire, the definition of the four frames was provided.

	(a) w/ Entities			(b) w/o Entities	
Approach	**Rouge-1**	**Rouge-2**	**Rouge-L**	**Rouge-1**	**Rouge-L**
$\mathcal{S}_\emptyset$	16.37	2.90	13.48	13.51	11.22
$\mathcal{S}_\mathrm{F}$	16.02	2.61	13.00	14.37	11.84
$\mathcal{S}_\mathrm{N}$	27.06	10.44	23.83	14.91	12.62
$\mathcal{S}_\mathrm{A}$	9.78	0.61	8.13	9.68	8.20
$\mathcal{S}_\mathrm{FN}$	**29.70**	**12.32**	**26.27**	**16.42**	**13.97**
$\mathcal{S}_\mathrm{FA}$	11.47	0.62	9.30	11.48	9.38
$\mathcal{S}_\mathrm{NA}$	24.54	9.25	21.72	12.04	10.22
$\mathcal{S}_\mathrm{FNA}$	25.83	10.36	23.01	12.58	10.64
GPT-2	11.97	1.14	9.80	10.66	8.96
GPT-2 + $\mathcal{S}_\mathrm{F}$	12.06	1.16	9.85	10.74	9.00

Table 4.11: Rouge-1, Rouge-2, and Rouge-L F_1-scores (a) with and (b) without considering named entities of all model variations (based on our strategies $\mathcal{S}_\mathrm{F}$, $\mathcal{S}_\mathrm{N}$, and $\mathcal{S}_\mathrm{A}$) compared to the GPT-2 baselines. Rouge-2 is ignored for (b), since entity removal makes it unreliable. The highest score in each column is marked bold.

4.2.6 Results and Discussion

This subsection discusses the automatic and manual evaluation results, to then analyze how our three training strategies affect generation. Finally, we show some examples from the reframed output and discuss the limitations of our approach.

Automatic Evaluation We here use ROUGE to assess the similarity between the generated text and the ground-truth text. As some model variations use named entities extracted from the ground truth, we also consider a ROUGE variation where named-entity matches are ignored in the computation.

Table 4.11 shows the results. We see that the GPT-2 baselines perform worse than most model variations in all ROUGE scores. Adding the *framed-language pretraining* strategy ($\mathcal{S}_\mathrm{F}$) improves GPT-2 to some extent, though. The other two strategies cannot be applied directly to GPT-2. When using either strategy in isolation, only *named-entity preservation* ($\mathcal{S}_\mathrm{N}$) improves the ROUGE scores over $\mathcal{S}_\emptyset$. Even though $\mathcal{S}_\mathrm{N}$ learns to reuse the named entities from the ground-truth texts, we also see some improvement for ROUGE without named entity overlaps. Using only *adversarial learning* ($\mathcal{S}_\mathrm{A}$) decreases the ROUGE scores the most. This matches our expectation that $\mathcal{S}_\mathrm{A}$ harms coherence.

Economic (e)	**Legality** (l)	**Policy** (p)	**Crime** (c)
tobacco	court	gun	death
said	said	said	said
gun	state	bill	gun
would	marriage	would	police
state	death	state	murder
million	law	marriage	year
new	sex	law	penalty
industry	supreme	house	law
year	judge	ban	state
smoking	same	new	two

Table 4.12: The top-10 words having the highest TF-IDF values for each of the four frame in $F = \{e, l, p, c\}$.

Among the strategy combinations, $\mathcal{S}_{\text{FN}}$ has the highest ROUGE score both with and without named entity overlaps. This suggests that $\mathcal{S}_{\text{F}}$ and $\mathcal{S}_{\text{N}}$ are important to generate texts of good quality. By contrast, $\mathcal{S}_{\text{A}}$ tends to decrease the ROUGE scores also here, for example, comparing $\mathcal{S}_{\text{F}}$ with $\mathcal{S}_{\text{FA}}$. Note, however, that ROUGE tells us little about the correct framing.

Framing Word Overlaps Table 4.12 lists the top-10 framing words in each frame. Some words are characteristic for more than one frame, such as "gun" (*Economic* and *Crime*). Via manual inspection, we found that the economic frame covers the gun-sailing market while the crime frame tackles gun-control issues. The frames also have distinctive words, such as "industry" (*Economic*), "judge" (*Legality*), "bill" (*Policy*), and "police" (*Crime*).

Table 4.13 shows the proportions of framing words used in the test set, before and after reframing. It becomes clear that the variations including *adversarial learning* ($\mathcal{S}_{\text{A}}$) increase the number of framing words the most. GPT-2 models generated even fewer framing words in each frame.

Intra-Frame Generation We first look at those generated sentences $s_{2,f}$ where the target frame f is the frame used in the ground-truth, s_2. Intra-frame generation can be seen as easier for a reframing model, since some frame information may be leaked in the previous or the next sentences.

The left block of Table 4.14 shows the results. GPT-2 + $\mathcal{S}_{\text{F}}$ is worst in almost every case. In terms of keeping the topic consistent, the best approach is $\mathcal{S}_{\emptyset}$. For coherence scores, however, *B.Coherence* ($\mathcal{S}_{\text{FN}}$) obtains the highest averaged coherence score (1.71), as expected from the pilot study. Similarly, the best one for framing (1.65) is *B.Framing* ($\mathcal{S}_{\text{A}}$). The high consistency be-

Approach	Economic	Legality	Policy	Crime
$\mathcal{S}_\emptyset$	10% (−2)	12% (−1)	12% (−1)	11% (−2)
$\mathcal{S}_\mathrm{F}$	11% (−1)	13% (+0)	12% (+0)	11% (+0)
$\mathcal{S}_\mathrm{N}$	11% (−1)	13% (+0)	12% (+0)	12% (−1)
$\mathcal{S}_\mathrm{A}$	15% (+2)	20% (+6)	12% (+0)	15% (+1)
$\mathcal{S}_\mathrm{FN}$	11% (−1)	13% (+0)	12% (+0)	12% (−1)
$\mathcal{S}_\mathrm{FA}$	17% (+4)	17% (+3)	18% (+5)	13% (+0)
$\mathcal{S}_\mathrm{NA}$	13% (+0)	18% (+4)	16% (+3)	15% (+2)
$\mathcal{S}_\mathrm{FNA}$	12% (+0)	19% (+5)	16% (+3)	17% (+4)
GPT-2	8% (−4)	10% (−3)	10% (−2)	9% (−3)
GPT-2 + $\mathcal{S}_\mathrm{F}$	9% (−3)	10% (−3)	10% (−2)	9% (−3)

Table 4.13: Proportion of word overlaps between the reframed texts and the top-100 TF-IDF words of all four frames for each model variation and the GPT-2 baselines. The numbers in parentheses show the difference to the texts before reframing (in percentage points).

	Intra-Frame				Inter-Frame			
	topic	coh.	fram.	avg	topic	coh.	fram.	avg
B.Coherence	1.63	**1.71**	1.59	**1.64**	**1.64**	**1.68**	1.60	**1.64**
B.Framing	1.59	1.65	**1.65**	1.63	1.58	1.61	**1.64**	1.61
B.Balance	1.57	1.61	1.62	1.60	1.56	1.63	1.62	1.60
GPT-2 + $\mathcal{S}_\mathrm{F}$	1.54	1.61	1.57	1.57	1.55	1.59	1.58	1.57
$\mathcal{S}_\emptyset$	**1.66**	1.66	1.61	1.64	1.63	1.66	1.60	1.63

Table 4.14: Manual evaluation: The *topic* consistency, *coh*erence, *fram*ing, and average scores (*avg*) in intra- and inter-frame generation for the model varations with highest coherence ($\mathcal{S}_\mathrm{FN}$), framing ($\mathcal{S}_\mathrm{A}$), and balanced ($\mathcal{S}_\mathrm{NA}$) scores in the pilot study, compared to baselines. The best score in each column is marked bold.

tween the pilot study judges and the crowdsourcing workers speaks for the reliability of the results. With an average score of 1.64, *B.Coherence*, is, with a tiny margin, the best among all approaches in intra-frame generation.

Inter-Frame Generation Inter-frame generation requires an actual *re*framing. Its results are shown in the right block of Table 4.14. Similar to intra-frame generation, the most coherent sentences were generated by *B.Coherence* (1.68), which is also best for topic consistency (1.64) this time, slightly outperforming $\mathcal{S}_\emptyset$. Overall, the best model in the inter-frame generation is *B.Coherence* again. *B.Balance* ($\mathcal{S}_\mathrm{FN}$) is the third-best in co-

	Coherence of $s_{2,f}$					**Framing of $s_{2,f}$**				
s_2	e	l	p	c	**avg**	e	l	p	c	**avg**
e	–	1.71	**1.79**	1.69	1.73	–	1.65	1.59	1.59	1.61
l	1.71	–	1.63	1.67	1.67	**1.75**	–	1.55	1.56	1.62
p	1.67	1.68	–	1.68	1.68	1.63	1.68	–	1.61	1.64
c	*1.57*	1.67	1.65	–	1.63	1.53	*1.48*	1.56	–	1.52
avg	1.65	1.69	1.69	1.68	1.68	1.64	1.60	1.57	1.59	1.60

Table 4.15: Manual evaluation: The average coherence and framing scores of reframing from s_2 to $s_{2,f}$ for each pair of source fram (rows) and target frame (columns) from $\{e, l, p, c\}$. The highest/lowest score of each dimension is marked bold/italic.

herence and the second-best in framing, but due to its comparably low topic-consistency score (1.56), it is the worst variation on average.

Taken together, the tiny but important difference between the intra- and inter-frame generations lies in the fact that $\mathcal{S}_\emptyset$ performs better in the intra-frame generation than in the other. This suggests that, while the baselines are useful in easier cases, in the actual reframing task our proposed strategies are still needed. Besides, we observe that the inter-frame generation scores are just slightly lower than those in intra-frame generation. Considering that reframing is notably more complicated than generating the same frame, we conclude that our model realizes our reframing goals well in principle. Altogether, the rather high scores suggest that the neural generation models perform strongly in general—or that our crowdworkers were not critical enough.

To get further insights in Table 4.15, we take a closer look at the different reframing directions (source frame to target frame), focusing on the best overall model in Table 4.14, *B.Coherence*. We find that it seems rather difficult to change crime-framed sentences (source c) to other frames, especially changing it to *Economic* (e). This observation may be explained by the low word overlap between *Crime* and other frames. On the contrary, changing the *Policy* frame (p) to others seems to work better on average. When discussing policies in context, it may be easier for models to add side effects regarding economics or crime, while this is not the case for other source frames.

Training Strategies

Original text
All that was true of interracial marriages shortly after World War II. **Today, interracial marriage has strong public support, and no successful politician or prominent public figure favors outlawing such unions.** The question is whether gay marriage is on the same trajectory or is so fundamentally different that it will never be legalized.
(**b**) **GPT-2 +** $\mathcal{S}_{\text{F}}$ (reframed text)
All that was true of interracial marriages shortly after World War II. **Actually, a number of people within a decade may have wanted to apply for visas. The "Hispanic** The question is whether gay marriage is on the same trajectory or is so fundamentally different that it will never be legalized.
(**c**) $\mathcal{S}_{\emptyset}$ (reframed text)
All that was true of interracial marriages shortly after World War II. **Bob Greene: Gay marriage is on the same trajectory as interracial marriage. He says it is so fundamentally different that it will never be legalized.** The question is whether gay marriage is on the same trajectory or is so fundamentally different that it will never be legalized.
(**d**) **B.Coherence** (reframed text)
All that was true of interracial marriages shortly after World War II. **Today, same-sex marriages are legal in many states, but they are not as legal as interracial marriages, according to a new study.** The question is whether gay marriage is on the same trajectory or is so fundamentally different that it will never be legalized.
(**e**) **B.Framing** (reframed text)
All that was true of interracial marriages shortly after World War II. **"It's a good thing that we're able to do this," said a spokesman for the tobacco industry, who is suing the tobacco companies.** The question is whether gay marriage is on the same trajectory or is so fundamentally different that it will never be legalized.
(**f**) **B.Balance** (reframed text)
All that was true of interracial marriages shortly after World War II. **Today's tight labor market dictates that employers consider workers based on the skills they possess rather than the partners they prefer. Gay couples must also consider the financial obligations they owe their employers, he says.** The question is whether gay marriage is on the same trajectory or is so fundamentally different that it will never be legalized.

Table 4.16: (a) Sample text from the media frames corpus Card et al. (2015). The bold sentence is labeled with the *policy* frame. (b-f) Reframed sentences with the five manual labeled approaches to the *economic* frame.

Framed-Language Pretraining ($\mathcal{S}_{\text{F}}$) Comparing GPT-2 and GPT-2 + $\mathcal{S}_{\text{F}}$ in Table 4.11, we observe that using $\mathcal{S}_{\text{F}}$ can slightly improve the text quality

in terms of ROUGE scores. However, the benefits of this training strategy are more obvious when combining it with $\mathcal{S}_{\mathrm{N}}$. For example, $\mathcal{S}_{\mathrm{FN}}$ has about two percentage ROUGE higher compared to $\mathcal{S}_{\mathrm{F}}$.

Named-Entity Preservation ($\mathcal{S}_{\mathrm{N}}$) To generate a coherent and topic-consistent text, preserving named entities turns out to be very important. In terms of ROUGE, strategy $\mathcal{S}_{\mathrm{N}}$ is the most powerful feature. On the other hand, the model achieving the highest topic and coherence score according to the crowdsourcing results in Table 4.14 (B.Coherence) also uses this strategy, together with $\mathcal{S}_{\mathrm{F}}$.

Adversarial Learning ($\mathcal{S}_{\mathrm{A}}$) In terms of neither automatic nor manual evaluation, applying adversarial learning gives no improvement to the text quality. However, including it can generate better-framed text: In both the pilot study and the crowdsourcing study, including adversarial learning resulted in the highest framing scores.

Examples Table 4.16 exemplifies the effect of sentence-level reframing, showing how the five manually evaluated models reframed a text from the policy to the economic frame. In this particular example, the intuitively little connection between the topic of gay marriage and the frame of economic makes the reframing task particularly challenging.

As the table shows, only two models successfully managed to change the focus, *B.Framing* and *B.Balance*. In particular, the result of the former mentions an opinion of "a spokesman for the tobacco industry", the latter uses the labor market's viewpoint. However, the text "It's a good thing that we're able to do this" in *B.Framing* appears to be rather vague and general. Besides, the text is related to economy only because it mentions the tobacco industry. On the other hand, *B.Balance* integrates gay marriage and economic more naturally by using the labor market to connect the two concepts.

4.3 Statement Bias Mitigation

In this section, we introduce our model to mitigate the statement bias of a given text. Specifically, given a piece of text containing left-oriented (or right-oriented) bias, our goal is to rewrite the text with opposite bias while keeping the same semantic meaning as much as possible. Relying on the news bias corpus created in Section §3.1, we extract the texts that can be used in our statement bias transfer task.

		Same Event (Q3)			
		Same	**Changed**	**Not Sure**	**All**
Bias (Q4)	**Changed**	57	1	0	58
	Same	28	1	0	29
	Not Sure	10	1	2	13
	All	95	3	2	100

Table 4.17: Counts of all possible combinations in the manual evaluation of whether the ground-truth headlines capture the same event with flipped bias.

4.3.1 Ground-truth Mitigation Instances

We took all 2196 opposite headline pairs (*left-oriented, right-oriented*), where both headlines of a pair are about the same event. We randomly selected 100 pairs as the validation set, another 100 pairs as the test set, and the remaining as the training set. To verify the test set, from Upwork we hired three experts in journalism editing to annotate all 100 test pairs. For each pair, the annotators had to answer four questions:

Q1. Do you understand headline 1?
{*yes* | *partially yes* | *no* | *not sure*}

Q2. Do you understand headline 2?
{*yes* | *partially yes* | *no* | *not sure*}

Q3. Do both headlines report on the same event?
{*same* | *mostly same* | *changed* | *not sure*}

Q4. Do the headlines have the opposite bias?
{*changed* | *partially changed* | *same* | *not sure*}

The resulting Fleiss'κ values were 0.97 (Q1), 0.97 (Q2), 0.62 (Q3), and 0.30 (Q4). All annotators understood almost all headlines, except for one with only two words: "Lerner speaks". The agreement for Q3 was substantial and fair for Q4. Majority voting was used for the final decision.

Table 4.17 shows the annotations of Q3 and Q4, combining *same* and *mostly same* for Q3, and *changed* and *partially changed* for Q4. From the 100 pairs, 95 were labeled as being on the same event, while only five pairs confused the annotators. For the bias label, 58 headline pairs have opposite bias, while the rest did not show any clear difference.

4.3.2 Statement Bias Mitigation Model

From the annotated headline pairs, we observed that not all headlines show bias. To enrich bias information in the training set, we added the content of

each article, split into sentences. We use these sentences as supplemental information during learning. Since we do not have a transferred version of each sentence in the content, we do not use the content for the validation and test set, and we evaluate the results only based on the headlines. Knowing that two sentences in a training pair may have different semantics, we need a model that learns to transfer bias, but at the same time infers the semantics of a sentence.

Formally, given a source sentence s_o along with its bias label b_o and its content z_o, during training, our goal is to generate the target sentence s_t with label b_t and content z_t, while z_o and z_t could be different. We are interested in transfer the bias from b_o to b_t and from b_t to b_o, so we train two encoders $E(s_k, b_k)$, $k \in \{o, t\}$, that learn to infer z_k:

$$z_k \sim E(s_k, b_k) \tag{4.4}$$

Analogously, we train two generators G to generate s_k given b_k and z_k:

$$\hat{s_k} \sim G(z_k, b_k) = p(s_k|b_k, z_k) \tag{4.5}$$

Given the parameters in E and G, θ_E and θ_G, the two autoencoders (one transfer from source to target, the other from target to source) is then optimized to minimize the reconstruction error from s_k to $\hat{s_k}$:

$$\mathcal{L}_{rec}(\theta_E, \theta_G) = \mathbb{E}_{s_k \sim S_k}[-\log p(s_k|s_k, E(s_{\bar{k}}, b_{\bar{k}}))],$$

where $\bar{k}$ is o when k is s, and $\bar{k}$ is s when k is o.

As in other generative approaches, we also learn to maximize the loss of the adversarial discriminator as follows:

$$\mathcal{L}_{adv} = -\log D_k(s_k)] - \mathbb{E}[\log -D_k(\hat{s_{\bar{k}}}))], \tag{4.6}$$

where D_k is the discriminator used to distinguish s_k from the transferred version $s_{\bar{k}}$.

Finally, the loss function aims to minimize the loss from reconstruction and the adversarial discriminators from two directions:

$$\mathcal{L}_{rec_{o\to t}} + \mathcal{L}_{rec_{t\to o}} - (\mathcal{L}_{adv_{o\to t}} + \mathcal{L}_{adv_{t\to o}}),$$

where $o \to t$ means changing from source to target and $t \to o$ from target to source. To train the model, the architecture of Shen et al. (2017) fits our needs. We thus replicate their cross-alignment setting: During training, we choose the same number of left and right sentences randomly and then train

the autoencoder from two directions in one batch. Even though the pairing information is saved by this architecture, the results are promising: Modifying the sentiment while maintaining semantics worked correctly in 41.5% of all cases.

Besides, generative models are known to often produce *UNK* (the out-of-vocabulary word), which is especially harmful in understanding the meaning of short sentences, as given in our task. To reduce the frequency of *UNK* in the generated outputs, we set the size of the beam search to 10, and keep the candidates with the fewest *UNK*.

4.3.3 Baselines

Besides the model we propose in the paper, we also experimented with other approaches that generate a text given another text. Specifically, we tried (1) training our model only with headline pairs, (2) the pointer generator (See et al., 2017) trained only with headline pairs, and (3) the sentiment and style transfer from Li et al. (2018). The pointer generator originally focused on abstractive summarization where it achieved high Rouge scores. It learns to copy words from the source to handle out-of-vocabulary issues. The sentiment and style transfer focuses on detecting the attribute (the sentiment words for instance), and trying to alter it by looking for the best candidates in a corpus.

However, even when finetuning their parameters, neither of these approaches generated readable outputs. Mostly, they just repeated words or phrases, such as "the the the" or "trump he same he for trump". So, without sufficient content in the training data, it seems hard to obtain a language model that generates meaningful sentences.

In particular, the pointer generator requires paired training samples, hence training with sentences from the content is not possible. The sentiment and style transfer does not require paired training samples, but its attribute detection mechanism requires an unequal distribution of sentiment words. From the experiment in bias analysis, we know that this assumption does not hold in our corpus. The model described in the approach section is an end-to-end model without any strong assumptions. Although it has a higher amount of parameters, it can produce more readable sentences.

4.3.4 Evaluation

To evaluate the results automatically, we measured the similarity between the generated and the ground-truth headlines via Rouge-1, Rouge-2, and Rouge-L, resulting in F-scores of 15, 3, and 12. In an additional manual

		Same Event (Q3)			
		Same	**Changed**	**Not Sure**	**All**
Bias (Q4)	**Flipped**	83	17	4	104
	Same	21	10	0	31
	Not Sure	23	33	9	65
	All	127	60	13	200

Table 4.18: Counts of all combinations in the manual evaluation of the generated compared to the ground-truth headlines in terms of event and bias.

Ground-truth headline pair		**Generated versions of the headlines**		**Evaluation**	
Headline	Bias	Headline	Bias	Event	Bias
John McCain urges republicans not to filibuster gun control.	left	*John McCain has elected to avoid gun control.*	right	same	changed
White House looks to salvage gun-control legislation.	right	*White House got to get bipartisan change.*	neutral	mostly same	partially changed
Obama accepts nomination, says his plan leads to a "better place".	left	*Obama blasted re-election, saying it a "very difficult" to go down.*	right	mostly same	changed
Lackluster Obama: change is hard, give me more time.	right	*Real GOP: debate is right, and more Trump.*	left	changed	changed

Table 4.19: Two left-right headline pairs, along with the rewritten versions generated by our approach. The bias of the ground-truth headlines is given in our corpus. The bias of the generated headlines is from the human annotators.

evaluation, another three editing experts answered Q2 to Q4 by comparing the original and generated headlines, with a Fleiss' κ of 0.61 (Q2), 0.51 (Q3), and 0.29 (Q4). Out of 200 generated headlines (100 left-to-right, 100 right-to-left), 73 were seen as understandable (Q2), which we see as a good result for a generative model. For Q3 and Q4, Table 4.18 details the results. For those headlines, where the content was kept (127), the bias was changed in 83 cases (65%). Even for those with changed meanings, 28% got the opposite bias.

Table 4.19 shows selected pairs of ground-truth and generated headlines. They demonstrate that our model keeps the event similar by using the same words, and changes bias by replacing or adding biased words. The generated headlines contain some grammar errors, but we see these as tolerable in machine-generated text on limited data.

In the first pair, the original headline states that McCain was pro-gun control, while the rewritten one implies he was against it — a successful

change. The ground-truth bias-changed headline in the second row mostly uses other words while being pro-gun control. The generated headline also keeps most words but turns out rather neutral. In the second pair, the original headline shows a positive opinion of Obama, and the generated headline is a negative opinion of him. When rewriting the ground-truth bias-changed headline (last row), the meaning is not kept. However, it is visible that the generated headlines are pro-Trump.

We point out that there is a difference between bias-changing and fact-changing. For example in the first pair, without knowing what John McCain stood for, we could neither guess his real opinion on gun control nor could we conclude what he supported or not. In fact, bias can be conveyed by emphasizing facts supporting a claim, as well as by hiding facts attacking a claim. In other words, we might see different facts about the same event with different types of bias. A news headline may be a conclusion, while the news content shows the facts supporting this conclusion. In such cases, no computational model will be able to change the content only using the text itself, as it is hardly possible to simply generate new facts. Including more articles reporting on the same event will be useful to help the model learn the unseen information. We see this as future work on article-level bias changing.

Finally, we found that an automatic evaluation of bias changing is limited. In the discussed examples, we see that even for a successful change, the overlapping of generated and ground-truth headlines is very low. The successful cases have a mean Rouge-1 score of 17, and unsuccessful ones of 15. Furthermore, if we divide the test pairs into those labeled as *same event* and *changed bias* (57 pairs) and the rest (43), we find that the former is more often rewritten successfully (43% vs. 20%). This suggests that filtering out noisy cases with the help of experts will help improve performance.

4.4 Summary

In this chapter we have demonstrated how to mitigate three kinds of media bias: *gatekeeping bias*, *coverage bias*, and *statement bias* using computational models. To the best of our knowledge, we are the first to attempt to study these three kinds of bias mitigation problems.

In Section 4.1, we propose to insert new information in order to mitigate the gatekeeping bias. Using a multi-task learning approach, we generate sentence that contains new information, keeps the topic, and is coherent with its context. Our method's ability to use any new information as input is one of its main advantages. At the same time, using a variable set of

classifiers provides an adaptable, flexible mechanism to fulfill the desired conditions. We will see how to extend the method here into another task in the chapter later.

Later in Section 4.2 we have introduced a reframing task for the coverage bias mitigation. We have cast it as a sentence-level fill-in-the-blank task. It involves generating new sentences with target frames while keeping their coherence and topic consistency with the surrounding context. To tackle the task, we have suggested three training strategies to control the framing and coherence of the generated sentences. While evaluating these strategies automatically and manually, we found that a combination of the techniques results in a successful reframing with acceptable coherence and topic consistency, even though no single strategy can satisfy the needs of reframing. Even though we are aware of the limitations of our approach, we contend that such sentence-level reframing is a big step towards full article reframing.

Lastly, based on the corpus created in Section 3.1, in Section 4.3 we have studied how to rewrite a news headline from right-oriented to left-oriented, and vice versa. The rewritten text helps to mitigate the statement bias. We have trained a neural network model based on the cross-alignment architecture of Shen et al. (2017). Our experiment results suggest that current state-of-the-art approaches struggle with this task. Even though our best-tested model did quite well, there is still much potential for development.

In this chapter, we have learned the mitigation strategies for addressing the three types of media bias. Notably, we have seen the role that computational models play in mitigating bias. In transition to the next chapter, our focus shifts towards an examination of the robustness of the developed models. This will provide insights into the reliability of the computational approaches employed in analyzing and mitigating media bias, offering a deeper understanding of their practical applications and limitations.

5
Beyond Textual Media Bias

> I may walk slowly but I never walk backward.
>
> Abraham Lincoln

In the previous chapters, we have learned how to analyze and mitigate different kinds of media biases. In this chapter, we would like to investigate if we can apply the learned knowledge to other domains of NLP tasks. Specifically, we are interested in two tasks: content transfer and biased snippet generation. For content transfer, it is an NLP task focusing on changing the content of the text while keeping others unchanged. We chose this task because it is very close to the gatekeeping bias mitigation task (details in the Section later). Biased snippet generation is the task of how to generate biased snippets in a web search scenario. Though the task itself is far away from media bias, we see the potential of applying media bias knowledge because we can also cast the snippet generation as a type of gatekeeping bias mitigation.

In the following, Section §5.1 studies how to cast content transfer as a gatekeeping bias mitigation task based on the paper (Chen et al., -). We use the model from the gatekeeping bias mitigation and apply it to the dataset of content transfer. In Section §5.2, we first perform a theoretical user study to see the potential usage of generated snippets (Chen et al., 2018a). The results suggest that the generated snippets perform as well as snippets from the Google search engine. Afterward in Section §5.3, we develop a snippet generation model to generate snippets. The presentation is based on our published paper Chen et al. (2020c).

Topic. Mattress
Reference I really would expect a better mattress than a rock.
Text. Two complaints though: the bed, although not uncomfortable, was a little lumpy. **The mattresses felt like they were made of cement.** And the free WiFi never actually worked for me.

Table 5.1: Examples of content transfer on a hotel review. The bold sentences are generated by the approach proposed in this work, given a *topic*, a *reference*, and the surrounding *text* as input. The generated sentence about mattresses can be entailed from the reference that the mattresses was like a rock.

5.1 Content Transfer

Text style transfer aims to change the style of the text while retaining its content. Driven by recent developments in neural network architectures, there has been much progress in text style transfer (Fu et al., 2018). However, limited attention has yet been paid to the opposite direction: retaining style while changing content, referred to as *content transfer* (Qin et al., 2019; Prabhumoye et al., 2019), such as completing a story following the same style.

The main goal of content transfer is to insert a new sentence into the text. Table 5.1 exemplifies the three conditions to be fulfilled, as proposed in other work on content transfer (Qin et al., 2019; Prabhumoye et al., 2019): *content transfer* with respect to the reference, *topic adherence* with respect to the discussed topic, and *coherence* with respect to the context.

Compared with the three requirements (information containing, topic keeping, and coherence) in our gatekeeping bias mitigation task (see Section §4.1), we found that the tasks of content transfer and gatekeeping bias mitigation are very close to each other. Therefore, we decided to reuse the multi-task learning model as in Section §4.1, with the following twisting: (1) We have a reference sentence as the new content to be inserted. (2) Topic is given a word. And (2) context is given as the surrounding sentences as in gatekeeping bias mitigation.

5.1.1 Experiments

We report our experiments in content transfer in this subsection. Especially, we focus on how to use the gatekeeping bias mitigation model to perform the content transfer task.

Datasets We consider hotel reviews in the experiments. Compared with the news articles in the gatekeeping bias mitigation scenario, the reviews

Training	Validation	Test
23,471	5,158	6,489

Table 5.2: The number of instances in training set, validation set, and test set for the dataset.

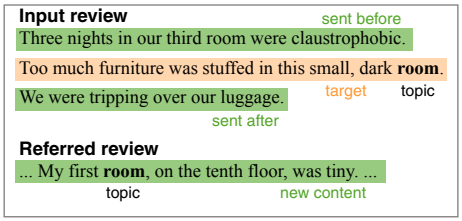


Figure 5.1: A training instance from the hotel reviews. Both the *target* and the *new content* have negative sentiments toward the *topic*, "room". In this figure, the inputs ($sent_{before}$, $sent_{after}$, and *new content*) are in green and the output (*target*) is in orange.

have shorter sentences and are more homogeneous since their topic-related scope is limited. By contrast, in Section §4.1) news articles are longer and more diverse, likely making them more challenging. The distribution of the dataset can be seen in Table 5.2.

We use the corpus of Wachsmuth et al. (2014), which contains a balanced set of hotel reviews from TripAdvisor with 300 reviews each for seven hotel locations, 60 each per star rating from 1 to 5. We use the hotels located in Amsterdam, Seattle, Sydney, and Berlin as the training set, hotels located in San Francisco as the validation set, and hotels located in Barcelona and Paris as the test set. The numbers in the parentheses indicate the number of hotels in the city. In each review, the sentiment of each statement is classified as positive, negative, or neutral. We focus on the sentences where a topic (so-called *product feature* in the paper), and positive or negative sentiments are annotated, for example, *room* in Figure 5.1(a) with negative sentiment.

Classifiers and Model Following the gatekeeping bias mitigation task, we train three classifiers outside the generation model, and the base model is still `facebook/bart-base` from the Huggingface library (Wolf et al., 2019). The base model selections are the same as in Section §4.1 and we retrain all the models.

Content Transferred Classifier Given a reference, r, and generated text, $\hat{y}$, this classifier predicts the probability of being transferred. Given a sentence having a sentiment toward a topic, we took all sentences from other reviews with the same hotel, holding the same sentiment. Similarly, for non-

Training			Validation			Test		
Cont.	Neu.	Non-cont.	Cont.	Neu.	Non-cont.	Cont.	Neu.	Non-cont.
23,471	34,478	16,925	5,158	7,225	3,861	6,489	9,205	4,890

Table 5.3: The number of instances in the training, validation, and test set for the content transferred classifier training. We randomly selected neutral (Neu.) and non-contain transferred (Non-cont.) samples to have equal number as contain transferred (Cont.) ones as much as possible.

Training		Validation		Test	
Positive	Negative	Positive	Negative	Positive	Negative
6931	6931	1445	1445	1841	1841

Table 5.4: The number of instances in the training, validation, and test set for the topic adherence and the coherence classifiers. The positive label means topic keeping or coherence, respectively; the negative label means no topic adherence or incoherence.

contain transferred, we selected sentences from other reviews for the same hotel about the same topic but with opposite sentiments. Neutral, they are sentences from other reviews mentioning a different topic. Table 5.3 shows the distribution of the labels. The macro-average F_1-score of the classifier is 0.86, and the F_1-score for the content transferred label.

Topic Adherence and Coherence Classifiers These two classifiers are the same as in Section §4.1. The data distribution can be seen in Table 5.4. The accuracies are 0.98 for topic adherence and 0.83 for coherence.

Baselines As baselines for our approach, we also select the two models as in Section §4.1: our previous paper (Chen et al., 2021) and the error correction model by Thorne and Vlachos (2021).

5.1.2 Results and Discussion

This section discusses the automatic and manual evaluation results of our approach and the baselines. We selected an example to analyze the model qualitatively and discuss the hyperparameters of the model.

Automatic Evaluation We first evaluate the generated texts and the fulfillment of the input conditions using ROGUE F_1-scores and available automatic metrics:

Approach	Rouge-1	Rouge-2	Rouge-L
Chen et al. (2021)	30.88	11.90	27.02
Error correction	**31.04**	**12.06**	**27.18**
Our approach	30.96	11.97	27.05

Table 5.5: Rouge-{1, 2, L} F_1-scores of the two baselines and our approach on the hotel reviews and news articles. The best score in each column is marked bold.

Approach	Content transferred↑	Topic adherence↑	Coherence↓
Chen et al. (2021)	**.651**	**.968**	40.65
Error correction	.643	.956	40.22
Our approach	.647	.965	**38.41**
w/o selection	.647	.956	39.96

Table 5.6: Automatic evaluation: Proportion of texts fulfilling the content transferred, topic adherence and coherence conditions. *w/o selection* denotes the proportion before candidate selection. The best score per column is marked bold.

ROUGE F_1-Scores Table 5.5 shows that the *error correction* model does best but the performance from all the approaches are very close to each other.

Condition Fulfillment We also consider the following automatic metrics to evaluate the condition fulfillment: BERTScore (Zhang et al., 2020) with its best model `deberta-xlarge-mnli` from Microsoft for successfulness of being content transferred; Vanilla `bart-large-mnli` model as the zero-shot topic classifier; And GPT-2 (Radford et al., 2019) for coherence.

Table 5.6 shows a similar trend as in Section §4.1: *our approach* has the lowest perplexity while it has the second-best performance in content transferred and topic adherence. Still, the two baselines and the two variations of our approach have significant differences among each other using automatic evaluations.

Manual Evaluation We follow the question set in Section §4.1 for our manual evaluation. We also randomly selected 100 instances from the dataset and we showed the participants the sentence before and after, the topic, and the new content to be added. The questions are:

Q1. What is the relationship between the sentence and the new content?
{ *The sentence entails the new content* (*1*)
| *The new content entails the sentence* (*1*)
| *The sentence partly entails the reference*(*0.5*)

Approach	Content transferred↑	Topic adherence↑	Coherence↓
Chen et al. (2021)	0.49	0.88	0.96
Error correction	0.53	0.83	0.98
Our approach	**0.54**	**0.89**	**0.99**

Table 5.7: Manual evaluation (main results): Mean scores of content transferred, topic adherence, and coherence on each dataset. The best score in each column is marked bold. The †symbols denoting a significance ($p < 0.05$) comparing to the second best.

| *The new content partly entails the sentence* (*0.5*)
| *Neither entailment nor contradiction* (*0*)
| *They partly contradict each other* (*-0.5*)
| *They contradict each other* (*-1*) }

Q2. Does the sentence cover the given topic?
{*yes* (*1*) | *partially* (*0.5*) | *no* (*0*) }

Q3. Is the sentence coherent with the others?
{*yes* (*1*) | *partially* (*0.5*) | *no* (*0*) }

The numbers in parentheses indicate the scores of each answer.

Also at a cost of about $0.33 per instance, we used Amazon Mechanical Turk to obtain four annotations per instance. We required the workers to have at least a 95% approval rate and at least 10,000 authorized HITs in order to obtain more trustworthy findings. Similarly, we applied the multi-annotator competence estimation to estimate the final label.

Table 5.7 presents the manual evaluation results. For *content tranferred*, our approach has the best score compared to the other two baselines by a small margin. Additionally, our method continues to be the best for topic adherence, while error correction is the poorest. Lastly, all models satisfy the coherence condition, with the exception of Chen et al. (2021), which performs somewhat worse.

Interestingly, we note that all models work much better on news articles. A reason may lie in the complexity of assessing whether information in news articles is right or wrong. In contrast, we assume that the higher topic adherence in the hotel reviews is due to the different text layouts. In news articles, the topics are usually more invisible. For example, the text in Figure 4.2b mentioned Nancy Pelosi with the pronoun "her". Such a writing style makes it more difficult to generate topic-relevant texts.

Qualitative Analysis We examine an example of generated text from one of the three methods in Table 5.8 as an illustration. The model was asked

Topic. view
Reference We arrived to a friendly welcome and a very comfortable King Premier room, which I had requested at the front of the hotel so that there was a *view of the city and waterfront*.
Chen et al. (2021). Fantastic Hotel, Brilliant staff, let down by NOISE. **And the view was spectacular.** I would have given this hotel a 4 or 5 rating.
Error correction. Fantastic Hotel, Brilliant staff, let down by NOISE. **Great location, great view of the Eiffel Tower.** I would have given this hotel a 4 or 5 rating.
Our approach. Fantastic Hotel, Brilliant staff, let down by NOISE. **I have stayed at this hotel before and had a fantastic view of the city from my room.** I would have given this hotel a 4 or 5 rating.

Table 5.8: Sample from a hotel review Wachsmuth et al. (2014). The bold sentences are generated by the baselines and our approach, given topic and reference fact.

to generate a sentence regarding the topic "view", given a fact and the surrounding sentences. We see that the model of Chen et al. (2021) generates rather generic text. While the error correction model output mentions the great view, the hotel is actually in Barcelona, not in Paris, so the statement regarding the Eiffel Tower cannot be true. In contrast, our model correctly states that the hotel has a city view which can be inferred from the reference fact.

Hyperparameters For the weights of each condition, we found the best combination is 0.4 for generation loss and 0.2 for all conditions (which are 0.7 for generation loss and 0.1 for all conditions in news articles). The higher generation loss for news articles suggests that it is harder to generate news texts, so the models have to learn more from the generation loss.

5.2 User Study of Snippets

Snippets are an essential part of a search results page: they incite users to view (to click) or to skip viewing a retrieved document. Already in 1991, Pedersen et al. (1991) proposed query-biased snippets, and they have proven useful until today (Tombros and Sanderson, 1998; White et al., 2002a,b).

These snippets are generated by reusing phrases and sentences from the web page that contain all or at least some of the query's terms. In summarization terminology, this is called *extractive summarization*. The alternative *abstractive summarization* relaxes the reuse constraint by allowing for abstracting the web page's content by paraphrasing, generalization, or sim-

plification. In this section, we first study the feasibility of query-biased abstractive snippet generation.

5.2.1 User Study Design

Within three crowdsourcing tasks, we first acquired paraphrases to generate abstract snippets, and then experimentally determined which kind workers prefer when given a pair of abstract and extractive snippets, and which kind is more useful to spot relevant results on a search results page.

Crowdsourcing Abstractive Snippets Given the insight from Bando et al. (2010) of the high overlap of human paraphrase snippets to machine-generated extractive snippets, paraphrase snippets represent a sufficient substitute for extractive snippets for our user study. To maximize the diversity of the set of pairs of reuse snippets and corresponding paraphrase snippets, we resort to crowdsourcing. Using the 150 topics provided for the TREC Web tracks 2009–2011, each topic's query has been submitted to Google's custom search API[1] to obtain high-quality search results. The top-5 search results of each query were collected, including title, URL, and Google's reuse snippet for a total of 750 snippets. Since Google's snippet generator sometimes shortens sentences to enforce a maximum snippet length (indicated by ellipses), we recovered the complete sentences from the linked pages. For crowdsourcing we relied on Amazon's Mechanical Turk (AMT), where we offered the task of manually paraphrasing the reuse snippets collected. The worker's instructions were to significantly rewrite a given snippet while maintaining its length and without removing important named entities, phrases, or quotes (e.g., "to be or not to be"). To foreclose easy cheating, copy and paste was disabled in the AMT interface. Each of the 750 reuse snippets was assigned to two different workers for paraphrasing (i.e., we repeated our experiment twice in a row to test its reliability). Submitted paraphrases were reviewed, rejecting those lacking changes or poor grammar, resulting in 1,500 pairs of reuse and paraphrase snippets.

Snippet Preference To assess snippet preference, we recruit 5 workers for each pair of extractive /paraphrase snippets to judge which of the two they would prefer for the given query and web page. The task interface showed instructions, a search box with the topic's query, the pairs of snippets side by side, formatted like standard search results, the associated web page in a frame below, and a text field to enter an assessment justification. Workers

[1] https://developers.google.com/custom-search/

were asked to assign one of four labels: snippet 1 or 2 is better, both are good, or both are bad. To avoid order bias, the positions of extractive and paraphrase snippets were randomized. After collecting all judgments, we tallied the scores as follows: an extractive or paraphrase snippet gets 1 point if a worker judged it to be better, both get a point if a worker judged that both are good, and neither gets a point otherwise.

The worker pool of AMT has been known to comprise dishonest workers, threatening the reliability of our study. We took several precautions: each worker judged at most two pairs of snippets ensuring diversity, and submissions were rejected if workers spent insufficient time, too much time, or if they failed to provide sensible explanations for their judgments, resulting in 4,235 individual workers and 7,500 accepted annotations. Only workers having at least an 80% acceptance rate and at least 100 successful assignments were invited. Furthermore, we conducted control experiments with respect to the variables snippet source, preference bias, snippet length, and random pairings to check how workers are affected.

Snippet Usefulness To obtain implicit feedback on a snippet's usefulness for spotting relevant search results, another group of workers judged the relevance of a search result to a query given different page configurations. The queries, corresponding web pages, and relevance scores were obtained from the topics used at the TREC Web tracks 2013–2014, which were based on the ClueWeb12. For each topic, we tried to collect 3 web pages judged as relevant and 3 judged as irrelevant that are still available on the live web and whose contents correspond to that found in the ClueWeb12.[2] For 29 topics, we were able to collect the desired set of 6 web pages. Following the aforementioned procedures, we collected reuse snippets using Google's custom search API and paraphrased them via crowdsourcing.

Workers were then exposed to search results pages comprising 3 results (1) with extractive snippets, (2) with paraphrase snippets, (3) without snippets (only titles and URLs), (4) with extractive snippets only (no titles or URLs), or (5) with captcha-style snippets to ensure workers read the snippets. In the latter case, the snippets just stated whether a result was supposed to be relevant or irrelevant. A search results page could contain 0 up to 3 relevant web pages. For mixtures of relevant and irrelevant pages, we tested all permutations of search result orderings. For each order, three workers provided labels, yielding a total of 10,440 annotations (29 topics, 8 relevance settings (0–3 results relevant), 5 snippet conditions (extractive, paraphrase, etc.), 3 results per search results page, and 3 annotators each).

[2]Topics from the previous TREC Web tracks were omitted since they are based on the ClueWeb09 which is insufficiently represented on today's web.

Each worker judged search results pages of 5 different topics based on the given information. To ensure annotation quality, we rejected results from workers who did not pass the captcha snippets, resulting in 546 individual workers in this experiment.

5.2.2 User Study Analysis

We conducted a careful statistical analysis of the crowdsourced snippet judgments. The snippet preference experiment rests on the hypothesis that users do not *consciously* care whether snippets are extractive or paraphrased from linked web pages as long as they are semantically equivalent. If true, there should be no statistically significant difference in terms of user preference. The snippet usefulness experiment rests on the hypothesis that users are not *unconsciously* negatively affected by paraphrase snippets. If true, users identify relevant web pages either way and there should be no statistically significant differences between the two kinds of snippets.

Descriptive Statistics The reuse snippets collected comprise an average of 1.9 sentences and 41.1 words; the longest snippet has 6 sentences and 122 words, and the shortest one is 1 sentence and 14 words. The paraphrase snippets comprise an average of 2.2 sentences and 40.5 words; the longest snippet has 6 sentences and 132 words, and the shortest one is 1 sentence and 9 words. The paraphrase snippets are significantly longer at the sentence level ($p < 0.05$), but neither are significantly shorter nor longer at the word level ($p > 0.05$). A reason might be that workers tended to split long sentences while paraphrasing. The workers spent an average 220.5 seconds to paraphrase a snippet at a maximum of 895 seconds (38 words) and a minimum of 14 seconds (also 38 words), an average of 49 seconds to judge a pair of snippets, and of 17 seconds to judge a web page's relevance when viewing a search results page. The inter-annotator agreement Fleiss κ for the snippet preference experiment, presuming the four labels to be independent, was 0.37, indicating a fair agreement, and 0.77 for the snippet usefulness experiment, indicating a substantial agreement.

Snippet Preference *Judgment distribution* Table 5.9 shows the distribution of judgments. Recall that workers were unaware which snippet was which; their judgments were mapped to the ground truth afterward. The amounts of judgments for *Extractive better* and *Paraphrase better* are roughly equal and about a quarter of workers had no preference (*Both good* plus *Both bad*). Only

Assessment	Judgments	
	absolute	relative
Reuse better	2,731	36.41%
Paraphrase better	2,652	35.36%
Both good	1,537	20.49%
Both bad	580	7.74%
Total	7,500	100.00%

Table 5.9: Distribution of judgments; 1,500 pairs of (reuse, paraphrase) snippets at 5 assessors each yield 7,500 judgments.

Experiment	Reuse	Paraphrase	p-value
all	3.06	2.97	0.51
Wikipedia	3.31	2.58	**0.00**
Non-Wikipedia	2.75	2.85	0.31
all	3.05	2.94	0.43
Wikipedia	3.18	2.64	**0.01**
Non-Wikipedia	2.77	2.82	0.58

Table 5.10: Average scores (number of votes) of reuse and paraphrase snippets with p values for a paired t-test, bold font indicating significance ($p < 0.05$); the two row groups correspond to two repetitions of the experiment.

580 pairs of snippets (7.74%) were judged *Both bad*, showing a high overall snippet quality.

Extractive snippets vs. paraphrase snippets

To check for snippet preferences, we performed a paired t-test on the pairs of reuse and paraphrase snippets. Since we repeated our experiment, collecting two different paraphrase snippets for each of the 750 reuse snippets, we can attest that our results can be replicated under the same conditions: the rows *all* in Table 5.10 show the results for the two repetitions. While the absolute average scores (number of votes) achieved by paraphrase snippets are slightly smaller than those of extractive snippets, no statistically significant difference was measured, given pretty high p values of 0.51 and 0.43, respectively.

Wikipedia snippets vs. non-Wikipedia snippets

From the 750 search results, 260 refer to Wikipedia articles. Considering only this subset, the rows *Wikipedia* in Table 5.10 show that users significantly prefer extractive snippets over paraphrases, which is not the case for non-Wikipedia results. The effect sizes under Cohen's d are small to

Experiment (par. = paraphrase, ex. = extractive)	(•,)	(,•)	p-**value**
(better, worse)	3.91	1.71	**0.00**
((better-par., worse-ex.), (better-ex., worse-par.))	2.85	2.78	0.41
(long, short)	2.91	2.70	**0.01**
((long-par., short-ex.), (long-ex., short-par.))	2.82	2.81	0.90
(better, worse)	3.89	1.75	**0.00**
((better-par., worse-ex.), (better-ex., worse-par.))	2.87	2.78	0.23
(long, short)	2.99	2.61	**0.00**
((long-par., short-ex.), (long-ex., short-par.))	2.79	2.83	0.60

Table 5.11: Average scores of pairs of snippets grouped by different aspects, with associated p values and one row group per experiment repetition.

medium (0.51 and 0.31, respectively). Upon review of Wikipedia snippet pairs, many of the extractive snippets have an apriori high writing quality, and it may have been difficult for the average AMT worker to compete with that.

Preferred snippets vs. unpreferred snippets To quantify the difference between snippets preferred by users (better) to those not preferred (worse), we reordered the snippet pairs accordingly, disregarding ties, and then applied a paired t-test. The rows (*better, worse*) in Table 5.11 show the results for each repetition of our experiment. As can be seen, there is an average 2.2 score difference between them, rendering the differences significant. However, when comparing the groups of snippet pairs (better-reuse, worse-paraphrase) with (better-paraphrase, worse-reuse), p-values of 0.41 and 0.23 indicate that snippet preference is independent of whether they are extractive or paraphrased.

Long snippets vs. short snippets We further investigated if snippet length affects preference (rows (*long, short*) of Table 5.11. A snippet belongs to the "long" snippets if it is the longer one of a pair, and to "short" snippets otherwise. On average, the long snippets have 44.7 words and the short snippets have 36.7 words. Our findings corroborate those of Maxwell et al. (2017), namely that users prefer longer snippets. Many of the assessment justifications from our workers support this finding. Again, when comparing the groups of snippet pairs (long-paraphrase, short-extractive) with (long-extractive, short-paraphrase), p-values of 0.90 and 0.60 indicate that the dimensions length and reuse are independent.

Extractive snippets vs. unrelated snippets As a control experiment to ascertain worker diligence, pairs of extractive snippets and unrelated snippets were shown to workers, where a given extractive snippet was paired with

	Reuse	Paraphrase	No snippet	Snippet only	Random
F-score	67.64	64.61	63.65	60.16	50.00

Table 5.12: F-scores of the snippet usefulness experiment indicating whether annotators correctly spot relevant search results under different search results page snippet conditions.

a random extractive snippet of a different web page of a different query. Of 1,500 judgments collected, workers preferred the snippet matching the query in 85% of the cases, confirming this experiment's setup validity.

Snippet Usefulness This experiment questioned whether users can identify relevant pages given different kinds of snippets—one of the key tasks snippets should support. A search result is labeled as relevant if more than half of the workers label it as relevant, and irrelevant otherwise. In Table 5.12 we show the F-score of the crowdsourced judgments based on snippets compared with the ground truth relevance labels obtained from the TREC assessors. The numbers of overall shown relevant and irrelevant web pages were balanced, such that random guessing yields a baseline F-score of 50%. In the captcha-style setting where the snippets just explicitly state that a web page is relevant / irrelevant, the workers achieved an F-score of 100% (since we excluded those who did not succeed in these check instances). As for the other snippet conditions, we find that although extractive snippets achieve the highest F-score (helping users best to judge a result's relevance), the performance of paraphrase snippets is not significantly worse ($p = 0.28$). Showing only extractive snippets (without titles or URLs) achieves the lowest F-score; no snippets (only title and URL) is better than showing only snippets, confirming that title and URL do play an important role. All settings are significantly better than random guessing. Otherwise, only reuse snippets are significantly better than showing only snippets ($p < 0.05$). The remaining pairings are not significantly different, corroborating that paraphrase snippets are no worse than extractive snippets.

We conclude that the combination of snippet, title, and URL is crucial to identify relevant web pages on search results pages, regardless of whether snippets are extractive or paraphrased. Nevertheless, there is room for improvement given the results obtained from showing the "perfect" snippet, which reveals the relevance of a web page (our captcha setup). The finding that both extractive and paraphrase snippets are useful supports our claim

that paraphrase snippets can replace extractive snippets in future information systems.

Reproducibility User studies need to be reproduced to determine whether the results of previous studies on a given problem of interest generalize and that they were not due to unidentified confounding variables or accidental flaws in the study setup. This pertains particularly to first-time studies since only an independent reproduction will provide sufficient confidence that the results obtained are valid and that they may generalize. Since our study is the first of its kind that provides an answer to the question of whether extracting text for snippet generation is a necessity, or whether paraphrased snippets are sufficient, we expect that sooner or later it will have to be reproduced for its results to be corroborated. The reproducibility of a user study rests on a clear description of its setup, which we tried our best to provide. But maybe even more so it rests with access to data, code, and supplementary material that was gathered throughout the study. To ensure the reproducibility of our results, we provide all data collected and the associated code open source.[3]

5.3 Abstractive Snippet Generation

In the previous section, we have shown that extractive snippets and paraphrased snippets are comparable with each other. In this section, we study how to generate paraphrased snippets automatically–abstractive snippets generation. In the following, we discuss how to acquire such abstractive snippets corpus. We also discuss how to cast the generation problem into a task that can be solved using the idea of gatekeeping bias mitigation.

5.3.1 Abstractive Snippet Corpus

For the construction of our snippet corpus, Webis Abstractive Snippet Corpus 2020[4], we considered anchor contexts as the source of collecting ground-truth abstract snippets. Our mining pipeline creates the corpus automatically from scratch, given a web archive as input and we assessed the quality of the corpus via crowdsourcing.

An anchor context is a text surrounding the anchor text of a hyperlink on a web page (see Table 5.13). Ideally, it explains what can be found on the linked web page, e.g., by summarizing its contents. The author of an anchor context personally describes the linked web page, enabling readers to decide whether to visit it or not, just like snippets on search results pages.

[3] https://github.com/webis-de/SIGIR-18

[4] Corpus: https://webis.de/data.html#webis-snippet-20

Query: Treasury of Humor
Snippet: anchor context Asimov, on the other hand, proposes (in his first jokebook, Treasury of Humor) that the essence of humour is anticlimax: an abrupt change in point of view, in which trivial matters are suddenly elevated in importance above those that would normally be far more important.
Document [. . .] Treasury of Humor is unique in that in addition to being a working joke book, it is a treatise on the theory of humor, propounding Asimov's theory that the essence of humor is an abrupt, jarring change in emphasis and/or point of view, moving from the crucial to the trivial, and/or from the sublime to the ridiculous [. . .]

Table 5.13: Example of an anchor context as training snippet. The anchor text that linked to the document is highlighted.

To identify useful anchor contexts that are fluent, meaningful, and close to this ideal, we employ a multi-step mining process. Table 5.14 overviews corresponding mining statistics.

Crawling Raw Anchor Contexts We mine anchor contexts from the ClueWeb09 and the ClueWeb12 web crawls,[5] focusing on their 1.2 billion English web pages (500 million from the ClueWeb09 and 700 million from the ClueWeb12). For every hyperlink, we extract its anchor text and 1500 characters before and after as anchor context, trading off the comprehensiveness and size of the resulting data. The extracted raw 18 billion and 13 billion anchor contexts, respectively, have been fed into the following nine-step pipeline.

Step 1: Intra-site links We assume that anchor contexts of cross-site links are more likely genuine pointers to important additional information compared to intra-site links: The vast majority of the latter are found in menus, footers, buttons, and images, entirely lacking plain text context. We discard all anchor contexts of intra-site links by matching the second-level domain names of the web page containing a given context with that of the linked page. More than 96% of the raw anchor contexts are thus removed in this step.

Step 2: Non-existing pages We discard anchor contexts that link to pages that are not available in the ClueWeb collections; most of them are dead

[5] See `https://lemurproject.org/clueweb09/` and `https://lemurproject.org/clueweb12/`

Mining pipeline	ClueWeb09		ClueWeb12	
	Remaining	Δ	Remaining	Δ
Raw anchor contexts	17,977,415,779		12,949,907,331	
1. Intra-site links	440,605,425	-97.6%	514,337,093	-96.0%
2. Non-existing pages	111,082,494	-74.8%	91,007,214	-82.3%
3. Non-English pages	107,819,314	-2.9%	91,007,214	-0.0%
4. Spam anchors	24,767,468	-77.0%	19,829,007	-78.2%
5. Stop anchors	17,188,286	-30.6%	15,837,168	-20.1%
6. Improper text	9,631,489	-44.0%	9,248,806	-41.6%
7. Duplicated	6,292,317	-34.7%	5,403,893	-41.6%
8. Text reuse	6,183,783	-1.7%	5,349,610	-1.0%
9. Short web pages	5,651,649	-8.6%	5,114,479	-4.4%
Unique pages:	2,499,776	–	1,557,330	–

Table 5.14: Statistics of the anchor context mining pipeline.

links on the live web. This pertains to 75% and 82% of the remaining anchor contexts.

Step 3: Non-English pages All anchor contexts whose (linked) page is non-English are discarded. We rely on the language identification done for ClueWeb09 encoded in its document IDs, whereas ClueWeb12 is advertised as an English-only collection.

Step 4: Spam anchors The Waterloo spam ranking provides spam scores for the ClueWeb09 and the ClueWeb12 (Cormack et al., 2011). As suggested, we remove anchor contexts whose (linked) pages' spam rank is $< 70\%$. However, we make an exception for anchor contexts whose linked pages have a relevance judgment from one of the TREC Web tracks (2009-2014), Session tracks (2010-2014), or Tasks tracks (2015, 2016).

Step 5: Stop anchors Anchor contexts whose anchor text is empty, or contains the words "click", "read", or "mail" are removed, since they led our models astray. We also remove multi-link anchor contexts to avoid ambiguous contexts not related to an individual link. As a heuristic, we require a minimum distance of 50 characters between two anchor texts, removing all others.

Step 6: Improper text To remove anchor contexts with improper text, we only keep those where (1) the anchor text has at most 10 words (in pilot studies, longer anchor texts were hardly informative or resulted from HTML parsing errors), (2) the anchor text is part of a longer text of at least 50 words (longer texts are a key indicator of meaningful and readable texts), (3) the sentence containing the anchor text has at least 10 words (longer sentences more often resulted in meaningful anchor contexts), (4) the anchor

context contains at least one verb as per the Stanford POS tagger (Toutanova et al., 2003), and (5) the anchor context has a stop word ratio between 10% and 70% as per Biber et al. (1999)'s (Biber et al., 1999) study of written English.

Step 7: Duplicated anchor contexts To avoid any training bias resulting from duplication, we remove duplicate anchor contexts linking to the same page from different pages. To quickly process all the pairs of anchor contexts for each page, we use locality-sensitive hashing (LSH) (Rao and Zhu, 2016). We first encoded all anchor contexts as 128-dimensional binary vectors based on word unigrams, bigrams, and trigrams and then removed one of the anchor contexts as "duplicate" to another if the cosine similarity of their vectors was larger than 0.9 (this value was determined in pilot studies). Another 34-42% of the anchor contexts were removed as duplicates.

Step 8: Text reuse Since our goal is *abstractive* snippet generation, we exclude all anchor contexts that are purely *extractive*. This was done by checking if the anchor context was completely copied from their respective linked pages. Partial reuse, i.e., due to reordering of phrases, however, has been retained as a mild form of abstraction.

Step 9: Short web pages Finally, we removed anchor contexts whose linked web pages contained less than 100 words to ensure a sufficient basis for summarization. Arguably, snippets need to be generated for shorter pages, too. However, we envision different, specialized snippet generators for different length classes of web pages, which we leave for future work.

Altogether, we obtained 10,766,128 ⟨anchor context, web document⟩ tuples from the two ClueWeb collections referring to 4,057,106 unique pages. The average length of an extracted anchor context is 190 words (longest: 728; shortest: 50). The average linked page is 841 words long (longest: 14,339; shortest: 100) and it has 2.65 anchor contexts, while about two-thirds (2,675,980 pages) have only one, and the most often linked page has 12,925 anchor contexts.

Query Generation The final step of constructing our corpus was query generation for the ⟨anchor context, web document⟩ tuples. While these tuples can already be used as ground truth for generic abstractive summarization (which we do as part of our experiments), they are still unsuited to train a *query-biased* abstractive snippet generation model for lack of a query. To be suitable training examples, we require for every tuple a query for which (1) the document is (at least marginally) relevant, and (2) the abstractive snippet surrogate is (at least marginally) semantically related.

One might consider the anchor text (i.e., the hyperlinked text) to be a suitable candidate for a query that sufficiently fulfills these constraints. A

cursory analysis, however, suggested that the texts the authors of the anchor contexts chose to include in their links are not necessarily well-suited for our purposes, even when excluding stop anchors containing words like 'click' and 'here' as per Step 5 of our anchor context mining pipeline. To avoid this potential for bias, we instead resorted to keyphrase extraction from the entire anchor context to generate queries. Regarding the web directory descriptions, this was the only alternative, anyway.

First, for each tuple, we parse the abstractive snippet surrogate and its associated document using the Stanford POS tagger (Toutanova et al., 2003) and extract all noun phrases with a maximum length of six words. Here, we apply the more limited definition of strict noun phrases by Hagen et al. (2012), where a strict noun phrase has only adjectives, nouns, and articles. This maximizes tagging reliability and limits the types of queries we consider. Second, to ensure a strong semantic connection between anchor context and document, we use only those noun phrases as queries that appear in both the abstractive snippet surrogate as well as the document. We generate at most three queries per tuple with an average length of 2.43 words (longest: 6; shortest: 1).

The additional constraint that the extracted query has to occur in both the abstractive snippet surrogate as well as the linked document limits the usable tuples. In the end, we obtained a total of 3,589,701 ⟨query, anchor context, document⟩ triples and 55,461 ⟨query, web directory description, document⟩ triples corresponding to 33.4% and 9.7% of the respective original sets of tuples.

The constraints we impose on the shape of queries and their relation to the abstractive snippet surrogates and the document may seem extreme. However, we argue that a tight control of the texts, and their relation to the query is crucial due to the noisy web data. We leave the study of relaxing these constraints and studying other types of queries (e.g., questions) for future work.

Web Content Extraction Web pages are a notoriously noisy source of text: A naive approach to content extraction, e.g., by simply removing all HTML markup, is frequently found to be insufficient. Instead, we adopt the content extraction proposed by Kiesel et al. (2017). Here, the web page is first "rendered" into a plain text format, so that blocks of text can be discerned. Then, noisy text fragments, such as menus, image captions, etc., are heuristically removed. This includes paragraphs with fewer than 400 characters, sentences with less than 50% letter-only tokens, and sentences without an English function word.

5.3.2 Abstractive Snippet Generation and Gatekeeping Bias Mitigation

Our main approach to generate abstractive snippets used our corpus of ⟨query, snippet, document⟩ triples for training. In fact, generating the snippets given a document and a query is actually rewriting the document biased to the query and shortening the document. To recap, in gatekeeping bias mitigation, our goal is to insert a new text into a given context that is biased to the topic and fluent. Therefore, the snippet generation task is similar to the task in gatekeeping bias mitigation while no context is given, and the query here can be seen as a topic in the gatekeeping bias mitigation task.

In this study, we adapt pointer-generator networks for snippet generation. In addition to comparing two variants of our model, we also consider four baselines, including one state-of-the-art abstractive summarization model, and one extractive summarizer with a paraphraser attached.

Pointer-generator networks with copy mechanism (See et al., 2017) are one of the most successful neural models used for sequence-to-sequence tasks, such as summarization and machine translation. They tackle two key problems of basic sequence-to-sequence models: reproducing facts and handling out-of-vocabulary words. At each time step, the decoder of this model computes a generation probability that decides whether to generate the next word or whether to copy a word from the to-be-summarized text. This provides a balance between extractive and abstractive aspects during the generation process, since, even in the abstractive summarization scenario, reusing some phrases/words is still an essential feature to preserve information from the source text that cannot be abstracted. For instance, named entities should not be exchanged. We note that this aligns with our aim of generating factually correct abstractive snippets with limited text reuse. For our experiments, we use the PyTorch implementation provided by OpenNMT (Klein et al., 2017).

Although pointer-generator networks have been shown to generate abstractive summaries reasonably well, they are not designed for generating query-biased abstractive snippets, nor to *explicitly* insert designated terms into a generated summary; the expectation for snippets on search engines is to at least include some of the query's terms or synonyms thereof. We, therefore, devise a pointer-generator network that guarantees by construction that the query occurs in an abstractive snippet.

Our *bidirectional generation model* generates an abstractive snippet using two decoders as depicted in Figure 5.2. We set up the query words to be the first words in their output and then the model learns to complete the

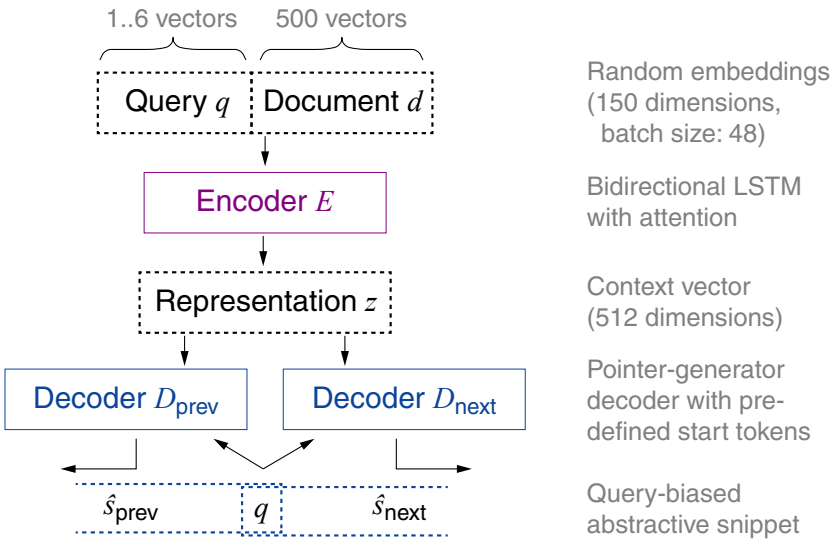


Figure 5.2: Diagram of our snippet generation model that depicts a sequence-to-sequence setup with two decoders, generating to the beginning and the end of a snippet starting with the query terms. The figure is reused from our paper (Chen et al., 2020c)

snippet in two directions: starting from the query to the end of the snippet and starting from the query to its beginning. Formally, given a query q, a document d, and the target snippet $s = \langle s_{\text{prev}}, q, s_{\text{next}} \rangle$, where s_{prev} (s_{next}) is the snippet before (after) the query q, our model trains an encoder E to encode the concatenation of query q and document d to a context vector $z \sim E(q, d)$. Furthermore, two decoders D_{prev} and D_{next} generate the snippet from vector z: one from the query's terms to the beginning of the snippet $\hat{s}_{\text{prev}} \sim D_{\text{prev}}(z, q)$; the other from the query's terms to the end of the snippet $\hat{s}_{\text{next}} \sim D_{\text{next}}(z, q)$. The final generated snippet is the concatenation of $\hat{s}_{\text{prev}}$, q, and $\hat{s}_{\text{next}}$.

5.3.3 Input Preparation

Our model is trained with the ⟨query, snippet, document⟩ triples from our corpus. Given the fact that the query's terms can occur anywhere in the document (even multiple times), and given the wide range of document lengths versus the limited input size of our model, we resort to an initial extractive summarization step as a means of length normalization while ensuring that the document's most query-related pieces of text are encoded in the first place. This, however, may negatively affect fluency across sentences. We use the query-biased extractive snippet generator by Liu et al. (2018), which computes query-dependent TF-IDF weights to rank

a document's sentences, and then selects the top 10 as input, truncating at 500 words.

5.3.4 Model Variants and Baselines

We train two variants of our model, one using anchor context triples (`AnCont.-QB`), and another one using DMOZ triples (`DMOZ-QB`). These models generate query-biased snippets as described above.

Furthermore, we consider four baselines. The first two baselines employ our model as well, but without enforcing query bias: we train one model on the anchor context triples (`AnCont.`) and another on the DMOZ triples (`DMOZ`) without using the query as predefined output. Thus, we can simplify our model by using just a single decoder. This way, there is a chance for the models to learn to generate query-biased snippets by themselves, however, for each generated snippet, there is also a chance that they will end up not being query-biased.

Another two baselines allow for a comparison with the state of the art in abstractive summarization and paraphrasing. The former is a single layer Bi-LSTM model trained on the CNN/Daily Mail corpus for abstractive summarization (`CNN-DM`).[6] This model implements a standard summarizer and will therefore not generate query-biased summaries. The paraphrasing baseline (`Paraphr.`) is a way of combining conventional extractive snippet generators with paraphrasing technology to achieve the same goal of producing snippets that are abstractive and that do not reuse text. This baseline operationalizes the manual creation of paraphrased snippets as in our previous user study (Chen et al., 2018a), albeit with a lower text quality as paraphrasing models are still not perfectly mature. We first create a query-biased extractive summary of the web page consisting of three sentences (Liu et al., 2018), and then apply the pre-trained paraphrasing model of Wieting et al. (2017).[7]

5.3.5 Evaluations

We carried out both an intrinsic and an extrinsic evaluation of the generated snippets for all model variants and the baselines. While the intrinsic evaluation examines the language quality of the generated snippets, the extrinsic evaluation assesses their usefulness in the context of being used on a search engine. For both intrinsic and extrinsic evaluation, we carry out

[6] `http://opennmt.net/Models-py/`

[7] `https://github.com/vsuthichai/paraphraser`

Model	Training	Validation	Test
DMOZ-QB	54,461	1,000	3,894
AnchorContext-QB	3,581,965	3,842	3,894
DMOZ	573,720	1,000	3,894
AnchorContext	10,758,392	3,842	3,894

Table 5.15: The number of instances for training, validation, and testing.

Model	ROUGE-1	ROUGE-2	ROUGE-L
CNN-DM	20.5	4.2	15.4
Paraphraser	14.3	1.1	10.5
DMOZ	15.6	1.7	11.4
DMOZ-QB	20.8	2.8	15.6
AnchorContext	23.0	5.0	18.0
AnchorContext-QB	**25.7**	**5.2**	**20.1**

Table 5.16: Evaluation results: Snippet overlap with the ground truth.

crowdsourcing experiments employing master workers from Amazon Mechanical Turk with a minimum approval rate of 95% and at least 5000 accepted HITs (Human Intelligence Task).[8]

Table 5.15 shows our corpus split into training, validation, and test sets. Note that the two baseline models `CNN-DM` and `Paraphraser` can be trained on the entirety of snippet-document pairs extracted from the ClueWeb collections and DMOZ, regardless of whether queries can be generated for them. From both the DMOZ descriptions and the anchor contexts, we randomly selected 1000 documents for validation. This resulted in 1000 DMOZ triples because of the one-to-one correspondence between DMOZ descriptions and linked documents, but 3842 anchor context triples. Likewise, for testing, we selected another 1000 documents each, resulting in 3894 anchor context triples.

5.3.6 Intrinsic Evaluation

We computed ROUGE F-scores for all the models as seen in Table 5.16. The ROUGE scores are computed by comparing the n-gram overlap between the snippets generated by each model and the reference snippets in the test set. Among the two baselines `CNN-DM` and `Paraphraser`, `CNN-DM` achieves higher ROUGE scores across all granularities. Because `CNN-DM` was trained to summarize articles while `Paraphraser` only paraphrases the first three sentences

[8] We paid an average hourly rate of $5.60 and a total amount of $622.50.

Model	Fluency	Factuality	Reuse
CNN-DM	2.42	**76.10**	83.17
Paraphraser	422.20	45.21	68.61
DMOZ	2.59	2.02	33.75
DMOZ-QB	1215.18	6.89	**29.55**
AnchorContext	**2.04**	6.19	30.37
AnchorContext-QB	2.31	17.89	45.36

Table 5.17: Evaluation results: Snippet quality: fluency is measured as perplexity (lower is better), factuality as strict noun phrase overlap (higher is better), and reuse as ROUGE-L precision (lower is better).

from an extractive summary, it is unsurprising that snippets generated by `CNN-DM` fit the gold standard better. Also, we observe that inducing query bias by reshaping the corpus leads to better ROUGE scores for both DMOZ descriptions and anchor contexts. The `AnchorContext-QB` model has the best performance among all variations. It shows that the model can successfully generate snippets close to the gold standard with help from the bidirectional architecture. However, notwithstanding the smaller training corpus of 54,461 instances, we also see the effectiveness of this approach in the `DMOZ-QB` model.

Next, we assessed fluency, factuality, and text reuse of the snippets concerning the to-be-summarized document. We selected 100 documents from the test set and corresponding snippets generated by each model. The size of the test set is reduced as we also employ manual evaluation alongside automatic evaluation. Also, we ask the human judges to annotate only high-quality examples that were chosen using their perplexity scores as mentioned below to showcase the potential of abstractive snippet generation.

Fluency

Calculating fluency automatically can be done using a language model where the perplexity score implies how fluent (probable) a text is, with lower perplexity for more fluent texts. We used a publicly available BERT model (Devlin et al., 2019) to compute perplexity scores for the snippets.[9] These perplexity scores were also used to select the test set for manual evaluation.

The average perplexity scores of our models and the four baselines are shown in the first column of Table 5.17. `CNN-DM` and `AnchorContext` generate fluent texts with low perplexities. While `CNN-DM` is trained on well-written news articles with extractive summaries (Grusky et al., 2018), the high per-

[9] We used *BERT-Large, Uncased* for our experiments.

Model	Agreement		Score				Avg.
	Maj.	Full	-2	-1	1	2	
CNN-DM	86%	28%	0	1	24	75	**1.73**
Paraphraser	78%	22%	7	17	40	36	0.81
DMOZ	81%	17%	2	4	43	51	1.37
DMOZ-QB	97%	51%	**51**	**41**	7	1	-1.34
AnchorContext	90%	32%	0	3	20	**77**	1.71
AnchorContext-QB	73%	10%	3	35	**44**	18	0.39

Table 5.18: Evaluation results: Snippet fluency.

formance (low perplexity) of the `AnchorContext` model can be attributed to the relatively large corpus of 10 million training examples. However, in the case of `AnchorContext-QB`, just the addition of query bias in the snippet generation process introduces breaks in the text flow, thereby introducing a small increase in the perplexities. `DMOZ`'s perplexity is similar to `AnchorContext` or `AnchorContext-QB`, showing that the DMOZ descriptions' language fluency is pretty high. In the case of `DMOZ-QB`, the poor performance can be attributed to a strong repetition of tokens (sometimes more than 10 times) in the generated snippets. Besides, we also see a pretty high perplexity for `Paraphraser`, which implies that simply rephrasing words without considering the whole context largely reduces the fluency of texts.

In addition to automatically computing perplexity scores, we performed a manual evaluation where we asked workers to score the fluency on a 4-point Likert scale from *very bad* via *bad* and *good* to *very good* corresponding to scores from -2 to 2. The workers were only presented with the generated snippets in the HIT interface. Table 5.18 shows the results of this qualitative evaluation. We achieved a high inter-annotator agreement with the lowest majority agreement of 73% and the highest one of 97%. Among all models, `CNN-DM` achieves the highest average fluency score, with `AnchorContext` performing competitively. `DMOZ` also achieves a rather high average score in fluency. Given the comparably smaller number of DMOZ descriptions available for training than that of anchor contexts, such a reduction of fluency can be expected from the generated snippets. Besides, the scores of query-biased model (`DMOZ-QB` and `AnchorContext-QB`) are significantly lower than the scores of query-unbiased models (`DMOZ` and `AnchorContext`). This shows that when our architecture generated snippets with the requirement to explicitly put the query words in the snippets, the model compromised the language fluency in order to meet this requirement.

Factuality This dimension is similar to the information-containing condition in the gatekeeping bias mitigation task. Cao et al. (2018) showed that neural text generation models can create fluent texts despite conveying *wrong* facts. One reason is that factual units, such as names, locations, or dates, are infrequent in corpora, which leads to their weak representation in the final embedding space. The proposed copy mechanism (Gu et al., 2016), and pointer generator networks (See et al., 2017) mitigate this problem to some extent. For example, in summarization, models with the copy mechanism learn to reuse some words/phrases from the documents to be summarized and simply copy them to the generated summary.

However, ROUGE cannot be used to evaluate factuality as it does not specifically count factual units while computing the n-gram overlap. Thus, we formulate the factuality evaluation as calculating the ratio of strict noun phrases preserved by the generated snippet for a given document: $|S \cap \hat{S}|/|\hat{S}|$, where S is the set of strict noun phrases in a document, and $\hat{S}$ is the set of strict noun phrases in its generated snippet. Recall that a strict noun phrase is defined as a noun phrase with a head noun and an optional adjective, which have also been exclusively considered for query generation. This ratio approximates the number of factual units from the document that are preserved by the generated snippet.

The factuality scores can be found in the second column of Table 5.17. We see that `CNN-DM` and `Paraphraser` have a much higher ratio of strict noun phrases than the other models. Manual inspection of the generated snippets reveals excessive copying of text from the document to the snippet by both models. This preserves a large number of factual units, albeit impacting the desired property of abstractiveness in the snippets. Also, this increases the amount of text reuse. The anchor contexts are relatively abstractive, which makes reproducing facts rather difficult for models trained on our corpus. However, generating query-biased snippets (`DMOZ-QB` and `AnchorContext-QB`) leads to some improvement in the factuality scores. We attribute this to concatenating the query term and the web page during the training which improves the strict noun phrase overlap.

Text reuse In addition to being fluent and factually correct, an ideal abstractive snippet also avoids text reuse from the document. We enforced this property during corpus construction by filtering out anchor contexts that largely reuse text from the web document. To evaluate the impact of this step, we calculate the amount of text reuse as ROUGE-L precision between the generated snippet (candidate) and the document (reference). A lower precision implies lower text reuse by the generated snippet. The third column of Table 5.17 shows the results of the automatic evaluation of text

Model	Agreement		Score				Avg.
	Maj.	Full	-2	-1	1	2	
CNN-DM	90%	20%	4	15	28	**53**	**1.11**
Paraphraser	81%	14%	4	22	37	37	0.81
DMOZ	70%	17%	14	**27**	37	22	0.26
DMOZ-QB	91%	40%	**38**	16	6	2	-0.82
AnchorContext	79%	19%	11	14	**38**	37	0.76
AnchorContext-QB	75%	16%	4	26	32	38	0.74

Table 5.19: Evaluation results: Summarization effectiveness.

reuse. The baselines have very high text reuse, especially `CNN-DM`, so that most of the generated snippets are sentences copied from the documents. As the training corpus for `CNN-DM` does not contain many abstractive summaries as references, this is not unexpected as the model's copy mechanism gains significantly higher importance during training. Our four model variations exhibit much lower text reuse. Except for `AnchorContext-QB`, the other three have similar text reuse rates—about one-third. This shows that our models have learned to balance reuse with abstractiveness.

5.3.7 Extrinsic Evaluation

Our extrinsic evaluation assesses how well the generated snippets can be used in practice. We designed two crowdsourcing tasks to evaluate summarization effectiveness and snippet usefulness.

Summarization Effectiveness A usable snippet should ideally describe the web document and help users make an informed decision about whether or not to click on a search result returned for the search query. In one HIT, we showed the query, a snippet generated by one of the models, and the summarized web page. Workers were asked to score how helpful the given snippet was at describing the document on a four-point Likert scale defined as follows: *Very poor* (-2): The snippet does not describe the web page and is useless. *Poor* (-1): The snippet has some information from the web page but doesn't help decide to visit the web page. *Acceptable* (1): The snippet has key information from the web page and helps decide to visit the web page. *Good* (2): The snippet describes the web page well and helps decide to visit the web page. The results of this task can be found in Table 5.19. We achieved a high inter-annotator agreement with the lowest majority agreement of 75% and the highest one of 91%. The table shows that `CNN-DM` best summarizes the documents, while `Paraphraser` and the anchor context models `AnchorContext` and `AnchorContext-QB` perform comparably

Model	Maj.		Full	
	Yes	No	Yes	No
CNN-DM	79	21	41	21
Paraphraser	73	27	23	27
DMOZ	60	**40**	10	**40**
DMOZ-QB	69	31	12	31
AnchorContext	82	18	34	18
AnchorContext-QB	**87**	13	**43**	13

Table 5.20: Evaluation results: Query bias as judged by crowd workers, where each study is based on $n = 100$ snippets, three votes each, and agreement is measured as 2/3 majorities and full agreement.

well. We see that DMOZ-QB has a much lower average score than the others. The low scores are reflected also by the low language fluency of the text and the low factuality as shown in the previous evaluations.

Additionally, we asked workers to judge if the snippet is query-biased when describing the document. This helps us further assess if our query-biased models do generate snippets that consider the search query in their description of the web document. Table 5.20 shows the full agreements among workers for this specific question. We observe that AnchorContext-QB and CNN-DM are the top two models where the snippets are query-biased (43 and 41). However, CNN-DM also has a higher number of *no* votes. Also, compared to query-unbiased models (DMOZ and AnchorContext), the query-biased models (DMOZ-QB and AnchorContext-QB) can generate snippets that are more query-focused. It follows that our process of shaping the training examples to be query-biased is successful and our models can learn to generate such snippets. Given the fact that the DMOZ descriptions are often too short to reliably shape them into query-biased training instances, the DMOZ model fails to generate a high number of query-biased snippets.

Snippet Usefulness With this crowdsourcing task, we evaluate whether the users of a search engine can identify relevant results for a given search query based on the generated snippet of each document. We follow our previously applied experimental setup (Chen et al., 2018a). First, we selected 50 topics as queries from the aforementioned TREC Web tracks, Session tracks, and Tasks tracks, ensuring that the queries had at least three relevant and three irrelevant documents judged in the datasets provided by the tracks. We evaluated each model independently by showing the search query and snippets of six documents (whose relevance judgments are known) generated by this model. The interface of this annotation task, as presented to the workers, can be seen in Figure 5.3, where workers judged each snippet to be

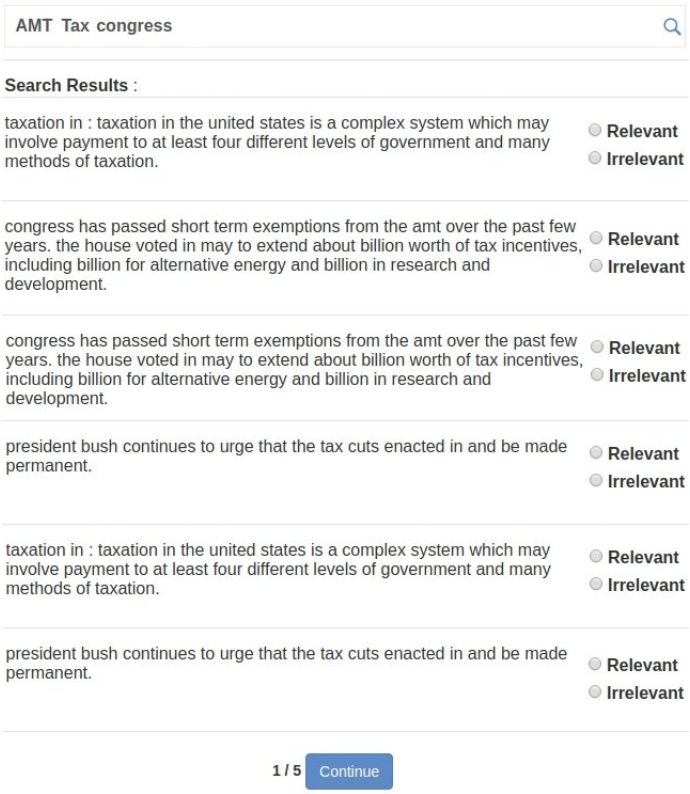


Figure 5.3: The interface used by workers in deciding snippet usefulness, showing several snippets next to radio boxes where the expected relevancy of a document on a search results page is to be predicted. The figure is reused from our paper (Chen et al., 2020c)

relevant or irrelevant. This task emulates the use of abstractive snippets in a practical setting.

Table 5.21 shows the results of this experiment. For comparison, we show the results of Chen et al. (2018a), where the extractive snippets as employed by Google achieved the highest F-score of 67.64 among their approaches. The baselines `CNN-DM` and `Paraphraser` perform similarly to each other but are still worse than Chen et al. (2018a)'s extractive snippets. Both DMOZ-based models performed worse than the baselines, while `DMOZ-QB` performs a lot better than `DMOZ`; its snippets have more query bias. The `AnchorContext-QB` model performs competitively to Chen et al. (2018a)'s extractive snippets (66.18 versus 67.64), while comprising significantly less text reuse (see last row of Table 5.17). This result is very promising, implying sufficiently abstractive snippets that are useful to identify relevant documents, like the extractive snippets of commercial search engines.

Model	F-score
CNN-DM	61.85
Paraphraser	60.49
DMOZ	46.03
DMOZ-QB	59.82
AnchorContext	34.86
AnchorContext-QB	**66.18**
Chen et al. (2018a)	**67.64**

Table 5.21: Evaluation results: Usefulness of snippets to crowd workers in selecting relevant documents, compared to our previous study (Chen et al., 2018a) of manually paraphrased snippets.

5.3.8 Examples of Generated Snippets

Table 5.22 and 5.23 show our example query "cycling tours" and an excerpt of a relevant document from the ClueWeb09 right below. Each of the models under evaluation has generated a snippet for this document, listed below the excerpt.

The `CNN-DM` model copies two sentences from the document (the second one appears later). This is unsurprising since the model's training corpus exhibits a strong extractive bias. By accident, one of the sentences contains a query term ("cycling"), rendering the snippet partially query-biased. A common problem with neural text generation is exemplified, namely the repetition of words and phrases ("Vermont"). For training and generation, numbers have been replaced with `<num>`. The `Paraphraser` model paraphrases the first sentence, accidentally removing the snippet's query bias while introducing terminology related to bicycling ("transmission"), but not related to the document's topic. It generates erroneous statements and has problems with the usage of definite and indefinite determiners. Still, some text remains untouched, so that a little less than half the snippet is reused text. These observations are in line with our evaluation results, where `Paraphraser` is found to exhibit lower fluency scores due to less reuse (Tables 5.17 and 5.18) and is less query-biased (Table 5.20) than `CNN-DM`. Our basic factuality measures do not yet include fact-checking, thus overlooking false statements.

Regarding our `DMOZ` model, if not for the repetition and a factual error at the end, it would have generated a to-the-point, query-biased snippet. Its style resembles that of DMOZ summaries, quite befitting a snippet (e.g., by omitting the subject in the second sentence). Still, this model achieves the lowest performance in summarization effectiveness (Table 5.19) and query

Query: cycling tours
Web document (excerpt; ClueWeb09 ID "clueweb09-en0020-20-15833") All bike new England routes are now available on one disk with over 200 cue sheets and maps from 7 years of touring in the Berkshires, the Massachusetts north shore, Cape Cod, Vermont, New Hampshire, [. . .] This special version of our north shore **cycling tours** is designed for active participants aged 55 and older through the Elderhostel travel organization exploring the England coast and historic Essex county [. . .] (see entire document at [ChatNoir])
CNN-DM All bike new England routes are available on one disk with over <num> cue sheets and maps from <num> years of touring in the Berkshires, the Massachusetts north shore, Cape Cod, Vermont, New Hampshire and Vermont. Rides are up to 45 miles a day, with shorter loops for all levels of **cycling** ability.
Paraphraser All the new England transmissions are now available on one disk with a blanket of <num> and maps from <num> years of travel in the northwest coast, a Massachusetts north coast, the Cape Cod, Vermont, New Hampshire [. . .]
DMOZ Guided **bike tours** in Massachusetts, Massachusetts, Rhode island, New Hampshire, Connecticut, Massachusetts and Massachusetts. Includes information on **tours**, **tours**, events, and contacts. Located in Boston.
DMOZ-QB Spring **tours cycling tours** offers a range of **tours** and services to individuals and groups in the north bay area. Includes information on **tours**, **tours**, events, and activities. Also includes a guide to the <eos>

Table 5.22: An example of a query, a relevant web document, and snippets generated by the studied models. The query's terms are highlighted bold, reuse gray, repetitions underlined gray, and errors underlined black.

bias (Table 5.20), although its measured fluency is much higher compared to that of DMOZ-QB. The latter also generates a rather to-the-point snippet, albeit also with repetitions and improper termination (<eos>). Starting with the query "cycling tours" and generating in two directions, the backward generation to the beginning of the snippet failed, introducing a false company name ("Spring tours cycling tours"). Neither of the DMOZ models reused any phrase longer than two words from the original document.

The two anchor context models provide the most fluent examples that do not reuse any text verbatim. The AnchorContext model does not repeat itself and accidentally includes at least one of the query's terms. It introduces a company not referred to in the document, and it introduces bike tours at

Query: cycling tours
Web document (excerpt; ClueWeb09 ID "clueweb09-en0020-20-15833") All bike new England routes are now available on one disk with over 200 cue sheets and maps from 7 years of touring in the Berkshires, the Massachusetts north shore, Cape Cod, Vermont, New Hampshire, [. . .] This special version of our north shore **cycling tours** is designed for active participants aged 55 and older through the Elderhostel travel organization exploring the England coast and historic Essex county [. . .] (see entire document at [ChatNoir])
AnchorContext For more information on the Deerfield river bike tour **cycling**, visit the Deerfield web site. The Deerfield is a small company offering group tour services for organizations and individuals in the north shore area of Massachusetts, southern New Hampshire and Vermont, and the popular Worcester mountains of western Massachusetts.
AnchorContext-QB Walking and **cycling tours**. The **tours** is a great place to start and enjoy the best of the great lakes in the United States and around the world, as well as some of the most beautiful and beautiful places in the world.

Table 5.23: An example of a query, a relevant web document, and snippets generated by the studied models. The query's terms are highlighted bold, reuse gray, repetitions underlined gray, and errors underlined black.

places not mentioned in the document. The `AnchorContext-QB` repeats itself and introduces "walking" tours. Also, it is very generous in its praise, making strong subjective claims unbefitting snippet language. This indicates some bias in the anchor context training data: Perhaps, more often than not, an author linking to another document has good things to say about it (e.g., when referring to the web document of a nice place one has visited). This merits further investigation and perhaps the inclusion of an additional filtering step based on sentiment analysis. Both models introduce errors relating to determiner usage.

Altogether, despite the encouraging evaluation results, our in-depth analysis of the example as well as others reveals a lot of room for improvement. All models except the mostly reusing `CNN-DM` model introduce language or factual errors to a greater or lesser extent, the factual errors being the most important issue to tackle in future work. The baseline models `CNN-DM` and `Paraphraser` disqualify themselves with respect to text reuse; they are hardly abstractive. A cause for the shortcomings of the two query-biased models `DMOZ-QB` and `AnchorContext-QB` may be the fact that they are

forced to start generating with the query, which may not be an optimal starting point, whereas query bias is important for snippet generation.

5.3.9 Model Ranking

Snippet usefulness—the capability of a model to generate snippets that enable humans to select relevant results—is the key measure to rank abstractive snippet generation models. Our `AnchorContext-QB` model performs best, achieving an F-score competitive to that of extractive snippets. Nevertheless, it does not achieve the crowdsourced effectiveness and fluency scores of `CNN-DM`, which achieves the second-highest usefulness score. The latter, however, mostly reuses text from the summarized documents: There is no practical advantage in training a neural snippet generation model that is not abstractive since state-of-the-art extractive snippet generators perform competitively with little development overhead.

The `Paraphraser` and the two DMOZ-based models are ranked third to fifth in terms of usefulness, while their ranking is reversed in terms of reuse. The `Paraphraser` and the query-biased DMOZ model have the lowest fluency among all models, while the remaining query-unbiased DMOZ model scores second to lowest in terms of usefulness. Nevertheless, the writing style of the snippets generated by the DMOZ-based models is closest to our expectation of a well-written snippet. It is conceivable, however, that by restoring the entire DMOZ directory and by retrieving archived versions of its linked pages, a substantially higher overall performance can be attained than is possible with the comparably small amount of training examples we could obtain. That size of the training data matters can be observed for the query-unbiased `AnchorContext` model, which is trained on 10 million examples and achieves the best fluency in terms of perplexity and second-best fluency as per crowd judgment while reusing the least of the original document. However, its usefulness score is the lowest of all models, showing that enforcing query bias may be necessary to ensure the model does not "hallucinate". Thus, increasing the number of query-biased anchor context-based training examples might allow us to combine the strengths of the two anchor context-based models.

5.4 Summary

This chapter has discussed how to extend our media bias knowledge into other domains of tasks. In Section 5.1 we have studied the task of content transfer. We first cast content transfer as a special case of our gatekeeping

bias mitigation model discussed in Section 4.1. We demonstrated that content transfer can be tackled based on the model we developed before.

In Section 5.2 and Section 5.3, we have studied the abstractive snippet generation task. We first conducted a user study showing that the abstractive snippets potentially have similar performance compared to current snippets in Section 5.2. We also discussed users' preferences for different types of snippets. The results suggest that generated snippets are useful for the users to find out the desired websites.

The last section of this chapter presents a snippet generator. We first present the anchor contexts as the source of snippets via a distant supervision manner. Our snippet generator treats the input query as a "topic" in the gatekeeping bias mitigation task. Our intrinsic and extrinsic evaluations show that the model generates biased snippets that can be used to accurately choose relevant websites on a search results page.

In this chapter, we have learned the robustness of our models across the two NLP tasks and we have seen these NLP tasks can be casted as the tasks we have studied in the previous chapters. Through the analyses and experiments of this chapter, we have gained valuable insights into our models. As we go on the last bit of this thesis, we aim to enlighten the potential implications and contributions of our bias analyses and mitigation strategies.

6 Conclusion

> The best way to predict the future is to invent it.
>
> Alan Kay

This chapter wraps up the dissertation. Section 6.1 reviews our key findings as well as any potential implications in Chapters 3, 4, and 5. The remaining part discusses our research questions. Later in Section 6.2, we discuss the remaining open problems and future work of this thesis.

6.1 Main Contributions and Implications

As discussed in Chapter 1, Chapters 3 to 5 study textual media bias from different perspectives. Chapter 3 studies bias analysis by creating a corpus and discussing how to detect sentence-level and article-level bias. In Chapter 4 we discuss how to mitigate three kinds of media bias. Finally in Chapter 5 we aim to employ our bias analysis and mitigation knowledge in other domains of tasks.

Chapter 3 introduces the webis-bias-flipper-18 corpus. Focusing on political events in the US, this dataset collects 6,447 news articles and their bias labels. This chapter starts with discussing the process of crawling the articles and performing statistical analysis of them. In the latter two sections, based on the corpus we study (1) how to detect article-level bias given sentence-level bias and (2) how to detect sentence-level bias given article-level bias.

We aim to provide answers for the first research question (how to analyze media bias) in Chapter 3. The created corpus provides the basis of this chapter. Via the discriminativeness ratio equation, we can capture important words that discriminate between the left and right biases. The analysis provides statistical answers for subquestion 1.1 (how bias is manifested in the texts). Most importantly, the analysis also shows the difference between biased text and other domains of text: biased texts frequently use nouns, and different nouns present different biases. With this in mind, we study two bias detection tasks, one detects article-level bias from sentence-level bias, and the other one is a reversed task: detecting paragraph, sentence, and word-level bias from article-level bias. These two sections provide answers to our subquestion 1.2 how these two levels of biases correlate with each other. First of all, the findings suggest that both directions of bias detection can be successfully achieved. Secondly, the results show how biases occur at different locations. For example, the last quarter is usually the most biased part of an article. Lastly, we provide experiment results that using only lexical information is limited to detecting article-level bias. We trained a Gaussian model using only bias pattern information to detect the bias, which suggests that bias patterns (including count, location, and transition of bias) are strong features for the article-level bias detection task.

Chapter 4 studies media bias mitigation for three different kinds of biases: gatekeeping bias, coverage bias, and statement bias. For each bias, we propose a strategy to mitigate and develop a neural network model in order to mitigate the bias in the text. For gatekeeping bias, the model learns to generate a sentence according to a given new piece of information. It helps reduce gatekeeping bias by considering other relevant information regarding one event. For coverage bias, we proposed to consider framing to increase the visibility of each side. The model rewrites the sentence to change from one frame to the other. Lastly in statement bias, we studied how to change the bias from left-oriented to right-oriented, and vice versa.

Chapter 4 tries to answer the second research question of how to mitigate the bias in the text. Accordingly, each section in this chapter answers one subquestion (subquestion 2.1 to 2.3) corresponding to three kinds of bias. To mitigate gatekeeping bias in subquestion 2.1, we aim to insert new information. Our answer is to use a multi-learning model to meet the three requirements of the desired text. For each requirement, one classifier is trained. After that, these classifiers are integrated into the model to guide the training process. By reframing we can balance the visibilities of each side. In subquestion 2.2, we worked on reframing and our answer is to train the model with three strategies: *framed-language pretraining*, *named-*

entity preservation, and *adversarial learning*. The results suggest that using the three strategies together can best reframe the text properly. Lastly in mitigating the statement bias (subquestion 2.3), our model learns to keep the syntax the same while changing words such as named entities or verbs. The results suggest that the model successfully captures the difference between left and right biases.

Chapter 5 aims to expand our study in order to see if we can apply the learned bias analysis and mitigation knowledge to other domains of tasks. In this chapter, we first study the task of content transfer and apply our multi-learning model from subquestion 2.1 to it. The latter two sections are about generating abstract snippets. We first perform a user study to investigate how the abstractive snippets are used. In the last section, we train a snippet generator to generate abstractive snippets.

We try to answer Research Question 3 (beyond media bias) using the findings in Chapter 5. For subquestion 3.1, we find that the multi-task learning model we developed in the previous chapter can be applied to a new task. After re-training the classifiers and the model, the experiments provide empirical results that the generator can be applied to the context transfer task. For tasks that are much different from the media bias task, we first carefully conduct a user study to know if the generated snippets are useful in web search scenarios. After that, we cast the abstractive snippet generation problem as a special case of gatekeeping bias mitigation, we trained a snippet generator and evaluated it via crowdsourcing. The results suggest that users can use the generated snippets to find the webpages they are looking for. The last sections answer subquestion 3.2 how to apply our media bias knowledge to a new domain. Our findings suggest that we can use our bias mitigation idea on the abstractive snippet generation task.

Overall, chapters 3 through 5 collectively present a thorough study of textual media bias. The entry point is Chapter 3 where we construct a media bias corpus and conduct analyses of media bias in news articles. The subsequent focus is on mitigating media bias where we have proposed distinct strategies tailored to tackle the three types of media bias. Finally in Chapter 5, we conclude the study by examining the robustness of developed models as we apply them to other NLP tasks.

6.2 Open Problems and Future Work

We highlight several parts of our research questions that have interesting follow-up questions or remain unsolved to wrap up this chapter and the

dissertation. We also address our research's limits and promising directions for future work.

In Chapter 3, we created the corpus, analyzed bias in news media especially on the linguistic level, and performed two types of bias identification tasks. The three sections in this chapter present a comprehensive study of bias analysis. Yet, a limitation here is that we do not cover detecting other types of bias in news media. Other media biases such as coverage bias might behave differently. In terms of providing a complete study of analyzing all types of textual media biases, follow-up researchers should study the missing part of the analyses. We especially, expect the development of models capable of capturing the structure within news articles, enabling the detection of other types of media biases.

Chapter 4 demonstrates how to mitigate all three kinds of biases. Though we have shown empirical results that the mitigations work in general, they are not yet perfect. In particular, we see the performance of statement bias mitigation is limited, and the model frequently alters the meaning of the text. In the future, with a more sophisticated method, or a model with the ability to strictly keep the semantics, we should be able to see a more successful statement bias mitigation.

Another interesting follow-up is to bring up new deep learning methods such as the diffusion model (Zou et al., 2023), which has shown outstanding performance in generating graphics. In the future, if the model is more mature to be used in NLP tasks, one should expect a better improvement in generating human-like texts.

Lastly in Chapter 5, we studied two domains of tasks and applied our bias mitigation knowledge to it. We have found these two tasks work with satisfied performance. However, one limitation here is clearly that other NLP tasks can be studied as well, and other NLP tasks may not be able to apply our knowledge to them. If the task is too far away from bias analysis or bias mitigation, we have to twist the task to fit into the cases we have studied. Future work here can be to study other NLP tasks and seek the limitations of our bias analysis and mitigation models.

Overall, one shared limitation is that we did not compare our approach against those arising NLP methods such ChatGPT[1]. Once the model is stable and open to be used as a comparable baseline, future work can test whether ChatGPT can perform media bias and mitigation better. However, while ChatGPT has demonstrated to successfully conduct many NLP tasks (Qin et al., 2023), its scope is not to replace all aspects covered in this

[1] `https://chat.openai.com/`

thesis. For example, we expect that our analyses of media bias still hold, and ChatGPT can enhance the bias detection accuracy and potentially contribute to scaling up the analyses by including more texts. Secondly, the remarkable text generation capability of ChatGPT can play a role in improving the text quality of the text after bias mitigation. Nevertheless, it is crucial that humans should always be involved in the mitigation process, as ChatGPT itself has limitations in independently addressing all types of media bias.

References

Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. Text summarization techniques: a brief survey. *arXiv preprint arXiv:1707.02268*, 2017.

James Allen. *Natural language understanding*. Benjamin-Cummings Publishing Co., Inc., 1995.

Lorena Leal Bando, Falk Scholer, and Andrew Turpin. Constructing Query-biased Summaries: A Comparison of Human and System Generated Snippets. In *Proceedings of Information Interaction in Context Symposium*, pages 195–204, 2010.

David P Baron. Persistent media bias. *Journal of Public Economics*, 90(1-2): 1–36, 2006.

Jerome R Bellegarda. Statistical language model adaptation: review and perspectives. *Speech communication*, 42(1):93–108, 2004.

Frank Bentley, Katie Quehl, Jordan Wirfs-Brock, and Melissa Bica. Understanding online news behaviors. In *Proceedings of the 37th Annual ACM Conference on Human Factors in Computing Systems*, 2019.

Douglas Biber, Stig Johansson, Geoffrey Leech, Susan Conrad, and Edward Finegan. *Longman Grammar of Spoken and Written English*. 1999.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. Generating sentences from a continuous space. In *CoNLL 2016*, pages 10–21, 2016.

Amber E Boydstun, Justin H Gross, Philip Resnik, and Noah A Smith. Identifying media frames and frame dynamics within and across policy issues. In *New Directions in Analyzing Text as Data Workshop, London*, 2013.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. Faithful to the original: Fact aware neural abstractive summarization. In *Proceedings of AAAI 2018*, 2018.

Dallas Card, Amber E. Boydstun, Justin H. Gross, Philip Resnik, and Noah A. Smith. The media frames corpus: Annotations of frames across issues. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 438–444, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-2072. URL https://www.aclweb.org/anthology/P15-2072.

Tuhin Chakrabarty, Christopher Hidey, and Smaranda Muresan. ENTRUST: Argument reframing with language models and entailment. *CoRR*, abs/2103.06758, 2021. URL https://arxiv.org/abs/2103.06758.

Wei-Fan Chen, Khalid Al Khatib, Benno Stein, and Henning Wachsmuth. Fact-based text content transfer through multi-task learning. In *Under Review*, -.

Wei-Fan Chen, Matthias Hagen, Benno Stein, and Martin Potthast. A user study on snippet generation: Text reuse vs. paraphrases. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1033–1036, 2018a.

Wei-Fan Chen, Henning Wachsmuth, Khalid Al-Khatib, and Benno Stein. Learning to flip the bias of news headlines. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 79–88, Tilburg University, The Netherlands, November 2018b. Association for Computational Linguistics. doi: 10.18653/v1/W18-6509. URL https://aclanthology.org/W18-6509.

Wei-Fan Chen, Khalid Al Khatib, Henning Wachsmuth, and Benno Stein. Analyzing political bias and unfairness in news articles at different levels of granularity. In *Proceedings of the Fourth Workshop on Natural Language Processing and Computational Social Science*, pages 149–154, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.nlpcss-1.16. URL https://aclanthology.org/2020.nlpcss-1.16.

Wei-Fan Chen, Khalid Al Khatib, Benno Stein, and Henning Wachsmuth. Detecting media bias in news articles using gaussian bias distributions.

In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4290–4300. Association for Computational Linguistics, 2020b. doi: 10.18653/v1/2020.findings-emnlp.383. URL https://doi.org/10.18653/v1/2020.findings-emnlp.383.

Wei-Fan Chen, Shahbaz Syed, Benno Stein, Matthias Hagen, and Martin Potthast. Abstractive snippet generation. In *Proceedings of The Web Conference 2020*, pages 1309–1319, 2020c.

Wei-Fan Chen, Khalid Al Khatib, Benno Stein, and Henning Wachsmuth. Controlled neural sentence-level reframing of news articles. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 2683–2693. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.findings-emnlp.228. URL https://doi.org/10.18653/v1/2021.findings-emnlp.228.

Dennis Chong and James N Druckman. Framing theory. *Annual Review of Political Science*, pages 103–126, 2007.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008.

Gordon V. Cormack, Mark D. Smucker, and Charles L.A. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Information retrieval*, 14(5):441–465, 2011.

Dave D'Alessio and Mike Allen. Media bias in presidential elections: A meta-analysis. *Journal of communication*, 50(4):133–156, 2000.

Claes H De Vreese. News framing: Theory and typology. *Information design journal+ document design*, 13(1):51–62, 2005.

Stefano DellaVigna and Ethan Kaplan. The fox news effect: Media bias and voting. *The Quarterly Journal of Economics*, 122(3):1187–1234, 2007.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceed-*

ings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Lisa Fan, Marshall White, Eva Sharma, Ruisi Su, Prafulla Kumar Choubey, Ruihong Huang, and Lu Wang. In plain sight: Media bias through the lens of factual reporting. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6342–6348. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1664. URL https://doi.org/10.18653/v1/D19-1664.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. Style transfer in text: Exploration and evaluation. In *Proceedings of the Thirty-Second Conference on Artificial Intelligence*, 2018.

Rama Rohit Reddy Gangula, Suma Reddy Duggenpudi, and Radhika Mamidi. Detecting political bias in news articles using headline attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 77–84, 2019.

Albert Gatt and Emiel Krahmer. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170, 2018.

Matthew Gentzkow and Jesse M Shapiro. What drives media slant? evidence from us daily newspapers. *Econometrica*, 78(1):35–71, 2010.

Sean Gerrish and David M Blei. Predicting legislative roll calls from text. In *Proceedings of the 28th international conference on machine learning*, pages 489–496, 2011.

Bradley W Gorham. Stereotypes in the media: So what? *Howard Journal of Communication*, 10(4):229–247, 1999.

Stephan Greene and Philip Resnik. More than words: Syntactic packaging and implicit sentiment. In *Proceedings of human language technologies: The 2009 annual conference of the north american chapter of the association for*

computational linguistics, pages 503–511. Association for Computational Linguistics, 2009.

Tim Groseclose and Jeffrey Milyo. A measure of media bias. *The Quarterly Journal of Economics*, 120(4):1191–1237, 2005.

Max Grusky, Mor Naaman, and Yoav Artzi. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of NAACL/HLT 2018*, pages 708–719, 2018.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of ACL 2016*, pages 1631–1640, 2016.

Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. Generating sentences by editing prototypes. *arXiv preprint arXiv:1709.08878*, 2017.

Matthias Hagen, Martin Potthast, Anna Beyer, and Benno Stein. Towards optimum query segmentation: in doubt without. In *Proceedings of CIKM 2012*, pages 1015–1024, 2012.

Felix Hamborg, Karsten Donnay, and Bela Gipp. Automated identification of media bias in news articles: an interdisciplinary literature review. *International Journal on Digital Libraries*, 20(4):391–415, 2019.

Catherine Happer and Greg Philo. The role of the media in the construction of public belief and social change. *Journal of social and political psychology*, 1(1):321–336, 2013.

Mareike Hartmann, Tallulah Jansen, Isabelle Augenstein, and Anders Søgaard. Issue framing in online discussion fora. In *Proceedings of NAACL-HLT*, pages 1401–1407, 2019.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. URL `http://arxiv.org/abs/1503.02531`.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard H. Hovy. Learning whom to trust with MACE. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 1120–1130. The Association for Computational Linguistics, 2013. URL `https://aclanthology.org/N13-1132/`.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *International Conference on Machine Learning*, pages 1587–1596, 2017.

Kuan-Hao Huang, Chen Li, and Kai-Wei Chang. Generating sports news from live commentary: A chinese dataset for sports game summarization. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 609–615, 2020.

Shanto Iyengar and Kyu S Hahn. Red media, blue media: Evidence of ideological selectivity in media use. *Journal of communication*, 59(1):19–39, 2009.

Mohit Iyyer, Peter Enns, Jordan L. Boyd-Graber, and Philip Resnik. Political ideology detection using recursive neural networks. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 1113–1122. The Association for Computer Linguistics, 2014. doi: 10.3115/v1/p14-1105. URL https://doi.org/10.3115/v1/p14-1105.

John WC Johnstone, Edward J Slawski, and William W Bowman. The professional values of american newsmen. *Public opinion quarterly*, 36(4):522–540, 1972.

Johannes Kiesel, Benno Stein, and Stefan Lucks. A large-scale analysis of the mnemonic password advice. In *Proceedings of NDSS 2017*, 2017.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. Semeval-2019 task 4: Hyperpartisan news detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 829–839, 2019.

Michelle YoungJin Kim and Kristen Johnson. Close: Contrastive learning of subframe embeddings for political bias classification of news media. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2780–2793, 2022.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *The 2nd International Conference on Learning Representations*, 2013.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017*, 2017.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 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*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

Chang Li and Dan Goldwasser. Encoding social information with graph convolutional networks forpolitical perspective detection in news media. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2594–2604, 2019.

Juncen Li, Robin Jia, He He, and Percy Liang. Delete, retrieve, generate: A simple approach to sentiment and style transfer. In *North American Association for Computational Linguistics*, 2018.

Wei-Hao Lin, Theresa Wilson, Janyce Wiebe, and Alexander G Hauptmann. Which side are you on? identifying perspectives at the document and sentence levels. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 109–116, 2006.

Yu-Ru Lin, James P. Bagrow, and David Lazer. More voices than ever? Quantifying media bias in networks. In *Proceedings of the Fifth International Conference on Weblogs and Social Media*, 2011.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. In *Proceedings of ICLR 2018*, 2018.

Siyi Liu, Sihao Chen, Xander Uyttendaele, and Dan Roth. Multioped: A corpus of multi-perspective news editorials. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 4345–4361. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.344. URL https://doi.org/10.18653/v1/2021.naacl-main.344.

Bill MacCartney. *Natural language inference*. PhD thesis, 2009.

David Maxwell, Leif Azzopardi, and Yashar Moshfeghi. A Study of Snippet Length and Informativeness: Behaviour, Performance and User Experience. In *Proceedings of SIGIR*, pages 135–144, 2017.

Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4): 1093–1113, 2014.

Julia Mendelsohn, Ceren Budak, and David Jurgens. Modeling framing in immigration discourse on social media. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2219–2263, 2021.

Jonas Mueller, David Gifford, and Tommi Jaakkola. Sequence to better sequence: Continuous revision of combinatorial structures. In *International Conference on Machine Learning*, pages 2536–2544, 2017.

Sendhil Mullainathan and Andrei Shleifer. Media bias, 2002.

Sendhil Mullainathan and Andrei Shleifer. The market for news. *American economic review*, 95(4):1031–1053, 2005.

Nona Naderi and Graeme Hirst. Classifying frames at the sentence level in news articles. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 536–542, 2017.

An Nguyen and Mark Western. The complementary relationship between the internet and traditional mass media: the case of online news and information. *Inf. Res.*, 11(3), 2006. URL `http://www.informationr.net/ir/11-3/paper259.html`.

Souneil Park, Seungwoo Kang, Sangyoung Chung, and Junehwa Song. A computational framework for media bias mitigation. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(2):1–32, 2012.

Thomas E Patterson and Wolfgang Donsbagh. News decisions: Journalists as partisan actors. *Political communication*, 13(4):455–468, 1996.

Jan Pedersen, Doug Cutting, and John Tukey. Snippet Search: A single phrase approach to text access. In *Proceedings of the 1991 Joint Statistical Meetings*, 1991.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Confer-*

ence on Empirical Methods in Natural Language Processing, pages 1532–1543, 2014.

Martin Potthast, Wei-Fan Chen, Matthias Hagen, and Benno Stein. A plan for ancillary copyright: Original snippets. 2018.

Horst Pöttker. News and its communicative quality: The inverted pyramid—when and why did it appear? *Journalism Studies*, 4(4):501–511, 2003.

Shrimai Prabhumoye, Chris Quirk, and Michel Galley. Towards content transfer through grounded text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2622–2632, 2019.

Reid Pryzant, Richard Diehl Martinez, Nathan Dass, Sadao Kurohashi, Dan Jurafsky, and Diyi Yang. Automatically neutralizing subjective bias in text. In *Thirty-Forth AAAI Conference on Artificial Intelligence*, 2020.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*, 2023.

Lianhui Qin, Michel Galley, Chris Brockett, Xiaodong Liu, Xiang Gao, William B Dolan, Yejin Choi, and Jianfeng Gao. Conversing by reading: Contentful neural conversation with on-demand machine reading. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5427–5436, 2019.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67, 2020.

BiChen Rao and Erkang Zhu. Searching web data using minhash lsh. In *Proceedings of SIGMOD 2016*, pages 2257–2258, 2016.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. Truth of varying shades: Analyzing language in fake news and

political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937, 2017.

Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. Linguistic models for analyzing and detecting biased language. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1650–1659, 2013.

Douglas A Reynolds. Gaussian mixture models. *Encyclopedia of Biometric Recognition*, 2009.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning*, pages 1278–1286, 2014.

Michael Ryan. Journalistic ethics, objectivity, existential journalism, standpoint epistemology, and public journalism. *Journal of Mass Media Ethics*, 16(1):3–22, 2001.

Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1073–1083, 2017.

Matthew Shardlow. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70, 2014.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, pages 6833–6844, 2017.

Henry I Silverman et al. Reuters: Principles of trust or propaganda? *Journal of Applied Business Research (JABR)*, 27(6):93–116, 2011.

James Thorne and Andreas Vlachos. Evidence-based factual error correction. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3298–3309. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.256. URL https://doi.org/10.18653/v1/2021.acl-long.256.

Anastasios Tombros and Mark Sanderson. Advantages of Query Biased Summaries in Information Retrieval. In *Proceedings of SIGIR*, pages 2–10, 1998.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of NAACL/HLT 2003*, pages 173–180, 2003.

Henning Wachsmuth. *Text Analysis Pipelines–Towards Ad-hoc Large-scale Text Mining*, volume 9383 of *Lecture Notes in Computer Science*. Springer, December 2015. ISBN 978-3-319-25740-2. doi: 10.1007/978-3-319-25741-9. URL `http://is.uni-paderborn.de/uploads/tx_sibibtex/Wachsmuth2015_Dissertation.pdf`.

Henning Wachsmuth, Martin Trenkmann, Benno Stein, Gregor Engels, and Tsvetomira Palakarska. A review corpus for argumentation analysis. In Alexander Gelbukh, editor, *Proceedings of the 15th International Conference on Intelligent Text Processing and Computational Linguistics*, pages 115–127, Berlin Heidelberg New York, 2014. Springer. ISBN 978-3-642-54902-1. doi: 10.1007/978-3-642-54903-8_10.

Kohei Watanabe. Measuring news bias: Russia's official news agency itar-tass'coverage of the ukraine crisis. *European Journal of Communication*, 32 (3):224–241, 2017.

Albert Webson, Zhizhong Chen, Carsten Eickhoff, and Ellie Pavlick. Are "undocumented workers" the same as "illegal aliens"? Disentangling denotation and connotation in vector spaces. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 4090–4105, 2020.

W Michael Weis. Government news management, bias and distortion in american press coverage of the brazilian coup of 1964. *The Social Science Journal*, 34(1):35–55, 1997.

Ryen White, Ian Ruthven, and Joemon M. Jose. Finding Relevant Documents Using Top Ranking Sentences: An Evaluation of Two Alternative Schemes. In *Proceedings of SIGIR*, pages 57–64, 2002a.

Ryen White, Ian Ruthven, and Joemon M. Jose. The Use of Implicit Evidence for Relevance Feedback in Web Retrieval. In *Proceedings of ECIR*, pages 93–109, 2002b.

John Wieting, Jonathan Mallinson, and Kevin Gimpel. Learning paraphrastic sentence embeddings from back-translated bitext. In *Proceedings of EMNLP 2017*, 2017.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.emnlp-demos.6`.

Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. Improved variational autoencoders for text modeling using dilated convolutions. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3881–3890, 2017.

Tae Yano, Philip Resnik, and Noah A Smith. Shedding (a thousand points of) light on biased language. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 152–158. Association for Computational Linguistics, 2010.

Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1404. URL `https://aclanthology.org/D19-1404`.

Elaine Yuan. News consumption across multiple media platforms: A repertoire approach. *Information, communication & society*, 14(7):998–1016, 2011.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=SkeHuCVFDr`.

Chunting Zhou and Graham Neubig. Multi-space variational encoder-decoders for semi-supervised labeled sequence transduction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 310–320, 2017.

Hao Zou, Zae Myung Kim, and Dongyeop Kang. Diffusion models in nlp: A survey. *arXiv preprint arXiv:2305.14671*, 2023.