Model Engineering College, Kochi
Department of Computer Engineering
B.Tech Computer Science Engineering
CSD334 Mini Project
Literature Survey

22CSA16 MDL22CS049 Aravind Ashokan
22CSA30 MDL22CS093 Harishanker S Nair
22CSA50 MDL22CS154 Pradyumn R Pai
22CSA51 MDL22CS155 Pranav P S

January 2025

1 Introduction

This project aims to develop a web extension that identifies and highlights bias in online news articles, empowering users to critically assess the information they consume. Leveraging advanced Natural Language Processing (NLP) techniques, the extension will analyze articles in real-time, identifying patterns and classifying sentiment to detect potential biases. By integrating a pre-trained BERT model, the system will offer a nuanced understanding of the article's content, providing users with an objective evaluation of its bias and encouraging informed decision-making.

2 Articles

2.1 Decoding News Bias: Multi Bias Detection in News Articles [1]

This paper explores the detection of multiple biases in news articles using Large Language Models (LLMs). It addresses the limitations of previous research by examining a broader range of biases, not just political or gender biases.
Key contributions include:
- Expanding the scope of bias detection to include political, gender, entity, racial, religious, regional and sensational biases.
- Using LLMs for dataset annotation.

- Comparing the performance of various transformer-based models in detecting these biases.

Methodology:
A dataset was created from six domains (Hollywood, fashion, finance, religion, politics, and sports). GPT-4o mini was used to label the data based on specific bias definitions. The model was given detailed system and user instructions. The dataset was filtered to include only articles with at least one bias. Transformer models (BERT, RoBERTa, ALBERT, DistilBERT, and XLNet) were trained using inverse frequency weighting to counter class imbalance. Multilabel Stratified KFold splitting was used to maintain label distribution.

Key findings: BERT performed the best overall, especially in political bias detection. Class imbalance was a challenge, particularly for racial and regional biases. LLM annotations had some inconsistencies, especially with religious, regional, and political biases, where articles were sometimes incorrectly labelled as biased.

Limitations and future work: The study acknowledges potential inconsistencies in LLM-based labeling. Class imbalance within the dataset is a significant limitation. Future work includes improving LLM annotation reliability, expanding the dataset and investigating alternative data augmentation techniques.

2.2 Neural Media Bias Detection Using Distant Supervision With BABE - Bias Annotations By Experts [2]

Introduction: This paper introduces BABE (Bias Annotations By Experts), a new high-quality media bias dataset for detecting bias in news articles. The dataset contains 3,700 sentences with bias annotations at both the word and sentence level. A key aspect of this work is that the annotations were performed by trained experts, not crowdsourced workers.

Key aspects:
Data Collection: The sentences were extracted from 14 US news outlets covering 12 controversial topics, collected from January 2017 to June 2020. The dataset consists of 1,700 sentences from the MBIC dataset and 2,000 additional sentences. The data collection focused on US media due to the increasing political polarisation in the country.
Expert Annotation: The annotators were master's students with experience in media bias, who underwent training to identify biased wording, distinguish between bias and polarizing language, and maintain a neutral viewpoint. The experts were paid for their work, and their annotations were reviewed and refined through weekly discussions.
Data Organisation: The dataset was divided into two subgroups:

SG1: 1,700 sentences annotated by eight experts.

SG2: 3,700 sentences annotated by five experts.

Evaluation: The expert annotations were compared to the crowdsourced labels in the MBIC dataset. The expert annotations showed significantly higher inter-

annotator agreement than the crowdsourced labels. Krippendorff's alpha was used as the agreement metric.
Bias Detection: The paper also presents a neural BERT-based classifier trained on the BABE dataset. The model was pre-trained using a distant supervision approach to learn bias-specific embeddings, further improving its performance. The best performing model achieved a macro F1-score of 0.804. The pre-training corpus consisted of news headlines from outlets with and without partisan leaning.
Comparison with Existing Work: The BABE dataset and the resulting bias detection system address the weaknesses of existing media bias datasets by having expert annotators, a broader range of topics, and word-level annotations. Previous work often relies on crowdsourcing, which has resulted in low annotator agreement.
Key Findings: The study shows that expert annotation leads to a better quality dataset, and that the BERT-based classifier using distant supervision significantly outperforms existing approaches. The paper also demonstrates the potential of neural network models in media bias detection.

2.3 Detecting Political Bias in News Articles Using Headline Attention [3]

Introduction: This paper introduces a Headline Attention Network for detecting political bias in news articles, along with a manually annotated dataset. The key idea is that the headline of an article influences how a reader perceives the rest of the text. The model uses an attention mechanism to focus on the parts of the article that are most relevant to the bias indicated in the headline.
Key aspects:
Problem: The paper addresses the challenge of automatically detecting political bias in news articles. This is important because media bias can distort facts and influence public Dataset: The researchers created a dataset of 1329 Telugu news articles, annotated for bias towards one of five political parties (BJP, Congress, TRS, TDP, YCP) or as unbiased. The articles were annotated by native Telugu speakers with political knowledge.
Headline Attention Network: The model has three main components:
Headline Encoder: Uses a bidirectional LSTM to create a contextual encoding of the headline.
Article Encoder: Uses a bidirectional LSTM to encode the article's words, incorporating contextual information.
Headline Attention Layer: This layer is key. It calculates the importance of each word in the article based on its relationship to the headline, then creates a weighted representation of the article.
Bias Detection: The weighted article representation is then used to classify the article's bias towards a particular political party.
Methodology:
The model mimics how a person reads an article, first registering the headline

and then interpreting the article in that context. The attention mechanism focuses on the most relevant parts of the news article based on the headline.
Results: The Headline Attention Network outperformed various baseline models, including Naive Bayes, SVMs, CNNs, and LSTMs, showing that attending to the article based on the headline significantly improves bias detection. The model achieved an accuracy of 89.54 percentage compared to the previous best baseline of 85.32 percentage.
Key Findings:
The study found that headlines are often used to express the ideological view of the news. The headline attention mechanism is effective in finding important words causing bias in a news article. Simply concatenating the headline and article does not improve accuracy as much as the headline attention mechanism. The researchers also provided a visualisation of the attention mechanism showing how the model selects words with strong emphasis on a person or a political party. Words that are most important in predicting bias towards a party are highlighted more intensely.
This work provides a new approach to detecting political bias by focusing on the relationship between the headline and the article, along with a new dataset for the Telugu language. This approach can also be extended to other text classification tasks involving titles/headlines and a body of text.

2.4 On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis [4]

Introduction:NLP is a field that deals with the interaction between computers and human language. It involves enabling computers to understand, interpret, and generate human language. Words are often considered the basic units of text in many languages. Tokenization is often the first module in an NLP pipeline, which transforms texts to sequences of words. Other preprocessing techniques such as lemmatization, lowercasing, and multiword grouping are often used alongside tokenization. Word embeddings are important in boosting the generalization capabilities of neural systems in NLP.
Text Preprocessing Techniques:
Tokenization: This is the initial step in the NLP pipeline and involves transforming text into sequences of individual words, using white spaces to delimit words in the example provided.
Lowercasing: This converts all text to lowercase. For example, "Apple" becomes "apple".
Lemmatization: This process reduces words to their base or dictionary form. For example, the words "asking" and "asked" would be reduced to the base form "ask".
Multiword grouping: This combines words that form a single multiword expression.

The paper notes that although these techniques have been studied in conventional text classification, there has been little attention paid to them in neural-based models. The study aims to address this gap, highlighting the importance of considering preprocessing steps when evaluating different models. The study also considers the impact of the preprocessing of the training corpus on the final performance of neural network text classifiers. It highlights that, in some cases, techniques like lowercasing and lemmatizing do not seem to help. For example, lowercasing may limit coverage by not including capitalized entities, and lemmatizing may miss inflected forms.

2.5 BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [5]

This paper introduces BERT (Bidirectional Encoder Representations from Transformers), a new language representation model designed to pre-train deep bidirectional representations from unlabeled text. Unlike previous models, BERT jointly conditions on both left and right context in all layers. This allows the pre-trained model to be fine-tuned with just one additional output layer to achieve state-of-the-art results on a wide range of natural language processing (NLP) tasks without substantial task-specific modifications.
BERT uses a masked language model (MLM) objective, inspired by the Cloze task, to overcome the limitations of unidirectional language models. The MLM randomly masks some input tokens and trains the model to predict the original word based on its context. This enables the model to fuse left and right context, allowing for deep bidirectional pre-training. In addition to MLM, a "next sentence prediction" task is used to jointly pre-train text-pair representations.
Pre-training and Fine-tuning:
The framework involves two steps: pre-training and fine-tuning. During pre-training, the model is trained on unlabeled data. For fine-tuning, the pre-trained parameters are used to initialise the model, and all parameters are fine-tuned using labelled data from the downstream tasks. Each downstream task has separate fine-tuned models, though they are all initialised with the same pre-trained parameters. A key feature of BERT is its unified architecture across different tasks, with minimal difference between the pre-trained and downstream architectures.
Model Architecture:
BERT uses a multi-layer bidirectional Transformer encoder. The model is available in two sizes: BERTBASE (12 layers, 768 hidden units, 12 attention heads, 110M parameters) and BERTLARGE (24 layers, 1024 hidden units, 16 attention heads, 340M parameters). BERTBASE is designed to have the same model size as OpenAI GPT for comparison, but it uses bidirectional self-attention.
Input Representation:
BERT can handle both single sentences and pairs of sentences by using a special classification token ([CLS]) at the beginning of every sequence, and a separator token ([SEP]) between sentences. Each token's input representation is constructed by summing token, segment, and position embeddings.

Pre-training Tasks:
BERT is pre-trained using two unsupervised tasks: the masked language model (MLM) and next sentence prediction (NSP). The MLM task involves predicting masked words in a sentence. The NSP task involves predicting whether two given sentences follow each other in the text.
Fine-tuning:
Fine-tuning involves plugging task-specific inputs and outputs into BERT and fine-tuning all the parameters end-to-end. For text pair tasks, the model uses a concatenated text pair with self-attention that effectively includes bidirectional cross-attention between two sentences. Fine-tuning is relatively inexpensive compared to pre-training.
Experimental Results:
BERT achieved state-of-the-art results on eleven NLP tasks, including GLUE, SQuAD v1.1, SQuAD v2.0, and SWAG. On the GLUE benchmark, BERT outperforms previous models by a substantial margin. In question answering (SQuAD), BERT also achieves significant improvements. The model performs well on the SWAG dataset for commonsense inference, outperforming existing systems.
Ablation Studies:
The paper includes several ablation studies to evaluate the relative importance of different components. These studies show that the deep bidirectionality of BERT and the next sentence prediction task contribute significantly to its performance. Larger models lead to better performance even on small datasets. The model is effective for both fine-tuning and feature-based approaches.
Comparison to Other Models:
BERT differs from models like ELMo and OpenAI GPT. ELMo is a feature-based approach using a shallow concatenation of independently trained left-to-right and right-to-left language models. OpenAI GPT uses a left-to-right architecture. BERT is a fine-tuning approach, and jointly conditions on both left and right context in all layers.

2.6 Automatic Text Summarization: A Comprehensive Survey [6]

Automatic Text Summarization (ATS) has become increasingly vital due to the exponential growth of textual data online and in archives. Manual summarization is not only time-consuming and expensive but also impractical for the massive volumes of data, making ATS a crucial solution. An effective summary must concisely capture the main ideas of the original text while minimizing redundancy.

ATS systems are broadly classified into single-document and multi-document summarization, with approaches categorized as extractive, abstractive, or hybrid. Extractive summarization selects and concatenates the most important sentences from the original text, offering speed and accuracy but often suffering from redundancy and lack of cohesion. Abstractive summarization, on the other

hand, creates new sentences based on an internal semantic representation, leading to more human-like summaries but facing challenges such as complexity and computational demands. Hybrid methods combine these two approaches, typically starting with extractive techniques before applying abstractive refinements to improve readability and coherence. While extractive methods dominate current research, abstractive and hybrid approaches are gaining attention due to their potential to better mimic human-generated summaries.

ATS systems are further categorized based on their outputs, such as generic versus query-based summaries, monolingual versus multilingual summaries, and supervised versus unsupervised techniques. Other classification criteria include the summary content, type, and domain. ATS systems rely on a range of tools and techniques, including statistical features, linguistic analysis, and soft computing approaches such as machine learning and fuzzy logic, to identify and rank key information. Preprocessing steps such as parsing and semantic analysis are essential for effective summarization.

Evaluation of ATS systems employs datasets like DUC, CNN, and LCSTS, with both manual and automated evaluation methods. Popular metrics include ROUGE, precision, and recall. However, challenges persist, including issues with multi-document summarization, redundancy reduction, user-specific customization, long text processing, and the lack of diverse non-English datasets. Ensuring the readability, cohesion, and semantic accuracy of summaries remains a significant hurdle, as does designing robust evaluation methods given the inherently subjective nature of summarization tasks.

2.7 Automatic Text Summarization [7]

This paper introduces an approach to automatic text summarization using a trainable summarizer. The summarizer considers several features, including sentence position, positive and negative keywords, sentence centrality, resemblance to the title, inclusion of named entities and numerical data, relative sentence length, bushy path, and aggregated similarity.

The approach models sentences as vectors of features, treating the summarization task as a classification problem by labeling sentences as either "correct" (belonging to the extractive summary) or "incorrect." The trainable summarizer learns patterns that lead to summaries by identifying feature values correlated with the "correct" or "incorrect" classes and assigns a score between 0 and 1 to each sentence. Sentences are extracted based on a predefined compression rate.

The paper uses a genetic algorithm (GA) and mathematical regression (MR) models to combine feature weights. In training mode, features are extracted from 50 manually summarized English documents and used to train the GA and MR models. In testing mode, features are extracted from 100 different English documents, which are then summarized.
The following features are used:
- Sentence Position: Sentences are ranked by their position in a paragraph.
- Positive Keyword: Keywords frequently included in the summary are identified using a formula involving the term frequency of the keywords in the sentence

and their probability of appearing in summaries.
- Negative Keyword: Keywords unlikely to occur in the summary are identified.
- Sentence Centrality: Vocabulary overlap between a sentence and other sentences in the document.
- Sentence Resemblance to the Title: Vocabulary overlap between a sentence and the document title.
- Sentence Inclusion of Named Entity: Count of proper nouns in a sentence.
- Sentence Inclusion of Numerical Data: Count of numerical data in a sentence.
- Sentence Relative Length: Penalizes short sentences using relative length.
- Bushy Path of the Node: Number of links connecting a sentence to other sentences.
- Aggregate Similarity: Sum of weights on the links to other sentences.

The GA model integrates the ten features using a weighted score function. A chromosome represents the combination of all feature weights. The GA is trained using 50 manually summarized documents, evaluating 100 generations to obtain an optimal combination of feature weights. During testing, the trained weights are used to rank sentences, and the highest-scoring sentences are included in the summary based on the compression rate.

The MR model also estimates feature weights. Feature parameters are used as input variables to produce a statistical model, which is then evaluated using testing data.

The experiment uses 150 English religious articles, with 50 manually summarized articles for training and 100 for testing. System performance is measured using precision, based on coverage between machine-generated summaries and manual summaries. The MCBA+GA approach, which uses only the first five features, serves as a baseline.

The effect of each feature on summarization is investigated individually. Results of the GA and MR models are presented and compared to the baseline. Both models outperform the baseline approach.

2.8 Enhancing Media Literacy: The Effectiveness of (Human) Anno- tations and Bias Visualizations on Bias Detection [8]

This paper investigates the effectiveness of using human and AI-generated labels to train people to detect media bias. The study explores how well people can identify bias in new, unmarked articles after being trained with labelled examples.

Two experiments were conducted. In the first, participants were trained with sentences marked as biased by either human annotators or AI models, and then tested on their ability to detect bias in new sentences. The results showed that both human and AI labels improved bias detection compared to a control group. Human labels were more effective than AI labels, though both were effective. The study also found that participants' ability to detect bias improved from

the training to the testing phase. The training effect was independent of the political leaning of the participants.

The second experiment tested the effectiveness of different visualisation strategies at both the sentence and article levels. The training included variations such as highlighting biased sentences, biased phrases, and politicized phrases. The test phase required participants to identify biased phrases in a new article. The study found that training with biased phrase labels was most effective in improving bias detection, while training with politicised phrase labels decreased performance. Additionally, more conservative participants were less accurate in identifying bias and were also less influenced by the sentence labeling training.

The paper concludes that both human and AI-generated bias labels can increase awareness of media bias, with phrase-level highlighting being the most effective method. Although automated labels are not as effective as human labels, they still have the potential to be useful for large-scale applications. The researchers also propose that their approach could be used to develop tools for news platforms and media literacy education.

3 Proposal

We can develop a web extension that identifies and highlights bias in online news articles, empowering users to critically evaluate the information they consume. Using Natural Language Processing (NLP) techniques and a machine learning model, the system will analyze articles in real time, pinpointing biased or misleading content and providing transparent annotations. Training data will be sourced from diverse media outlets, ensuring a balanced perspective, while preprocessing steps like text cleaning and n-gram generation will enhance model accuracy. By promoting media literacy and addressing the issue of polarization, this extension aims to foster informed decision-making and contribute to a more transparent information landscape.

4 Conclusion

This project aims to bridge the gap between content consumption and critical analysis by providing a robust tool for identifying and understanding bias in online news articles. By leveraging cutting-edge NLP techniques and machine learning models, the proposed web extension will empower users to engage with media more thoughtfully and objectively. Ultimately, this solution aspires to combat the effects of media polarization, promote transparency, and encourage balanced perspectives, fostering a more informed and equitable information ecosystem.

References

1. Bhushan Santosh Shah, Deven Santosh Shah, Vahida Attar. "Decoding News Bias: Multi Bias Detection in News Articles". In: (2025)

2. Timo Spinde, Manuel Plank, Jan-David Krieger, Terry Ruas, Bela Gipp, Akiko Aizawa. "Neural Media Bias Detection Using Distant Supervision With BABE - Bias Annotations By Experts". In: (2021)

3. Rama Rohit Reddy, Suma Reddy Duggenpudi, Radhika Mamidi. "Detecting Political Bias in News Articles Using Headline Attention". In: (2019)

4. Jose Camacho-Collados, Mohammad Taher Pilehvar. "On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis". In: (2018)

5. Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: (2019)

6. Mohamed Abdel Fattah, and Fuji Ren. "Automatic Text Summarization". In(2008)

7. Wafaa Samy El-Kassas, Cherif Salama, Ahmed Rafea, Hoda K. Mohamed. "Automatic Text Summarization: A Comprehensive Survey". In: (2020)

8. Timo Spinde, Fei Wu, Wolfgang Gaissmaier, Gianluca Demartini, Helge Giese. "Enhancing Media Literacy: The Effectiveness of (Human) Annotations and Bias Visualizations on Bias Detection". In: (2024)