WikiBias as an Extrapolation Corpus for Bias Detection

Karla Salas-Jimenez[1,2], **Francisco López-Ponce,**[1,2]**,**
Sergio-Luis Ojeda-Trueba[1], **Gemma Bel-Enguix**[1,3]
[1]Grupo de Ingeniería Lingüística - UNAM
[2]Posgrado en Ciencias e Ingeniería de la Computación - UNAM
[3]Departament de Filologia Catalana i Lingüística General - Universitat de Barcelona
{karla_dsj,francisco.lopez.ponce}@ciencias.unam.mx, {SOjedaT,gbele}@iingen.unam.mx

Abstract

This paper explores whether it is possible to train a machine learning model using Wikipedia data to detect subjectivity in sentences and generalize effectively to other domains. To achieve this, we performed experiments with the WikiBias corpus, the BABE corpus, and the CheckThat! Dataset. Various classical models for ML were tested, including Logistic Regression, SVC, and SVR, including characteristics such as Sentence Transformers similarity, probabilistic sentiment measures, and biased lexicons. Pre-trained models like DistilRoBERTa, as well as large language models like Gemma and GPT-4, were also tested for the same classification task.

1 Introduction

Subjectivity permeates all spheres and experiences of human life. Language, as a representation of reality, is not exempt from subjectivity. When an author's perspective is presented as absolute truth, the text is said to contain subjective bias. Technically, it is cognitively impossible to write a text or construct a corpus without some form of bias. Although showing the author's position is not always a wrong approach, and in some genres it is even considered advisable, this is not always the case. A multitude of textual content such as textbooks, scientific articles or news presentations need to maintain neutrality as much as possible by avoiding bias.

The creation of objective texts is a long standing concern for academia as well as for many areas of society. Science, law, information, politics and governmental communication, among others, require verifiable texts that leave aside the author's subjectivity. In journalism, for example, the objective, fact-based style has traditionally been encouraged.

In 2001 Wikipedia introduced its Neutral Point of View (NPOV) policy[1], which applies to all articles written in this collaborative encyclopedia. The NPOV encompasses the following principles: a) avoid stating opinions as facts, b) avoid stating seriously contested assertions as facts, c) avoid stating facts as opinions, d) prefer nonjudgmental language, and e) indicate the relative prominence of opposing views. To comply with this policy, published texts are periodically reviewed and neutralized.

In order to achieve neutral language, Wikipedia performs periodic reviews of articles, attempting to identify and eliminate bias elements. This has allowed the development of various resources by comparing original and de-biased versions of articles, such as the NPOV corpus (Recasens et al., 2013) and WikiBias (Pryzant et al., 2020).

Subjective bias is a problem that goes beyond the used lexicon. Depending on the domain in question various forms of bias appear. In this paper we ask if it is possible to train a ML model (using a bias detection dataset) that generalizes well enough to be extrapolated to other domains. The training corpus is the WikiBias corpus, explicitly elaborated on the neutralization processes of Wikipedia. We ask ourselves if the information learned from Wikipedia can correctly classify bias in different contexts.

The paper is structured as follows: Section 2 explains the state of the art corpora and algorithms for bias detection in English. The experiments performed with different corpora and the results are explained in section 3. Section 4 discusses the conclusions and future work. Finally, the paper closes with the limitations of this work in section 5.

2 Related Work

Bias detection systems are a recent development in NLP, which has grown in recent years in part due to research conducted with Wikipedia-based corpora. One of the first approaches corresponds

[1] Wikipedia: Neutral point of view

Proceedings of the First Workshop on Advancing Natural Language Processing for Wikipedia, pages 46–52
November 16, 2024 ©2024 Association for Computational Linguistics

to (Recasens et al., 2013), who had the goal of identifying the word that introduces subjective bias. Their work was based on the study of Wikipedia reviews, considering the edition history of different articles (Max and Wisniewski, 2022; Zanzotto and Pennacchiotti, 2010). Recasens et al. (2013) proposed a classification of bias into two categories: framing bias (such as words of praise or specific perspectives) and epistemological bias (related to presupposed or implied propositions). They collected the NPOV Corpus for their study, which contains Wikipedia edits especially aimed at suppressing bias. To carry out the automatic identification of bias, the authors collect a 'bias lexicon' from the NPOV corpus. The presence or not of biased words serves a characteristic in a logistic regression system, obtaining 34% of accuracy. Pryzant et al. (2020) extended this corpus, and created the Wiki Neutrality Corpus (WNC), by adding a third type of bias: demographic bias, defined as text with presuppositions about particular genders, races, or other demographic categories (e.g. all engineers are male). In the work, the authors proposed two ways to neutralize the biased text: a modular approach, that divides the problem into two subtasks: detection and edition; and a concurrent system combining the two subtasks into a single step. In both cases, the detection was carried out using a BERT-based detector (Devlin et al., 2018), and a LSTM decoder.

More recently Zhong (2021), identified that the WNC corpus (Pryzant et al., 2020) has a series of issues: first, there's a lot of noise in the corpus, some sentence pairs are not related to bias mitigation, they're only style or grammar correction editions, but they're marked as biased. A second problem occurs in the mechanism of mitigation. Many times, it is necessary to make more than one correction in the sentence to neutralize it, a fact that was not initially contemplated. Therefore, the authors proposed a new corpus to provide a solution to these problems, the WikiBias corpus.

This resource has a fine labeling, indicating what type of bias is in each example: framing, epistemological or demographic. In addition to Wikipedia-based corpora, other resources have been created in recent years that focus on other domains, especially news. The MBIC (A Media Bias Annotation Dataset Including Annotator Characteristics) consists of 1700 sentences belonging to (Spinde et al., 2021) press news. The main feature of this corpus is the detailed information about the annotators of the corpus, so that this can help in bias detection. BABE (Bias Annotations By Experts) (Spinde et al., 2022) is a news corpus that consists of 3,700 sentences, 1,700 from MBIC (SG1) and an 2,000 additional texts (SG2). The texts, containing controversial topics, were extracted from 14 US news platforms from January 2017 to June 2020. For each sentence, the BABE corpus indicates the political posture, if the sentence is biased, and which words introduce this bias. In the last years, as part of a CLEF laboratory, the CheckThat! (Barrón-Cedeño et al., 2024) lab has been proposed. Task 2 of this lab aims to determine whether a sentence is subjective or not, and build their corpus, comprised of news sentences in English and Italian about politics, COVID-19, civil rights, and economy. It is worth noting that the annotators considered the quotations to be objective since they are not written by the author, as well as the emotions since they cannot be refuted (Ruggeri et al., 2023).

Regarding the methods of detection, in recent years, transformers have represented the state of the art in this field of study. Spinde et al. (2022) compares the performance of several models in the corpus BABE, reaching a highest result of F1=0.804 with BERT + distant. Raza et al. (2022) obtained an F1 of 0.75 with DistilBert. From the generative perspective, a lot of research has been carried out in order to detect and analyze LLM generated biased content (Fan et al., 2024; Hada et al., 2023). Lin proposes strategies to debias an LLM as well as to better understand biased answers (Lin et al., 2024).

3 Experiments and results

We test the performance of different models for bias detection. Our experiments include classic ML models trained with linguistic features, fine-tuned Transformers, and instruction-tuned LLMs. We used the DBias Python package (Raza et al., 2022), a Transformer based classifier, to generate a baseline.

3.1 Datasets to compare

Wikibias constitutes the primary corpus and is divided in three subsets: train, test, and validation sets. This corpus addresses general topics by drawing upon Wikipedia articles, as detailed in the section 2. With this training and validation sets, the models described below were developed, with the exception of section 3.4. The distribution of the classes of each of the sets used can be seen in the

Table 1.

To test the various models, three datasets were utilized: the Wikibias test set, the SG2 set from the BABE corpus and dev_test from CheckThat!.

The SG2 set from BABE was selected for use in this study due to the fact that the labels in this set were peer-reviewed, in contrast to the MBIC set, which was crowdsourced.

Furthermore, although both the SG2 set and the CheckThat! are news-based, they have been annotated under different agreements , which makes it appropriate to evaluate the models of the present study on both of them.

Corpus	**Bias**	**No-Bias**
WikiBias		
train	1975	3051
validation	403	663
test	784	1314
SG2	973	864
CheckThat!	532	298

Table 1: Distribution of classes in each corpus

3.2 ML models with features

For this approach, we used the following training characteristics: a) Sentence BERT (Reimers and Gurevych, 2019) similarity, b) sentiment analysis, based on the python package pysentimiento (Pérez et al., 2023), and c) the number of adjectives, adverbs and total words contained in the biased lexicons reported or collected by Recasens (Recasens et al., 2013).

Using these features, the following models were trained: Logistic Regression , Support Vector Classification (SVC), Support Vector Regression and Naive Bayes . Also we calculate the percentage of sentences in each class and incorporate the class-weight parameter into the models to address the issue of imbalance classes.

We took the best performing model (SVC, the rest of the models had an F1 value of 0.59 on average) and tested it with data from the other two mentioned corpora (SG2, CheckThat!). Results are shown in Table 5, in the SVC section.

The results show that the performance obtained in WikiBias is maintained in CheckThat!, and even improves when tested with SG2.

3.3 Transformers

We used DBias (Raza et al., 2022) to implement the first experiments with Transformers. DBias is a Python library that uses DistilBERT as a binary classifier for bias detection. The results serve as a baseline for our Transformer-based experiments.

In a second experiment. we implemented DistilRoberta as per standard Transformer usage. We passed all the sentences of the WikiBias corpus without any preprocessing through the pretrained model in order to fine-tune.

In a third experiment, we modified the input. Instead of one sentence, two sentences were given: the first one being the sentence obtained in the training corpus, the second one a masked version of it. We verified which words in the original sentence appear in Recasens' biased lexicon (Recasens et al., 2013). Those words were switched for the PBias word. Based on this new input, another set of fine-tuning and testing was carried out. Table 2 shows the results of these three experiments applied to the three corpora.

Model	**Acc**	**Prec**	**Rec**	**F1**
DBias				
WikiBias	0.54	0.65	0.54	0.57
SG2	0.667	0.67	0.66	0.66
CheckThat!	0.57	0.57	0.57	0.56
DestilRoberta				
WikiBias	0.72	0.63	0.61	0.62
SG2	0.69	0.75	0.59	0.66
CheckThat!	0.64	0.66	0.62	0.64
DestilRoberta sentence + mask				
WikiBias	0.61	0.59	0.61	**0.65**
SG2	0.70	0.66	0.85	**0.75**
CheckThat!	0.63	0.60	0.89	**0.71**

Table 2: Different experiments with Transformers applied to the corpora WikiBias, SG2, CheckThat!.

Notice that although the DBias package reports an F1 value of *0.75* on the MBIC (Raza et al., 2022), it does not perform equally well when tested on different corpus. The results are just slightly above a random classifier.

Moreover, the masked sentence approach proved to be the best methodology in this case. The use of carefully masked sentences that indicate word positions (and in a way word types) susceptible to bias, helped the model perform efficiently in all of the test scenarios. This can be seen in the increase of the F1 value by almost a decimal point in certain cases. Compared to the classic ML models the pretrained approach surpasses SVC.

Upon analyzing the previously reported performance, a follow up round of experiments was carried out. An examination of the instances in which the models exhibited errors revealed that these mainly corresponded to instances of epistemological bias. Thus, we ran an experiment in which these sentences were omitted. Additionally, the weight of the classes was incorporated into the loss function to address the imbalance of classes.

Inspired by this modification of training classes, we fine-tuned DistilRoberta 3 more times: first omitting epistemological bias during training, second using only epistemological bias, third using only framing bias. We decided to omit training only with demographic bias due to a lack of data for this final category. Table 3 describes the results of the second round of experiments.

	Acc	**Prec**	**Rec**	**F1**
Only framing and demographic				
WikiBias	0.70	0.67	0.69	0.68
SG2	0.63	0.64	0.63	0.63
CheckThat!	0.64	0.63	0.64	**0.64**
Only epistemological				
WikiBias	0.68	0.53	0.56	0.51
SG2	0.62	0.62	0.62	0.62
CheckThat!	0.53	0.54	0.54	0.53
Only framing				
WikiBias	0.70	0.68	0.70	**0.69**
SG2	0.65	0.66	0.65	**0.65**
CheckThat!	0.62	0.63	0.63	0.62

Table 3: Results of removal some biased sentences. The best results for each corpus are marked in bold.

It is worth noting that the performance in the WikiBias corpus improved following the elimination of sentences exhibiting epistemological biases. This suggests that this type of bias is harder to classify than the other two.

Other experiments were carried out, such as an ensemble of the SVM with DestilRoberta, in which the epistemological biases were also removed. Scores are not reported since this hybrid model did not improve previously shown results.

3.4 LLMs

Most of bias related research with LLMs focuses on detecting when an LLM produces a biased answer, nonetheless, for these experiments we focused on having LLMs classify sentences in order to detect bias on their own. State-of-the-art work shows that a structured Clue and Reasoning approach (Sun et al., 2023) has worked with widely used LLMs such as GPT3, for classification tasks. Due to token limits in newer models as well as a lack of computational resources this longer approach could not be replicated. Instead simpler zero-shot, one-shot, and few-shot learning variations were carried out for the classifying problem. Two models were evaluated using the WikiBias corpus. The selected models are Google's Gemma (Gemma Team, 2024) and OpenAI's GPT-4o (OpenAI, 2024).

For the zero-shot experiments a single instruction to determine if a sentence is biased was given, no information regarding the nature of bias was added. In the one-shot and few-shot cases the example sentences used were extracted solely from the WikiBias corpus. For the one-shot variation, the extracted sentence was initially chosen at random, but after a few experiments a framing bias sentence proved best, more info regarding prompts can be seen in Appendix A. For the few-shot variation one of each class was presented. Gemma's prompt had to be modified with start and end of turn tokens, where as for GPT the prompt was divided in three roles: system, user, and assistant.

<start_of_turn> user I want you to analyze this sentence {i}[2]*. Based on the following biased sentence I want you to determine if the previous sentence is biased. Example sentence: In 1995 he narrated the excellent TV miniseries Watergate. Think and reason before responding. Answer with a 0 or 1. 0 if there is no bias. 1 if there is.<end_of_turn>*.

For GPT we have the following:

System: *"You detect bias in sentences."*,

User: *f"Determine if the following sentence is biased or not. Answer only with 0 and 1. 0 if it isn't biased, 1 if it is. The sentence is {i}."*

Assistant: *"The following is an example of a biased sentence: in 1995 he narrated the excellent TV miniseries Watergate."*.

Table 4 shows each model's evaluation divided by the type learning variations on WikiBias corpus sentences, and Table 5 illustrates that the most optimal model is the DestilRoberta sentence + mask, which employs the three types of biases.

[2] It was implemented using f-strings, hence the {, }.

	Acc	Prec	Rec	F1
Gemma				
Zero-Shot	0.5	0.52	0.52	0.5
One-Shot	0.64	0.63	0.62	**0.62**
Few-Shot	0.46	0.22	0.22	0.22
GPT4o				
Zero-Shot	0.4	0.46	0.48	0.36
One-Shot	0.6	0.55	0.51	0.4
Few-Shot	0.38	0.4	0.41	0.34

Table 4: LLM for bias classification on WikiBias.

	Acc	Prec	Rec	F1
SVC				
WikiBias	0.66	0.61	0.60	0.60
SG2	0.63	0.64	0.64	0.63
CheckThat!	0.59	0.59	0.59	0.59
DestilRoberta sentence + mask				
WikiBias	0.61	0.59	0.61	**0.65**
SG2	0.70	0.66	0.85	**0.75**
CheckThat!	0.63	0.60	0.89	**0.71**
Gemma One-Shot				
WikiBias	0.64	0.63	0.62	0.62
SG2	0.51	0.34	0.34	0.32
CheckThat!	0.51	0.5	0.5	0.49

Table 5: The best results of each approach. The most favorable outcomes for each corpus are presented in bold.

4 Conclusion

Despite bias detection still being a challenging task in NLP, models trained on the WikiBias corpus are capable of detecting bias in news corpora such as SG2 and CheckThat!. These results indicate that the WikiBias corpus is a good resource for general bias detection, as it already contains more subtle biases. This is probably due to the fact that the goal of Wikipedia articles is to provide knowledge in an unaltered form. This presents inherent differences when compared to various media outlets that talk from a particular perspective and not only report hard facts. A fine grained analysis shows that epistemological bias is more challenging to identify, as it is often introduced through the use of frequent words such as "is," "many," and "so," which makes it dependent on the context of the discourse.

Analyzing results from the one-shot instance and the fine-tuned encoder models, we believe that framing bias represents a more recognizable form of bias for Transformer based methods. Both LLMs and Encoders perform at their best when their fine-tuning or instruction tuning is based on this type of bias. This could be due to the lexical nature of framing bias where adding one or two words instigate said bias.

Finally, we observed that simple instruction-tuned LLMs are not efficient for this task, barely reaching scores obtained by Encoders or classic ML models. Surprisingly few-shot learning was the worst performing instance of an LLM implementation. We theorize that having examples from various classes of bias, particularly without an explanation of each class, hinders the model since a lexical pattern of bias can't be generalized. Another factor might be the token related, adding two additional sentences might push the instruction prompt beyond an adequate amount of tokens.

5 Limitations

Detecting bias in sentences is a challenging task in Natural Language Processing (NLP). Biases can exist at various linguistic levels and often lack clear lexical representation. Among the three main types of bias—epistemological, framing, and demographic—epistemological biases are particularly difficult to detect and address, both for humans and computational algorithms. This difficulty is also evident in this study, as the different methods introduced in the paper fail to identify these biases effectively.

Because biased sentences can be hard for humans to distinguish, labeling also carries a degree of subjectivity. Some corpora include sociodemographic information about the annotators, providing additional information to the systems and algorithms so they can learn from the provided examples. However, this is not the case in our experiments, making the task even more challenging due to the inherent subjectivity.

Moreover, the task presents an additional layer of difficulty. The algorithms and techniques used in the experiments may already be biased. Transformers like DistilRoBERTa, for example, are trained on a large amount of biased data, which means that biases are inherently embedded in these models.

Acknowledgments

This research was funded by CONAHCYT (CF-2023-G-64) and PAPIIT project IT100822. G.B.E. is supported by a grant for the requalification of the Spanish university system from the Ministry of

Universities of the Government of Spain, financed by the European Union, NextGeneration EU (María Zambrano program, Universitat de Barcelona).

Karla Salas-Jimenez (CVU: 1291359), Francisco López-Ponce (CVU: 2045472), and Sergio-Luis Ojeda-Trueba (CVU: 2013549) thank the CONAHCYT graduate degree scholarship program.

References

Alberto Barrón-Cedeño, Firoj Alam, Tanmoy Chakraborty, Tamer Elsayed, Preslav Nakov, Piotr Przybyła, Julia Maria Struß, Fatima Haouari, Maram Hasanain, Federico Ruggeri, Xingyi Song, and Reem Suwaileh. 2024. The CLEF-2024 CheckThat! Lab: Check-Worthiness, Subjectivity, Persuasion, Roles, Authorities, and Adversarial Robustness. In *Advances in Information Retrieval*, pages 449–458, Cham. Springer Nature Switzerland.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Zhiting Fan, Ruizhe Chen, Ruiling Xu, and Zuozhu Liu. 2024. Biasalert: A plug-and-play tool for social bias detection in llms. *Preprint*, arXiv:2407.10241.

Gemma Team. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

Rishav Hada, Agrima Seth, Harshita Diddee, and Kalika Bali. 2023. "fifty shades of bias": Normative ratings of gender bias in GPT generated English text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1862–1876, Singapore. Association for Computational Linguistics.

Luyang Lin, Lingzhi Wang, Jinsong Guo, and Kam-Fai Wong. 2024. Investigating bias in llm-based bias detection: Disparities between llms and human perception. *Preprint*, arXiv:2403.14896.

Aurélien Max and Guillaume Wisniewski. 2022. Mining naturally-occurring corrections and paraphrases from wikipedia's revision history. *Preprint*, arXiv:2202.12575.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Reid Pryzant, Richard Diehl Martinez, Nathan Dass, Sadao Kurohashi, Dan Jurafsky, and Diyi Yang. 2020. Automatically neutralizing subjective bias in text. In *AAAI*.

Juan Manuel Pérez, Mariela Rajngewerc, Juan Carlos Giudici, Damián A. Furman, Franco Luque, Laura Alonso Alemany, and María Vanina Martínez. 2023. pysentimiento: A python toolkit for opinion mining and social nlp tasks. *Preprint*, arXiv:2106.09462.

Shaina Raza, Deepak John Reji, and Chen Ding. 2022. Dbias: detecting biases and ensuring fairness in news articles. *International Journal of Data Science and Analytics*, pages 1–21.

Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. 2013. Linguistic models for analyzing and detecting biased language. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1650–1659.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Federico Ruggeri, Francesco Antici, Andrea Galassi, Katerina Korre, Arianna Muti, and Alberto Barrón-Cedeño. 2023. On the definition of prescriptive annotation guidelines for language-agnostic subjectivity detection. *Text2Story@ ECIR*, 3370:103–111.

T. Spinde, L. Rudnitckaia, K. Sinha, F. Hamborg, B. Gipp, and K. Donnay. 2021. Mbic – a media bias annotation dataset including annotator characteristics. *Preprint*, arXiv:2105.11910.

Timo Spinde, Manuel Plank, Jan-David Krieger, Terry Ruas, Bela Gipp, and Akiko Aizawa. 2022. Neural media bias detection using distant supervision with babe–bias annotations by experts. *arXiv preprint arXiv:2209.14557*.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8990–9005, Singapore. Association for Computational Linguistics.

Fabio Massimo Zanzotto and Marco Pennacchiotti. 2010. Expanding textual entailment corpora fromWikipedia using co-training. In *Proceedings of the 2nd Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, pages 28–36, Beijing, China. Coling 2010 Organizing Committee.

Yang Zhong. 2021. *WIKIBIAS: Detecting multi-span subjective biases in language*. Ph.D. thesis, The Ohio State University.

A Prompts

The prompts seen in the article correspond to final prompts used for the reported experiments. Prompt engineering had to be carried out before actually obtaining said prompts, mainly working our way up from very simple instructions, sometimes even avoiding start of turn tokens. The following list includes the prompts previously used for Gemma. The brackets correspond to the f-string implementation.

1) Tell me if this is biased: {i}.

2) You detect bias in sentences. Is this sentence biased? {i}.

3) Determine if the following sentence is biased: {i}.

The following correspond to GPT prompts.

1) **System**: You detect bias. **User**: Determine if the following sentence is biased: {i}. **Assistant:** NONE.

2) **System**: You detect bias. **User**: The following sentence might contain bias, determine if so. **Assistant**: NONE.

3) **System**: You detect bias in sentences. **User**: Determine if the following sentence is biased or not. **Assistant**: A biased sentence contains a non objective point of view of said sentence.

As can be seen in both lists, initial prompts are very simple, sometimes even omitting that the model is analyzing sentences, as well as start and end of turn tokens or certain roles for GPT. The most interesting type of prompts are those like the second or third prompt for GPT. The second prompt doesn't tell the LLM that the sentence has bias, but it suggests that that is the case. This particularly triggered the LLM to produce mainly positive classifications. The third case contains an ambiguous definition of bias, leading to very inconclusive reasoning.

Similarly behaviour prompts, such as adding instructions to answer with 0 and 1s depending on each classification case, were added after various iterations of experiments. Said values were added in order to facilitate the classification reports.