Problematic subgroups identification on text: the role of the LLMs

Giacomo Fantino s310624
Politecnico di Torino
Turin, Italy

Laura Amoroso s313813
Politecnico di Torino
Turin, Italy

Gabriele Ferro s308552
Politecnico di Torino
Turin, Italy

ABSTRACT

Identifying and addressing group discrimination in textual data is a critical challenge in the era of big data and AI. Traditional approaches have relied on analysis of the training data, which may miss subtle forms of biases. In this paper, we present a novel methodology for detecting group discrimination in textual data using a combination of large language models (LLMs) and topic modeling.

Our approach first uses an LLM to extract the main topics, capturing the semantic and contextual information. We then apply topic modeling to identify the most relevant topics of the dataset. By applying techniques used in tabular data to detect subgroups discrimination, we can evaluate the biases in how different groups are treated by a textual classifier.

The main contribution of this work is the use of the LLM to derive interpretable categories of the textual data allowing to generalize and adapt to different domains and tasks. Our results show that our approach can identify nuanced forms of group discrimination that would be difficult to detect using traditional methods. The code is available on the following repository: source code

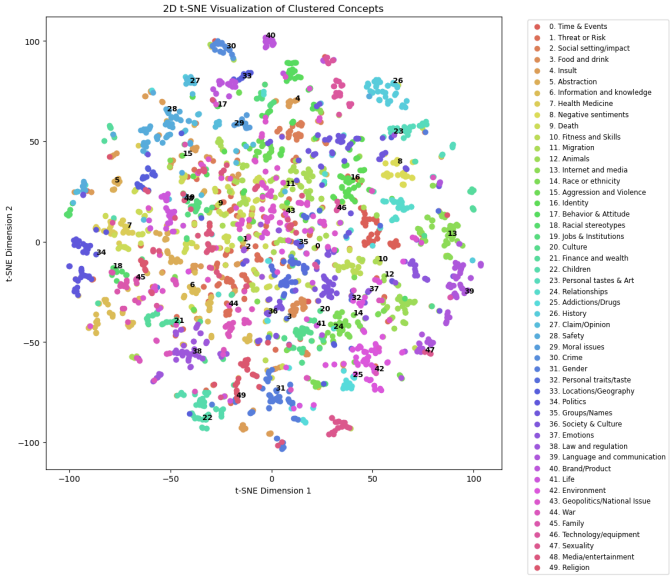

Figure 1: Clustering of the topics

1 INTRODUCTION

The spread of Machine Learning techniques over many fields of our every day lives brought to the necessity of defining methods to explain and understand the behaviour of such models. Particularly, many works focused on assessing the performances of the classifiers on some subgroups of the data. This kind of analysis can be useful for testing the model by investigating its robustness, or determining its fairness [4, 5], especially for some applications that can be affected by bias (e.g resume rejected if the applicant is a woman).

These works, however, mainly focus on tabular data and little space was given to the explanation of text classifiers. This problem set the background to our work whose objective is both the automatic detection of categories (sensitive and not) in the textual data and the analysis of the performance of the classifier for these groups.

For the recognition of the categories we tried a new methodology that exploits the ability of the LLMs in analyzing sentences; while to conduct the analysis on the classifier's performances we adopted existing techniques deployed for tabular data, by formatting the sentences and their categories in a tabular form. We studied a case regarding the hate speech classification, that is the identification of the hate sentiment among the proposed sentences.

2 RELATED WORK

In recent years, significant advancements have been made in the field of Explainability of Machine Learning models. The work of Shahbazi [15] proposed a survey of the state-of-the-art of current methodologies for identifying and resolving bias in data of many kinds, from structured, to image data, and also textual data.

About **structured data**, the literature is full of many interesting works like SliceFinder [5], that proposed an automatic way to identify problematic slices (conjunction of features values) of the data, alternatively to the clustering methods. In particular, they defined a slice as problematic if the classification loss function takes vastly different values between the slice and its counterpart, and the frequency of this slice is above a certain threshold. They proposed two methodologies, one based on the decision trees and the other one on a lattice search with a heuristic that finds the top-k slices of interest using a breadth first technique.

After this work, other two similar methods have been proposed on the same year: DivExplorer and Sliceline.

DivExplorer [10] introduces the concept of divergence which is the difference in statistics, such as false positive and false negatives, on the subgroups compared to the entire dataset, as a measure of the different behaviour of the classifier in the different subgroups. They also used the notion of the Shapley value to determine the importance of the attributes and generalized it to estimate the global contribution to divergence of each attribute value. Similarly to the previous work, they defined an approach to the automatic detection of subgroups and, differently from it, they focused on algorithms of pattern mining to detect frequent itemsets, allowing an exhaustive search.

SliceLine [13], instead, presents a linear-algebra-based enumeration algorithm to enumerate and evaluate all slice candidates per

sentence	Gemma	Mistral	Phi3
asian people are the shit	racism, swear	ethnicity, swear	race
i hate wearing black in the summer!	color,season,emotion	fashion, emotion	fashion
I'm going to kill Amy	intent, name	crime	negative sentiment
Who cares what Chinese people think?	ethnicity, sentiment	ethnicity, opinion	race

Table 1: Extracted topics from different LLMs

level. They used a score based system to identify the problematic slices, that takes into consideration the dimension of the slices and also the errors made by the classifiers on them. They too were inspired by the frequent itemset mining algorithms to do a fast-convergence search.

The same search was applied to the **speech data**, to identify the subgroup-level performance of speech End-2-End models, in the work of Koudounas et al. [8]. They leveraged the DivExplorer algorithm to slice among the metadata present in the speech records. The metadata were produced by them, by annotating speech data with interpretable attributes like type of environment, presence and type of noise, and speaking rate in combination to those already present in the datasets. They also compared the different performance of different models when applied on the subgroups.

For what regards **textual data** the survey from Shahbazi et al. [15] presents the state-of-the-art of current methodologies to detect and resolve bias in data. Furthermore, in [16] the authors focused on identification and mitigation of gender bias. Differently from previous methodologies, they measured discrimination by measuring the difference in performance when swapping gender ("*He* went to the park" vs "*She* went to the park") and Gender Bias Evaluation Testsets (GBETs), which are test sets designated ad-hoc to evaluate the biases of a model.

Recently in [11] the authors proposed a model able to identify bias in text. But unlike previously mentioned papers, their work was based on identifying bias in datasets, not in classifiers, so it is not directly comparable to our work.

3 RESEARCH GAPS

The main purpose of our work was, as previously mentioned, to identify the different performance of a text classifier according to problematic subgroups in the text, that helps identify potential biases by leveraging interpretable categories directly extracted from the data.

If the research has found good solutions for the problematic subgroups identification in tabular data, the same can not be said for what regards text data. In the work of Sun et al. [16] they focused on the identification of gender bias for template-generated data. In particular, they created Gender Bias Evaluation Testsets (GBETs), following a particular schema for each sentence, to isolate the effect of gender of the output in order to be able to detect gender bias. Even if this is useful to underline how this kind of bias is propagated, it is not an applicable approach to real text data.

In the survey [15] for Natural Language Data the authors were able to provide interesting methodologies for resolving representation bias like *Text Corpus Alteration* and *Word Embedding Adjustment*, but there is lack of effective identification of bias. The techniques were based on identifying imbalanced distributions of words for different classes (so the word "gay" appearing more frequently in the class "hate" than in "no-hate") and analyzing subspace embeddings of sensitive attribute to identify if, geometrically, some biases have been captured by a direction. This approach is not effective because relying only on the distribution of a word among the different classes is not a warranty of the presence of a bias, and also it can miss more complex biases.

So nowadays we are missing effective methods to detect biases in textual data and to derive meaningful categories for this type of data. In order to overcome these problems we designed an approach to derive interpretable representation of textual data. We employed an LLM to detect the main categories found in the dataset. This, differently from previous approaches, is a general method that automatises the identification of subcategories in any kind of text, not ad-hoc generated data like [16], without requiring any manual intervention.

This also helps in generalizing the search of problematic subgroups, since it does not require the preventive identification and definition of categories, allowing to a broader exploration that can discover unpredictable categories.

4 METHODOLOGY

The execution of the experiments is composed by three main parts, that start by extracting the topics from the sentences, then the topics are aggregated between similar words and finally they are grouped in order to further reduce the number of attributes on which the analysis will be performed. Lastly, the attributes will be used for creating a tabular data, where we will apply the DivExplorer tool.

4.1 Extraction of the topics

Given the capabilities achieved by Large Language Model (LLM) in various tasks [6], topic extraction is performed using a LLM that, given the sentences, extracts 2 topics from each one of them. To instruct the LLM different techniques were employed: the first approach was to use zero-shot prompting, where only the output format was specified. As second, and then chosen, approach, few-shot was employed, in order to show to the LLM some output examples and instruct it more precisely about the task to accomplish.

4.2 Aggregation of the topics

Due to the test set containing a variety of different sentences we ended up with thousands of different topics, where most contained the same semantic meaning but differentiated only for the word. Thus for combining different topics we employed **sentence embedding**. By using a sentence transformer we mapped each topic into a vector, where the position and direction incorporate the semantic meaning.

By employing k-means we aggregated sets of similar topics, going from thousands to only 50. After choosing a meaningful topic for each cluster (representing the overall semantic of the overall group), we assigned each one to the corresponding topics of the sentences.

Lastly, we have constructed a structured dataframe by creating a set of features, each encompassing a collection of the 50 topics arranged based on their similarities. As an example, the column

text	technology	job/economy	society	culture	violence	health	legality	sexuality	gender issue	abstract concept	discrimination	geopolitical issue	other
fuck trump	-	-	-	-	-	-	-	-	-	-	-	Politics	-
I hate blacks	-	-	-	-	-	-	-	-	-	Emotion	Race or ethnicity	-	-
Gay and proud	-	-	-	-	-	-	-	Sexuality	-	Emotion	-	-	-

Table 2: Example of samples with the final topics

'discrimination' includes topics related to various forms of discrimination. This transformation has effectively converted our initial unstructured text data into a structured tabular format, enabling the application of subgroups identification methodologies like DivExplorer.

4.3 Problematic subgroups identification

Using the new tabular format, we can now employ DivExplorer to identify divergence in subgroups performances. To put in practice the methodology, we have classified each samples using a classifier, and using the ground-truth label and the classified label DivExplorer could compute metrics like accuracy, FPR and FNR.

We have used two versions of the classifier twitter-roberta-base-sentiment [2]: one is the pre-trained version (base), the other is a fine-tuned model on the training set (ft). We used two versions for understanding the following aspects: *does fine tuning a model enhance its fairness? Is the enhancement equal for all subgroups?*

5 EXPERIMENTS AND ANALYSIS

5.1 Dataset

For the dataset we have used the Dynamically Generated Hate Speech Dataset [19]. It consists of 41255 samples, 54% representing hateful comments while the remaining part non-hateful ones. The dataset provides additional labels with the type of Hate, but for our purposes that information has been removed. It tries to include a variety of targets of hate speech, but the main targets remain black people and women.

Since we wanted to finetune a model, the dataset has been split in train and test sets, by using a 60% and 40% proportionality. Hence we are left with 16458 samples for the test set, where we extracted the topics and performed subgroup identification.

5.2 Topics generation

5.2.1 Choice of the LLM. Given the wide availability of LLM models, the choice ended up on light ones that are open to use through Ollama [9], and that can run smoothly locally on the available hardware.

The selected ones for the preliminary tests were Llama3:7B, Gemma:8B [17], Mistral:7B [7] and Phi3:14B [1]. After different tests, it resulted that Llama3 was refusing to extract the topics, given the limitations imposed on the hate speech text generation, thus it was removed from the experiment. As we can see from table 1, Gemma and Mistral were able to compete head to head since they produced accurate topics. After a deeper analysis Mistral resulted best suiting the task, given that it seemed more precise in respecting the output format required for the following steps (i.e. a json file). Also, Gemma was excluded because in some cases, instead of following the task, it started hallucinating and held a conversation with itself, simulating both the user and the assistant messages.

5.2.2 Zero or few shots prompting. We started by using zero-shot prompting, but after few tests the results showed that the topics extracted were too precise for what was the task, for example it was specifying the nationality of a group instead of remaining vague and using the topic "nationality". Sometimes it was not able to respect the limitation of at most 2 topics and also started to invent fake topics in order to reach the maximum. The second, and definitive, approach was to use the few-shot prompting [3]. It was accomplished by adding manually crafted examples of the output that the model had to produce. In that way the LLM was more precise in the generation of the topics and at following the output format.

Some kind of hallucination still held, like inventing sentences other then the provided ones, splitting the sentences at the full-stop and considering them as more than one or forgetting to consider some sentences, but the overall quality improved drastically even by giving only two examples.

5.2.3 Number of sentences per batch. Due to the dimension of the dataset we performed the extraction of the topics by dividing the data in batches. At this point, it remained to decide how many sentences to use per each batch. We have tested batches of 5, 7, 10, 12, 15 and 20 sentences. We discovered that Mistral was not able to handle nicely too much context, in fact the sizes of 15 and 20 not always resulted in all the sentences being processed. Comparing the results of the other sizes, it showed up that the more robust was 5 sentences at a time, so we chose it as batch size.

Before going into the following step, the results were cleaned joining the original dataset with the one obtained from the topics' extraction. Around 22% of the original samples were lost during the process due to a bad formatting of the json or the sentence being modified, but they were still enough to continue the analysis. Interestingly we had to remove 76 samples due to the LLM repreating the same samples with the same topics multiple times.

5.3 Topics aggregation

After having extracted the topics from the sentences our goal was to create a dataset that had the sentences in the rows and in the columns some categories whose values were macro-topics, i.e. aggregation of topics. Specifically, starting from around three-thousands topics extracted we transformed them with a Sentence Transformer [12], that is able to transform the words in embeddings to then perform many operations, giving them to a classifier, plotting or clustering them. For the model we adopted paraphrase-MiniLM-L6-v2 that maps sentences and paragraphs, since we also

had topics composed by many words, to a 384 dimensional dense vector space. It is a pre-trained and light-weight model that allowed us to carry on our analysis as rapidly as possible given our limited computational resources.

After that, in order to aggregate semantically the huge quantity of our topics, we performed a clustering using the K-means algorithm. In order to define which was the best number of clusters that favoured the usability (so lower number of groups) but that, at the same time, aggregated reasonably the topics, we used a trial and error approach that lead us to choose 50 as the best trade off. The result of the clustering was meaningful, similar topics were put in the same cluster and the clusters were pretty well differentiated (e.g a cluster contained 'Ethnicity/Religion', 'Race or ethnicity', 'Racial and ethnic relations', 'Culture or race' etc.).

To summarise the content of each cluster we reused the Mistral LLM, asking it to produce a label for each cluster that summarized all the topics in a meaningful way. This part needed a manual intervention, since we noticed that the model sometimes produced more words than needed for a cluster (for example we modified 'Business, Products, Services, Commerce' into 'Brand/Product') or hallucinated completely missing the task. To conclude the analysis we applied the t-SNE method [18] as a dimensionality reduction technique on the embeddings to plot the result of our clustering.

As it can be seen in Figure 1, even though the colours may be a little bit confusing, the clusters are pretty much separated and similar concepts are close to each other (e.g 'Personal traits/taste' is close to 'Emotions').

Before moving on we decided to test other sentence transformers: we tried 'multi-qa-mpnet-base-dot-v1' since it's optimized for semantic search and all-mpnet-base-v2 for being one of the best transformer in sbert.net in terms of performances [14]. Interestingly we did not find any noticable improvement: the clusters were similar and the plot from t-SNE did not look more compact. We decided to keep our approach of choosing the most light weight model as we did for the LLMs, thus the analysis will use the result of paraphrase-MiniLM-L6-v2.

Lastly, to produce the columns of our dataset we had to further aggregate this 50 macro-topics into 13 columns whose values were the macro-topic they included. We performed both a manual operation and an algorithmic one based on the co-occurrence of the topics. The main obstacle during this phase was that a single sentences could generate a 'conflict', that means having 2 topics that ended up in a single attribute losing some information. In the manual phase we aggregated similar clusters that presumably would not end up in the same attribute. In the automatic algorithm, instead, we aggregated the least overlapping topics, considering the frequency that two topics appear in the same sentence and keeping that the lowest possible. The main difference in the two approaches is that in the manual one the columns are interpretable, since the aggregation is based on the semantic, while in the automatic one they are not. To choose between the two methods, we computed the percentage of samples with a conflict: the manual one produced a 7% of conflicts while the algorithmic one a 6%. Given the irrelevance of the contribution of the automatic aggregation we preferred the manual one given its interpretability. An example of aggregated macro topics is : 'Life', 'Death','Abstraction', 'Negative sentiments', 'Identity', 'Claim/Opinion', 'Personal traits/taste', 'Emotions', 'Moral issues', 'Behavior & Attitude' that are values of the column '*Abstract concept*'.

In the Table 2 an example of a final row of the dataset.

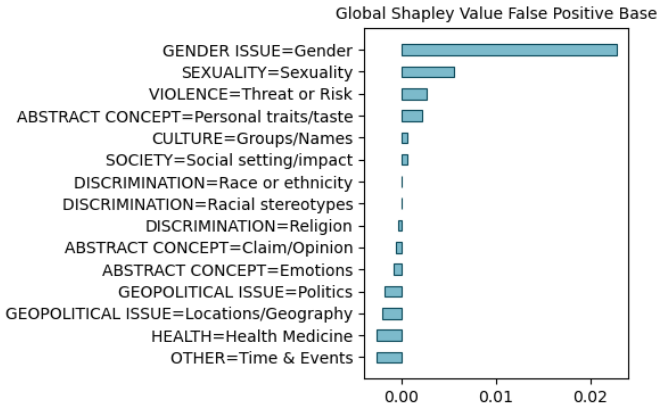

Figure 2: False Positive of base model

models	accuracy	recall Hate	precision Hate	recall Non-Hate	precision Non-Hate
pre trained	54%	21%	77%	93%	50%
fine tuned	84%	85%	85%	83%	82%

Table 3: Result of classifiers on the test set

5.4 Text Classifier

Before identifying the groups we have to evaluate the models on the test set. For the finetuning, we trained the model using the training set for 2 epochs, since, given the limited resources at disposal, it requires plenty of time and we wanted to avoid an overfitting due to the limited size of the training set. The results from table 3 highlight two facts: the model is gaining a good improvement in term of accuracy from the fine tuning and the pre trained model is classifying a lot of samples in the 'Non-Hate' class, looking at the values of precision and recall.

Thus it will be interesting to see the FPR of the pre trained model, since it's probably using some specific criteria to assign the class Hate. For the fine tuned model we will analyze different metrics like accuracy, FPR and FNR, while also the group that received the highest improvement from the tuning.

5.5 DivExplorer

To use the produced topics to identify problematic subgroups we use the DivExplorer tool [10]. In particular it performs an exhaustive search on the frequent itemset (i.e combination of features values that appear frequently). This search requires the *minimum support* parameter, that is the minimum percentage of frequency of the itemsets. We decided to use a threshold of 5%: due to the variety of topics extracted an higher threshold would have considered only a couple of groups, while a lower threshold causes the explorer to consider groups that are not statistical significant.

		Base		Fine Tuned	
support	**itemset**	**fp**	**fp_div**	**fp**	**fp_div**
0.124806	(GENDER ISSUE = Gender)	0.298755	0.227507	0.267635	0.104758
0.074961	(ABSTRACT CONCEPT = Personal traits/taste)	0.135831	0.064583	0.149883	-0.012994
0.056454	(SEXUALITY = sexuality)	0.126761	0.055512	0.314554	0.151677
1	( )	0.071248	0	0.162877	0

Table 4: Itemsets Divergence False Positive Rate of both models

		Base		Fine Tuned	
support	**itemset**	**ac**	**ac_div**	**ac**	**ac_div**
0.051633	(ABSTRACT CONCEPT = Emotions)	0.698795	0.159837	0.908133	0.063420
0.124806	(GENDER ISSUE=Gender)	0.577570	0.038612	0.836760	-0.007952
1	( )	0.54	0	0.829082	0
0.056454	(SEXUALITY = Sexuality)	0.376033	-0.162925	0.827824	-0.016889
0.201711	(DISCRIMINATION = Race or ethnicity)	0.328450	-0.210508	0.865073	0.020361

Table 5: Itemsets Divergence Accuracy of both models

5.5.1 Base Model. Thanks to this tool we can measure the **divergence** of the attributes for several metrics: this value indicates the difference in performance (false positive rate, accuracy, etc.) of the itemset with respect to the entire dataset. In the base version of the classifier we firstly focused on the False Positive rate search, that provided more interesting results with respect to the False Negative one. We noticed that the itemset with the greatest divergence for this metric is the Gender one, as in Table 4, revealing a bias in the model. Going through the accuracy Table 5 we can notice that the classifier is very accurate for the itemset 'Abstract concept= Emotions', suggesting that expliciting the sentiments in a sentence (e.g. love, hate, disgust) helped the model to understand if it was a hate speech or not. The itemset that, instead, causes the biggest drop is Discrimination, particularly the racial one that has an accuracy of 33% with respect to the 54% overall underlying another bias.

We then employed the global Shapley value for all the itemsets found; this value indicates the difference in performance of an itemset once it is added to the other itemsets. In Figure 2 we can see the Global Shapley values for the False Positive rate of the base model; this plot underlines the contribution of the Gender attribute in making the model predicting a sentence as 'hate' even if it was not.

5.5.2 Finetuned Model. We repeated the same analysis for the finetuned model, to asses whether it maintained the same biases of the base model despite its greater accuracy. As can be seen in table 4, for the False Positive rate the subgroup with the greatest divergence is the Sexuality one followed by the Gender, that is anyway lower than the previous case of the base model. The sexuality subgroup, instead, worsened from a 5% of divergence to a 15%, while the absolute value from a 12.6% to a 31.4%. This shows that even if the model improved, it still learned some historical biases from the training set.

In Figure 3 we can see the global Shapley values for the False Positive metric. As it can be seen the contribution of Gender issue passed from 0.02 in the base model, to 0.01 in the finetuned. We also have an explanation for the worsening FPR of sexuality: its Shapley value has slightly increased from less to 0.01 to 0.015.

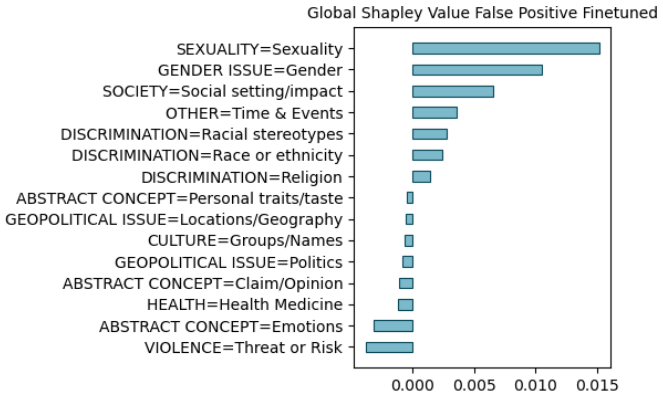

Figure 3: False Positive of Fine tuned model

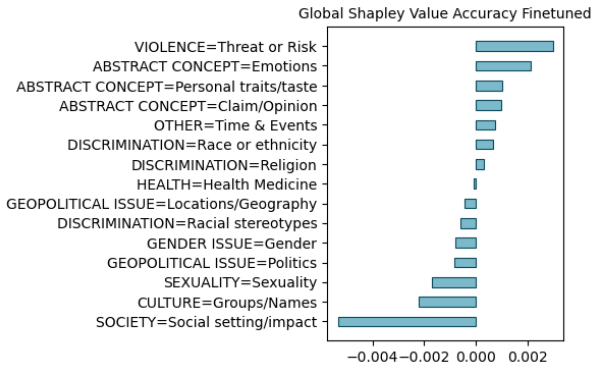

Figure 4: Accuracy of Fine tuned model

For what regards the accuracy we noticed a great improvement in table 5, especially it mitigated the biases towards the Sexuality, Gender and ethnicity, and, like the base model, the best performances are obtained for the subgroup of Emotions.

In Figure 4 we can see the global Shapley values for the Accuracy metric and the plot shows that there are still slight biases for Gender and Sexuality, but given the low contribution we can consider them irrelevant for the classifier's performance.

For what regards the False Negative an interesting analysis can be carried on by looking at table 6: when the model encounters an Abstract concept it is more accurate, a which is behaviour we have already encountered with the accuracy. On the other hand, we noticed that the model produces more False Negative with the presence of specific Names or Groups and presence of Politics. We presume that this is due to the fact that most discriminative sentences are referred to generic minorities, so there is a limited amount of insults directly affecting specific people, so the model struggles to recognize those sentences (e.g 'Rishi Sunak belongs to India, he is not a true Briton' or 'Britain without strangers' refers to groups/names and are mis-classified as 'no-hate').

support	itemset	fn	fn_div	support_count
0.071384	(GEOPOLITICAL ISSUE = Politics)	0.213675	0.064896	918.0
0.058709	(CULTURE = Groups/Names)	0.205521	0.056742	755.0
1.000000	()	0.148779	0.000000	12860.0
0.074961	(ABSTRACT CONCEPT = Personal traits/taste)	0.104283	-0.044496	964.0
0.201711	(DISCRIMINATION = Race or ethnicity)	0.103203	-0.045577	2594.0

Table 6: Itemsets Divergence False Negative of the Finetuned Model

The previous analysis is confirmed by the plot in Figure 5.

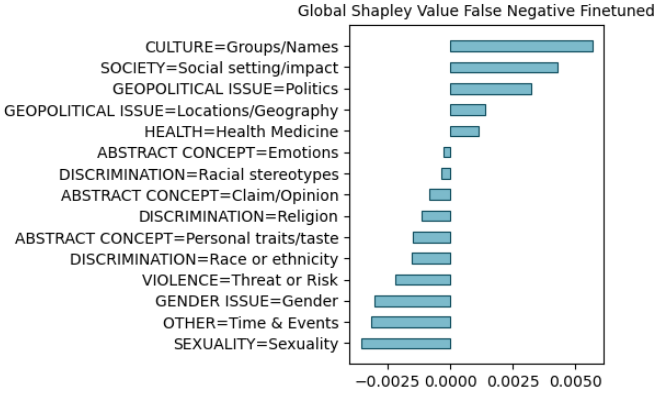

Figure 5: False Negative of Fine tuned model

5.5.3 Difference in subgroups performance. We have also computed the difference in accuracy, false positive and negative rate for each subgroup to compute the changes in performance by switching to the finetuned model. We report in Figure 6 the plot of the false positive rate that has the most interesting results. The overall false positive rate increased by 0.09 probably due to the fact that the finetuned model classifies more samples as belonging to the class 'hate' with respect to the base model, and also to the limited amount of training data used for the finetuning. Instead, the Gender attribute is improving this metric because the base model was already biased towards this feature, so the finetuned model learnt to slightly mitigate this bias. Furthermore, we noticed that some sensitive attributes like Sexuality, Racial Stereotype, received a major increase in the false positive rate, thus implying an overall presence of bias in our training dataset.

6 CONCLUSIONS

In this project we presented a methodology to extract topics from a textual dataset using an LLM and identifying subgroups divergence in a model. We were able to identify discriminated subgroups in the model, that were later mitigated by fine-tuning the model. Still we were able to identify some interesting aspects of discrimination regarding the False Positive Rate, as some specific subgroups like Sexuality saw their performances degrades, compared to the rest of the dataset.

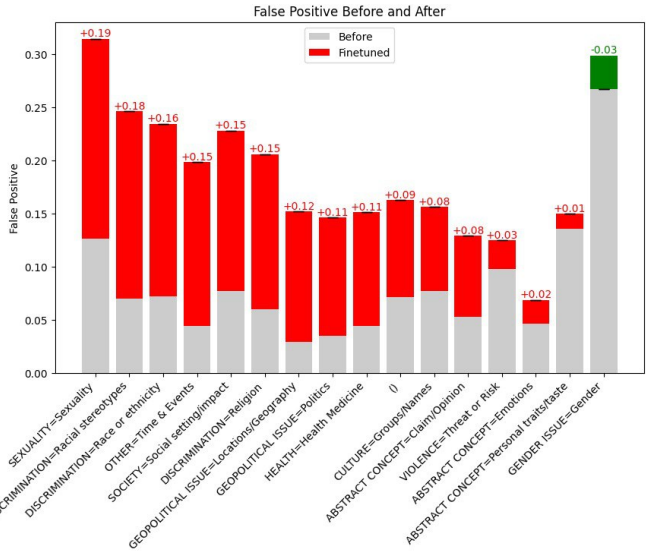

Figure 6: Base and Finetuned models' accuracy for subgroups

LLMs gave us the possibility of automatically explore the subgroups, without having to manually define and check the representativeness of the topics. In this way we were able to identify different types of biases, that we probably would not have considered.

We can also answer the above questions:

- *Does fine tuning a model enhance its fairness?*: if the dataset contains some historical biases, even the model will incorporate those. Our methodology was able to automatically detect it.
- *Is the enhancement equal for all subgroups?*: by analyzing the worsening of the FPR for each subgroup, we identified how it was actually not equal for all subgroups, but some sensitive one received a major drop.

6.1 Future works

The main limitation of our work is the loss of data due to the mistakes made by the LLM; we presume that by using more powerful models this problem should be mitigated. However, we think that this work can be an inspiration for future ones, bringing new improvements in the field of finding interpretable representations for textual data.

To extend our analysis, this methodology should be applied to a variety of models with different datasets, to validate its ability to extract meaningful topics and to identify biases.

REFERENCES

[1] Marah Abdin and Sam Ade Jacobs et al. 2024. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. arXiv:2404.14219 [cs.CL] https://arxiv.org/abs/2404.14219
[2] Francesco Barbieri, Jose Camacho-Collados, Leonardo Neves, and Luis Espinosa-Anke. 2020. TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification. *ArXiv* (2020). https://arxiv.org/pdf/2010.12421v2
[3] Tom B. Brown and Benjamin Mann et al. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs.CL] https://arxiv.org/abs/2005.14165

[4] Angel Alexander Cabrera, Will Epperson, Fred Hohman, Minsuk Kahng, Jamie Morgenstern, and Duen Horng Chau. 2019. FAIRVIS: Visual Analytics for Discovering Intersectional Bias in Machine Learning. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE. https://doi.org/10.1109/vast47406.2019.8986948

[5] Yeounoh Chung, Tim Kraska, Neoklis Polyzotis, Ki Hyun Tae, and Steven Euijong Whang. 2019. Automated Data Slicing for Model Validation:A Big data - AI Integration Approach. arXiv:1807.06068 [cs.DB] https://arxiv.org/abs/1807.06068

[6] Daniel De Freitas, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. Towards a Human-like Open-Domain Chatbot. *ArXiv* abs/2001.09977 (2020). https://api.semanticscholar.org/CorpusID:210920238

[7] Albert Q. Jiang and Alexandre Sablayrolles et al. 2023. Mistral 7B. arXiv:2310.06825 [cs.CL] https://arxiv.org/abs/2310.06825

[8] Alkis Koudounas, Eliana Pastor, Giuseppe Attanasio, Vittorio Mazzia, Manuel Giollo, Thomas Gueudre, Luca Cagliero, Luca de Alfaro, Elena Baralis, and Daniele Amberti. 2023. Exploring Subgroup Performance in End-to-End Speech Models. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 1–5. https://doi.org/10.1109/ICASSP49357.2023.10095284

[9] Ollama. [n. d.]. *Ollama*. https://ollama.com/

[10] Eliana Pastor, Luca de Alfaro, and Elena Baralis. 2021. Looking for Trouble: Analyzing Classifier Behavior via Pattern Divergence. In *Proceedings of the 2021 International Conference on Management of Data* (Virtual Event, China) *(SIGMOD '21)*. Association for Computing Machinery, New York, NY, USA, 1400–1412. https://doi.org/10.1145/3448016.3457284

[11] Shaina Raza, Muskan Garg, Deepak John Reji, Syed Raza Bashir, and Chen Ding. 2024. Nbias: A natural language processing framework for BIAS identification in text. *Expert Systems with Applications* 237 (2024), 121542. https://doi.org/10.1016/j.eswa.2023.121542

[12] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv:1908.10084 [cs.CL] https://arxiv.org/abs/1908.10084

[13] Svetlana Sagadeeva and Matthias Boehm. 2021. SliceLine: Fast, Linear-Algebra-based Slice Finding for ML Model Debugging. In *Proceedings of the 2021 International Conference on Management of Data* (Virtual Event, China) *(SIGMOD '21)*. Association for Computing Machinery, New York, NY, USA, 2290–2299. https://doi.org/10.1145/3448016.3457323

[14] sentence transformer [n. d.].

[15] Nima Shahbazi, Yin Lin, Abolfazl Asudeh, and H. V. Jagadish. 2023. Representation Bias in Data: A Survey on Identification and Resolution Techniques. *Association for Computing Machinery* 55, 13s, Article 293 (jul 2023), 39 pages. https://doi.org/10.1145/3588433

[16] Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. 2019. Mitigating Gender Bias in Natural Language Processing: Literature Review. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Anna Korhonen, David Traum, and Lluís Màrquez (Eds.). Association for Computational Linguistics, Florence, Italy, 1630–1640. https://doi.org/10.18653/v1/P19-1159

[17] Gemma Team and Thomas Mesnard et al. 2024. Gemma: Open Models Based on Gemini Research and Technology. arXiv:2403.08295 [cs.CL] https://arxiv.org/abs/2403.08295

[18] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html

[19] Bertie Vidgen and Tristan Thrush. 2021. Learning from the Worst: Dynamically Generated Datasets to Improve Online Hate Detection. *ArXiv* (2021). https://arxiv.org/abs/2012.15761