Final Project Report

Sumuk Rao
Georgia Institute of Technology
CS 6795 - OMSCS
California, USA
srao326@gatech.edu

***Abstract*—This paper explores the concept of bias in a legal context. The main types of biases explored in this paper are anchoring bias, attentional bias, availability heuristic, and confirmation bias. These biases along with many others are unintentionally introduced by many individuals which creates an unfair environment when dealing with contracts. Furthermore, these biases bleed into large language models and provide lawyers with biased suggestions to edit contracts. This is an computational model based paper which explores the possibility of creating a model to mitigate these biases which get introduced while using large language model based tools in a contract lifecycle management software. This tool utilizes the concepts of Bayes Theorem, Connectionism, and Affective Computing to create neural network, Bayes networks, and sentiment analysis to device a binary classification model. This tool is created in Google Colab and built in Python 3. Based on testing, we conclude that we can build an effective model with 70% accuracy that can detect biases in user-large language model interactions.**

I. Introduction

When you are feeling hungry, you create an account on DoorDash and order meal to be delivered straight to your door. When you are running late for the office, you book an Uber that takes you straight to your meeting. When you want to talk to your loved ones, you open WhatsApp and check in on your loved ones. Just like that, you have signed countless terms and conditions contracts to responsibly use these online services. Contracts and agreements form the basis of every transaction, ranging from simply using an app all the way to purchasing a new home. Given this is a foundational piece of our everyday lives, it is important that we maintain the integrity of these policies and contracts.

In a recent paper, Brooklyn Law School examined the impact of false consensus bias and stated "These studies suggest that judges should take seriously the disagreement of other judges in determining whether contractual language is subject to multiple interpretations. Otherwise, litigants may become unwilling participants in a lottery whose result is determined by the idiosyncratic interpretation of the judge assigned to their case" [5]. This proves that while it may be unintended, legal professionals may not always recognize ambiguity in contract language since they may assume that their interpretation of the language is widely shared. This ambiguity may later pose a risk of unfair outcomes where a plaintiff or a defendant may approach a case with varying interpretations of the same context. This article suggests that definitions in a contract must be concrete and it must explicitly state all interpretations to ensure fair outcomes.

An article written by Boston University examines an impact of Arbitrator's Bias in contract language. It mentions "The arbitrator might be tempted, even subconsciously, to add a sentence to an award that could later be cited in another case. Such an arrière pensée might lead to disparaging or approving some legal authority or argument regularly presented in similar disputes, and thus intended to persuade in a different matter where the arbitrator's firm acts as counsel" [4]. This quote emphasizes that bias may not be intentional but could be a result of subconscious responses to other professional activities. The arbitrator could be involved in other cases, so they may be biased to modify the current contract in a certain way to aid in their other or future cases. This introduces external influence creating unfair outcomes for both the defendant and the plaintiff.

The biases that exist in legal text and person to person interactions eventually bleed into our technology as we utilize real world examples to train large language models. An article written by Stanford mentions "Numerous studies over the last several years, including research from Stanford Law School and Stanford University, have demonstrated that LLMs exhibit racial biases in their responses" [3]. The article further mentions that "regulators might consider requiring companies that deploy AI models to conduct rigorous bias audits, maintain transparency about their AI usage, and ensure compliance with anti-discrimination laws" [3]. It continues to be a problem where tools like ChatGPT and other large language models are utilized to draft and edit contract documents. The user using these tools may be bringing certain biases while using the tools and in turn, the large language model will produce biased results as well. The only way to mitigate this is to identify and avoid the biases that exist in user produced content along with responses from large language models.

As shown from various articles, biases exist in all legal texts and interactions and is now bleeding into the technology we use. To ensure fairness for all individuals, we must explore how we can mitigate these issues in the legal world. This paper explores the question **How can a contract lifecycle management (CLM) system be improved to detect and mitigate cognitive biases from both human interaction and generative AI during the contract negotiation process to ensure fair outcomes?** This specifically looks into how bias is created when users use a large language model tool within a CLM system to write or edit legal documents.

II. Model/Tool Design

I examined my research question by creating a unique intersection between cognitive science and machine learning. I analyzed topics taught in the course videos and the textbook to pick concepts that would be relevant in identifying bias in legal context. Once I finalized my list of concepts, I brainstormed various technologies I can use to map these concepts on a technical landscape. This allowed me to create complex models based on cognitive science to produce accurate data and quality analysis. The topics I examined for this paper are cognitive biases, affective computing, bayesian network, and connectionism.

A. Cognitive Biases

In regards to cognitive biases, I relied on learning this topic through online resources and articles, with a focus on anchoring bias, attentional bias, availability heuristic, and confirmation bias. The main article I used for this learning, through Clevland Clinic, states that anchoring bias is using "pre-existing information or the first piece of information you come across to base your decision" [2]. Attentional bias is when you "pay attention to certain facts while ignoring others" [2]. Availability heuristic is using "any information that you can recall easily and quickly to make a decision" [2]. Confirmation bias is when "you automatically pay attention to things that confirm your own beliefs" [2]. While there are many other types of biases, I focused on these top 4 biases to limit the scope of my research. My aim was to identify if these types of biases could be detected when users are interacting with large language models.

B. Affective Computing

In order to successfully identify these, my first step was understanding the concept of affective computing. This topic was discussed deeply in Chapter 10 of the textbook. The textbook states affective computing "is computing that relates to, arises from, or deliberately influences emotions" [1]. I realized that emotions can play a huge role in creating bias so my goal was to use technology to understand different emotions within text before I identify biases within that text. In my project, I used the TextBlob library to quantify a positive or negative sentiment from a given text and appended that to the semantic embeddings generated by the BERT model. This allowed me to label any given response with a quantifiable sentiment value to further enrich the complex models to improve the chances of detecting bias.

C. Connectionism and Bayesian Network

The complex models used to identify bias relied heavily on the concepts of Connectionism and Bayesian Networks. The first part of this model captures the semantic meaning of the text by using sentence embeddings from the BERT model. These embeddings are then pushed through a feedforward neural network, where it learns the non-linear patterns associated with bias. This was done by using the Keras library to build the neural networks. This contained two hidden layers with 256 and 128 neurons, using the ReLU activation function. I also utilized dropout layers which prevents the model from overfitting. The dropout rates were 30% and 20%. The output consisted of a single neuron which was a probability between 0 and 1 to represent the likelihood of a given statement showing bias. This represents the connectionist paradigm where it mimics the way human cognition adapts based on experience. This concept is covered in great detail in Chapter 7, 8, and other parts of the textbook which let me understand how neural networks process information similar to the brain. To further strengthen this model, I utilized Bayesian Networks to perform explicit probabilistic reasoning. I used the Sklearn library to create discrete clusters using Kmeans. I use these cluster memberships with the Bayesian Network to understand the conditional probability of bias. This is similar to the mind where the brain updates a belief based on probabilistic inference. This concept of Bayes Theorem is further touched upon in Chapter 2 of the textbook. By combining these two concepts, I created a hybrid cognitive architecture to maximize the accuracy in identifying bias.

D. Technology Used

This entire project was completed using a Google Colab Notebook. This platform provides a space to generate code in a step by step basis and also host the code on an online platform for free so it can be shared. This Google Colab Notebook utilized a regular CPU and ran on Python 3. The Pandas and NumPy library were utilized to sort, clean, and store the gathered data. Furthermore, I used the TextBlob library to identify sentiment in the text before using the model. I also utilized the Sentence Transformer library to vectorize the text using BERT. Then I utilized the Keras library to create the neural networks. I also utilized Sklearn library to create discrete clusters for KMeans. These were the main libraries used throughout the project, the results section goes into detail on how these were used and how they provided an output.

E. Data Gathering

After substantial online searching, I realized that it is extremely difficult to find human and large language model interactions labeled with different types of biases. Due to this, I decided to create my own dataset for this project. First, I utilized ChatGPT to generate three different types of contracts: employment hiring contract, sales agreement, and a technology licensing agreement. I then fed each contract into a new ChatGPT agent and asked it to play different roles and create randomly generated user questions and possible LLM responses based on the provided document. For the employment hiring contract, I used the roles of legal counsel and human resources individual. For the sales agreement, I used the roles of legal counsel and sales associate. For the technology licensing agreement, I used the roles of legal counsel and chief technology officer. This provides an accurate representation of the real world since legal counsel is involved in almost all of the contracts in the company, whereas certain individuals of other departments are involved in specific contracts. ChatGPT

generated 500 questions and responses per role, per contract and labeled each question and response as containing bias or not. This allowed me to have 3000 points of data to train and test my model. The limitations of this approach is discussed in a later section.

III. Results

Throughout this project, I not only developed a model to detect bias in a given statement, I also examined relationships between different variables to further understand the data.

A. Bias Transformation Matrix

For each data set provided, there was a simulated question asked by a user, a simulated response from the LLM, bias label of the question asked (input), and a bias label of the response received (output). I used this data to build a Bias Transformation Matrix as seen in Appendix A. My goal here was to identify if there was a relationship between bias existing in the question asked and bias existing in the response received by the LLM. Based on the graph shown, there does not seem to be a high correlation between the bias labels present in the input and the output. Due to this we are unable to form any conclusion for individual categories of biases. The only substantial inference from this graph would be that if a bias exists (any type of bias) within the question of a user, there is at least a 75% probability that there exists a bias in the response of the LLM.

B. Bias in User Question by Role

My next task was to graph the bias labels per role to see if certain individuals produced more of one type of bias than another. The result of this experiment is shown in Appendix B. The results here indicate that all roles show a lower level of confirmation bias than any other bias, where HR individuals show the lowest level at 0 questions showing this bias. Across the graphs, it seems like Anchoring Bias seems to be the most common across all roles. The rest of the types of Biases seem to be evenly spread across all roles. There are few things to note when inferring these results. Further research needs to be done to determine if the two mentioned biases are indeed the most common or least common biases between the roles or if it is simply easier or harder for these types of biases to interpret for computer models.

C. Creating a Model to Identify Different Types of Bias

My initial approach was to create a model to identify a specific type of bias that would exist in a users input question. This model used the user role, simulated user question, along with the bias label for the question. For this, I encoded the labels and vectorized the text to create a logistic regression classifier. I created a 80-20 split to ensure I had enough data points to test the model. The results of the model are shown in Appendix C. This shows that the model was able to accurately predict Anchoring Bias 14% of the time, Attentional Bias 29% of the time, Availability Heuristic 11% of the time, Confirmation Bias 0% of the time, and no bias 21% of the time. Unfortunately, this means that the model performed poorly and is not able to successfully identify specific types of bias given a role and a user question.

D. Binary Classification of Bias

Since the model did not do well in predicting specific types of biases, my next approach involved simplifying the prediction. I modified the bias labels to be either 1 or 0, indicating whether a given text displays bias or not. While keeping the same 80-20 split and using the same simulated user question and role, I fed this through an XGB Classifier and the accuracy of this model can be seen in Appendix D. This model performed much better than the previous model where it was able to correctly identify bias 48% of the time and correctly identify non-bias text 51% of the time. Although these accuracy scores are much lower than ideal, these show promising steps toward creating a good model.

E. Identifying Bias in LLM Responses

Based on the above experiments, I concluded that user questions contained a very small number of words where the model may not do well in identifying bias with just the question itself. For this part of the project, I used all parts of the data point: the role, user's question, bias label of the question, response from the LLM, along with the label of bias for the response from the LLM. Once again, I modified the bias labels so that it was either a 1 or 0. I used one hot encoding to to encode the role column and used the BERT model to create text embeddings for the user questions and LLM responses. I once again used the XGB Classifier but this time also utilized a Neural Network along with the Bayesian Network and combined them using Ensemble Evaluation to identify whether a given data point showed bias. As shown in the screenshot listed in Appendix E, this model performed much better where it was able to correctly identify bias 71% of the time and correctly identify no bias 70% of the time. While I was hoping to reach higher than 80%, this shows great improvement than the initial models created.

IV. Discussion and Conclusion

The variety of results provided above provide many insights of how to analyze bias in a legal context.

A. Correlations in Biases

The first two graphs provide better context on the data that is available. The first graph regarding Bias Transformation Matrix showed us that if the user question to the LLM contained bias, then there is a 75% or more probability that the response from the LLM will contain bias as well. This further proves the article by Stanford mentions above that bias from a user bleeds into large language models which in turn produces biased outputs for the user. This shows the necessity of both auditing and ensuring information fed into large language models are free from bias and flagging and fixing biased information coming out of large language models. This data could be further combined with the graph from Bias in User

Question by Role. By understanding what types of bias each user tends to introduce, we can provide specific training to those individuals to avoid those biases. Furthermore, we can tune models to identify those biases and prevent the users from interacting with large language models if their input text contains those biases. Understanding these trends will help prevent bias in legal documents.

B. Limitation in Understanding Different Biases

Our initial model produced very poor results as shown in Appendix C. This proves that when providing small amounts of text, our model may not perform well in identifying the specific type of bias that exists. This may be a limitation for any type of statistical based model since these types of models perform poorly in understanding semantic meanings of words. The vector based conversions may work well in assigning certain words to bias categories based on its presence in the training dataset, but that same word can show bias in many different ways based on the context it is used in. As of the today, large language models tend to be the only type of models which can also effectively pick up context. Moving forward, we may need to include large language models such as ChatGPT or Gemini to determine the specific type of bias present.

C. Binary Classification of Bias

Our final model, as shown in Appendix E, shows very strong ability to detect bias in a user-LLM interaction. It is able to perform accurately about 70% of the time. This tool can be improved by using real life data that is customized to a specific CLM software and can be integrated wherever LLM tools are present. Every time a user decides to utilize an LLM tool to edit or create contract language, we could run the user question and the LLM response through this model and flag to the user whether they are producing biased or unbiased content. This would control the amount of unintentional biased clauses added into a specific contract.

D. Conclusion

Based on the discussion above, we can now answer our research question of "How can a contract lifecycle management (CLM) system be improved to detect and mitigate cognitive biases from both human interaction and generative AI during the contract negotiation process to ensure fair outcomes?". We have two specific tools from this research project that we can utilize to improve fairness in contracts. Firstly, we can identify the types of biases different user roles tend to create and provide extra training to those individuals. Furthermore, we can deploy our created model and integrate it with LLM chat interfaces to detect bias in a user-LLM interaction and prevent users from editing/creating contract language when a bias is detected. By utilizing these tools we can ensure fairness is maintained in our policies.

V. Limitation

While this research project successfully created a model to predict bias in user-LLM interactions, there are many limitations which can be improved in future projects.

A. Definitions of Bias

In this research project we identified bias by observing 4 different types of bias: anchoring bias, attentional bias, availability heuristic, and confirmation bias. While observing the Bias Transformation Matrix and the Bias in User Question by Role, we made conclusions and observations based on existing data, however there is no validation on if this is representative of the real world. Since I used ChatGPT to generate the data, the graphs depict how well ChatGPT can create and label those types of biases. Further research is required to identify which types of biases predominantly exist in real world legal context. Furthermore, future research experiments should utilize real life interactions between users and large language models and manually label each interaction with a bias. Future projects should also define specific definitions for bias.

B. Complicated Models

This research tried to combine concepts of cognitive science and technology. Due to this, I explored complex neural networks and Bayesian networks to develop the model. While this provided high scores, it may have been too complicated for the data we have. Since the data we have are short pieces of texts and interactions, we may be able to create a simple regression model along with LLMs. The complicated models on its own may be time and space consuming to run and provide the same precision output as the regression models. This will require further testing to see which models work best for this kind of dataset.

Despite these limitations, the research project effectively showed how we can use models to improve fairness in CLM software.

VI. Distribution of Responsibilities

Total Hours Completed: 106 Hours Hours Left: 3 Hours (Demo Preparation)

Week #	Task #	Task Description	Estimated Time (Hours)	Complete? (Y/N)
3	1	Create the template task list.	0.25	Y
3	2	Choose research question	0.25	Y
3	3	Read past example papers	2	Y
3	4	Brainstorm project ideas and impacts	2	Y
3	5	Discuss with peers about ideas	1	Y
4	6	Literature Review on ideas	4	Y
4	7	Plan technical feasability of ideas	2	Y
4	8	Finalize on topic based on research	1	Y
4	9	Find articles specific to the topic	2	Y
4	10	Explore existing tech regarding topic	1	Y
5	11	Analyze articles and cite sources	1	Y
5	12	Explore textbook and lectures for theories	1	Y
5	13	Create a plan and list hours	0.5	Y
5	14	Create rough draft of pitch	5	Y
5	15	Proof read, Edit, and Submit	1	Y

Fig. 1.

Week #	Task #	Task Description	Estimated Time (Hours)	Complete? (Y/N)
PROJECT PITCH DUE				
6	16	Understand textbook theory for practice	1	Y
6	17	Find other literature to support ideas	4	Y
6	18	Learn about basic Flask Python setup	1	N
6	19	Learn about SpaCy	2	Y
7	20	Learn about NLTK	2	Y
7	21	Learn about Hugging Face Transformers	2	Y
7	22	Learn about Affectiva Package	2	N
7	23	Learn about PyTorch	2	Y
8	24	Create technical plan and layout	3	Y
8	25	Set up basic backend with Flask	1	Y
8	26	Set up basic DB connection for data	1	Y
8	27	Implement basic model with NLTK/SpaCy	3	Y
9	28	Include Transformers and Affectiva	2	Y
9	29	Use PyTorch to complete the pipeline	2	Y
9	30	Do a basic end to end test of system	4	Y
9	31	Host the system so it is accessible	2	Y

Fig. 2.

Week #	Task #	Task Description	Estimated Time (Hours)	Complete? (Y/N)
OPTIONAL MIDPOINT CHECK-IN DUE				
10	32	Pause dev work to record learnings	2	Y
10	33	Analyze initial results to ensure accuracy	2	Y
10	34	Develop basic front end to accept data	3	Y
10	35	Connect the front end to backend	2	Y
11	36	Test end to end system with human data	2	Y
11	37	Test with generative AI output data	2	Y
11	38	Develop front end to show visual output	3	Y
11	39	Aggregate summary to show biases	3	Y
12	40	Create test data for business user	1	Y
12	41	Create test data for sales user	1	Y
12	42	Create test data for legal user	1	Y
12	43	Compare biase detection for all users	3	Y
13	44	Record findings and provide visual	2	Y
13	45	Find bugs and improve the system	1	Y
13	46	Begin final paper on research methods	2	Y
13	47	Begin final paper on research results	2	Y
14	48	Find and record potential gaps	2	Y
14	49	Literature review on future growth options	2	Y
14	50	Finalize tech demo	3	N
14	51	Submit the final paper	4	Y

Fig. 3.

Week #	Task #	Task Description	Estimated Time (Hours)	Complete? (Y/N)
FINAL REPORT DUE				
15	52	Buffer to fix or improve the tech demo	2	Y
15	53	Record and Submit the presentation	2	Y
15	54	Buffer to fix or improve presentation	2	Y
15	55	Buffer to fix or improve presentation	2	Y
FINAL PRESENTATION DUE				

Fig. 4.

VII. Appendix

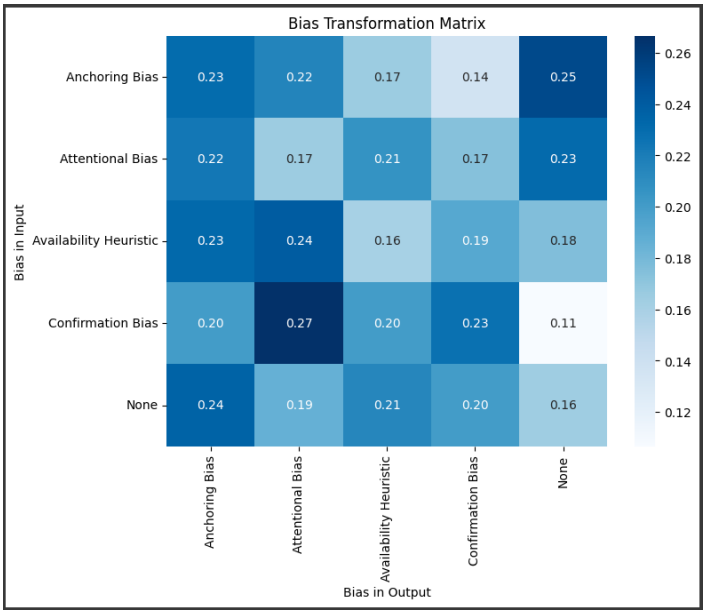


Fig. 5. Figure A

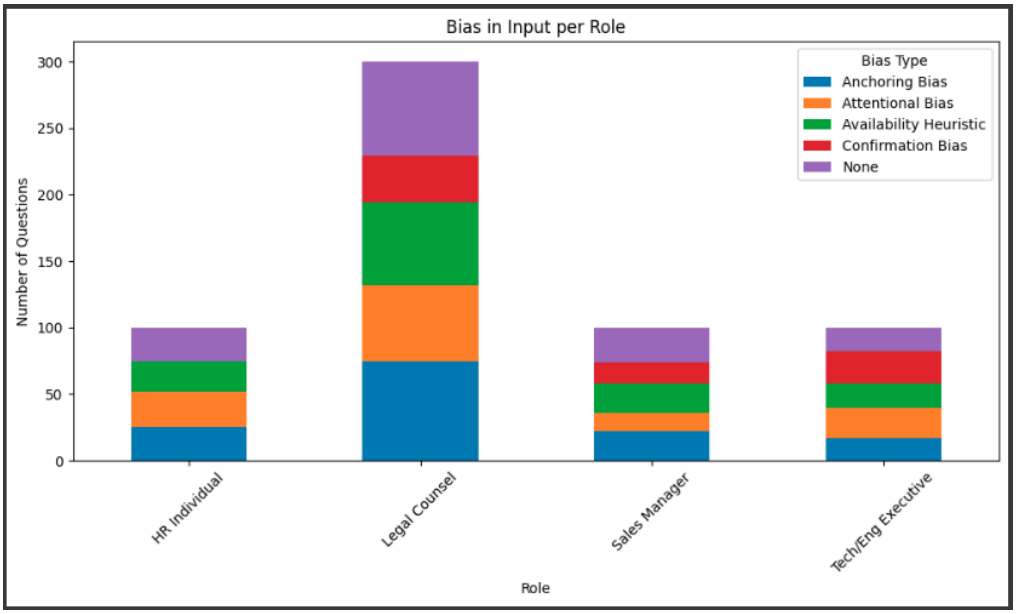
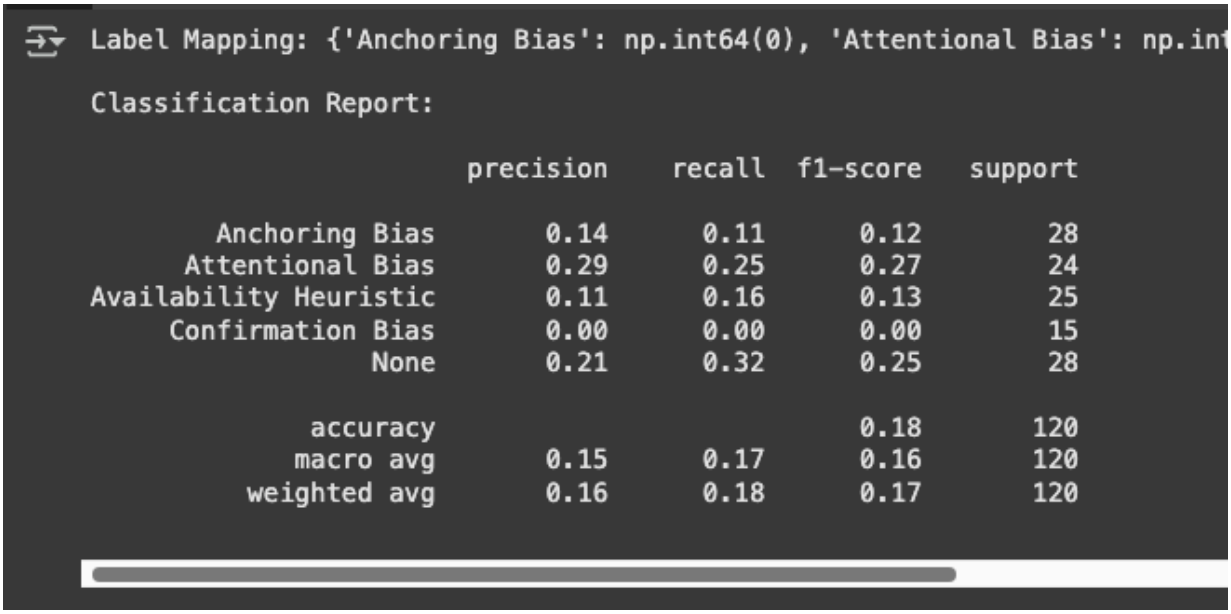
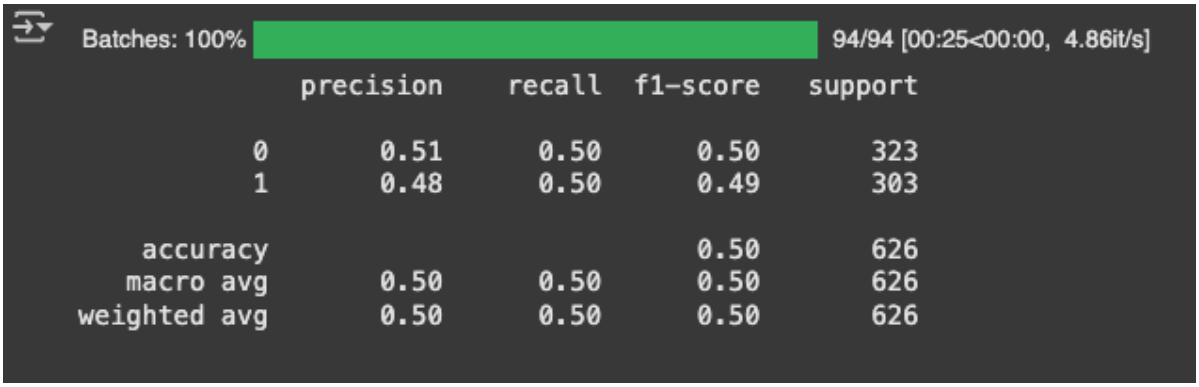


Fig. 6. Figure B



```
Label Mapping: {'Anchoring Bias': np.int64(0), 'Attentional Bias': np.int

Classification Report:

                       precision    recall  f1-score   support

        Anchoring Bias       0.14      0.11      0.12        28
     Attentional Bias        0.29      0.25      0.27        24
Availability Heuristic       0.11      0.16      0.13        25
     Confirmation Bias       0.00      0.00      0.00        15
                  None       0.21      0.32      0.25        28

              accuracy                           0.18       120
             macro avg       0.15      0.17      0.16       120
          weighted avg       0.16      0.18      0.17       120
```

Fig. 7. Figure C



	precision	recall	f1-score	support
0	0.51	0.50	0.50	323
1	0.48	0.50	0.49	303
accuracy			0.50	626
macro avg	0.50	0.50	0.50	626
weighted avg	0.50	0.50	0.50	626

Fig. 8. Figure D

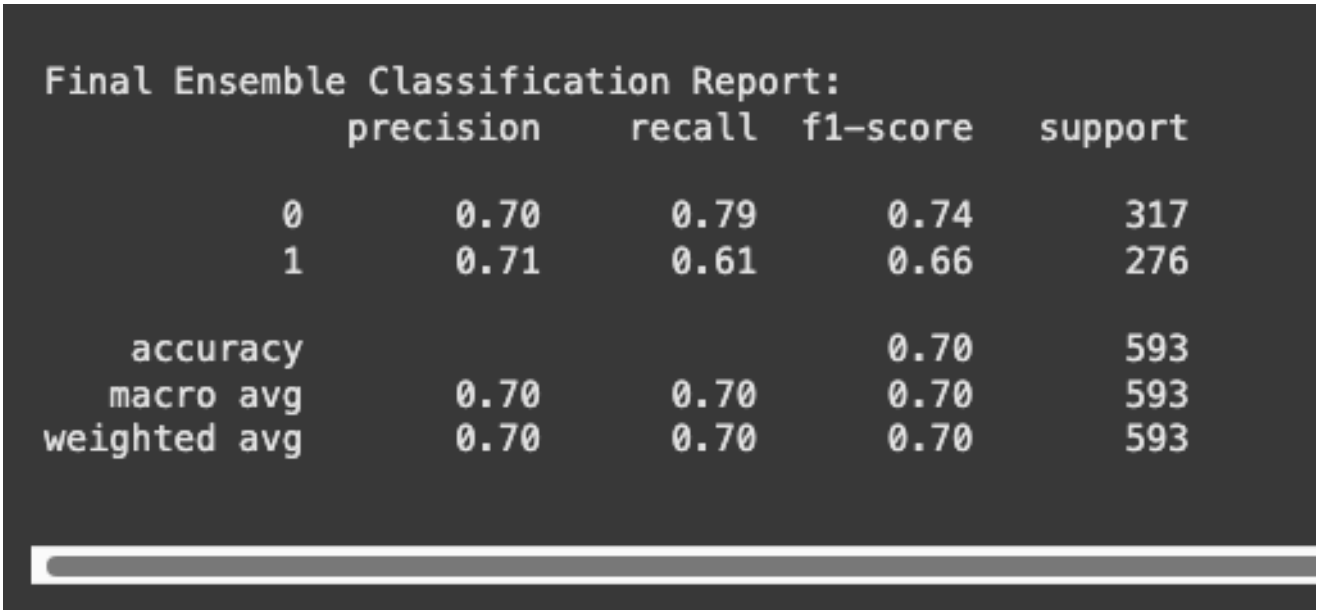


Fig. 9. Figure E

REFERENCES

[1] P. Thagard, Mind: Introduction to Cognitive Science. MIT.

[2] Cleveland Clinic, "What cognitive bias is and how to overcome it," Cleveland Clinic, https://health.clevelandclinic.org/cognitive-bias (accessed Apr. 9, 2025).

[3] "New study takes novel approach to mitigating bias in LLMS," Stanford Report, https://news.stanford.edu/stories/2025/02/bias-in-large-language-models-and-who-should-be-held-accountable (accessed Apr. 9, 2025).

[4] W. W. Park, "ARBITRATOR BIAS," Boston University School of Law, https://scholarship.law.bu.edu/cgi/viewcontent.cgi?article=1014&context=faculty_scholarship (accessed Apr. 10, 2025).

[5] L. Solan, T. Rosenblatt, and D. Osherson, "False Consensus Bias in Contract Interpretation," BrooklynWorks, https://brooklynworks.brooklaw.edu/cgi/viewcontent.cgi?article=1275&context=faculty (accessed Apr. 9, 2025).