Explainable Domain Specific Large Language Models: A Law Case Study

Majid Ghasemi[†]
Gurunameh Singh[‡]
Pratham Shah[*]
Tegveer Gogi [**]

Abstract A state-of-the-art question-answering system designed specifically for dissecting and understanding SEC legal documents is presented in this groundbreaking project. An important feature of this system is that it integrates the power of the Mistral 7B language model with an innovative knowledge graph approach, augmented by advanced natural language processing. An important feature of this system is its emphasis on explainability, a breakthrough in the application of artificial intelligence to legal matters. By utilizing this system, we intend to transform the landscape of automated legal assistance, making it not only more efficient but also more transparent and intuitive when dealing with the nuanced world of legal issues. It paves the way for a new era in legal document analysis, where AI-driven clarity and precision will become indispensable tools for legal professionals and stakeholders alike.

Keywords: Large Language Models; Knowledge Graphs; Explainable LLM; Explainable Knowledge Graph; Law LLM
Code Available on Github

1. Introduction

Over the past few years, large language models have taken the machine learning industry by storm. In the first instance, we were introduced to transformers, followed by a variety of Large Language Models (LLMs) such as GPT-3, BARD, PaLM, and LLAMA. Based on the logic of the attention mechanism from the transformer's architecture, LLMs are primarily based on the logic of the attention mechanism. By examining the words in sentences, it attempts to determine if there are any linguistic dependencies. As a result, it has a large number of parameters and, therefore, requires a lot of computation. Because of this, we will use pre-trained language models, which are considerably smaller. Pre-trained language models have the disadvantage of being statistical models; they learn inherent relationships between words. Therefore, they perform well when it comes to words that occur frequently

Wilfrid Laurier University, December 2023
[†]ghas1560@mylaurier.ca
[‡]chha8740@mylaurier.ca
[*]shah4771@mylaurier.ca
[**]gogi0860@mylaurier.ca

but poorly when it comes to words that occur less frequently. The statistical model also has the disadvantage of not being able to make logical judgments due to its nature as a statistical model. Although LLMs can infer based on semantic and syntactic knowledge, they perform poorly when faced with reasoning tasks (1). Using knowledge graphs, we are able to overcome these disadvantages. There is a high occurrence of rare words and rules that convey relational information, thus enhancing their reasoning capabilities (2).

There is an interpretability problem associated with LLMs, as with all other neural network models. It is difficult to trust these models because of their black-box nature. According to our previous discussion, LLMs use attention mechanisms to determine the underlying dependencies of human language. It is often difficult for humans to understand these relationships, which results in a lack of interpretability. If it is used in critical areas such as the military and banking, this can pose a problem. The black box behavior of the models should be explained somehow (3).

The recent approaches that have been utilized can be divided into two categories: Model-intrinsic, which can be applied to simple models due to their intrinsic simplicity, and Post-hoc, which provides an explanation after the model has been trained. The post-hoc method would obviously need to be used in this instance. As a result, the model would be able to explain the inferences that it made based on the data, which can be studied and used as a means of improving future inferences. In addition to using LLMs in Natural Language Processing (NLP), they can also be classified into two parts based on their use: Natural Language Generation (NLG) and Natural Language Understanding (NLU). The focus will be on NLU, and we will also explain the inferences made by LLMs (4).

In this study, we opted for a publicly available law knowledge graph and dataset (5) to further study the explainability along with other metrics that will be discussed in depth in the upcoming sections.

Upcoming sections are outlined as follows: In section 2, we delve into the literature review to know the gap at the intersection of Explainable AI (XAI) and LLMs. Section 3 describes the methodology we chose for the developed model. In section 4, we talk about our experiments and the results were achieved, and finally, in section 5, we wrap the report up and talk about the future research directions.

2. Literature Review

Studies have been conducted in which knowledge has been derived from pre-trained language models. Pre-trained language models offer the advantage of being able to train them on other data sets much more quickly. The

disadvantage of this method is that it does not learn data that was previously relevant. In this paper (6), the adapter is used to solve the problem of flushing the previous data used to train the model. Using a K-adapter, multiple genres of knowledge are infused without losing the original weights of the pre-trained model. The evaluation is based on the classification of entities, the answering of questions, and the classification of relations. The K-adapter methodology has been used to develop other similar ideas. Supposedly, the researchers injected common sense knowledge that could be easily incorporated. In order to overcome the hurdle, it was necessary to fill in the domain gaps between the knowledge on which LLM was already trained and the newly acquired external data. As a result, two adaptive layers were injected inside the pre-trained model, which were trained by knowledge triplets.

In this paper (1), further methods of injecting knowledge are discussed, including fused, embedding combined, unified data structures, knowledge-supervised, retrieval-based, and rule-guided injection.

The use of knowledge graphs has become increasingly popular among researchers as a means of improving explainability. Through the use of its relationship models, it captures complex underlying dependencies that were difficult to represent using traditional LLMs. By integrating knowledge graphs with LLMs and using graph attention neural networks, researchers have attempted to obtain human-understandable explanations for LLM inferences (7).

3. Methodology

In light of what we discussed in the previous sections, it is imperative to study LLMs with the specification of Explainability, especially in fields requiring complete understanding and trust, such as medical systems and law. According to our knowledge, the law field is not well studied like other fields, and in addition, terms that are used within the legal context need to be understood. It is also important that people are aware of the LLM decision process and the Knowledge Graphs (KGs) associated with it.

Figure 1 below illustrates the development process of our study and the steps we took. In order to make it easier to understand, we will define each and every concept and source in the following subsections.

3.1 Dataset, Ontology, and Knowledge Graph

The collection of data is the initial step in any data science project. Having the appropriate dataset is critical to ensuring the correctness of the outcomes. In our project, we utilized a publicly available dataset (5) that was gathered and constructed as a knowledge graph using litigation releases from the SEC's

website. The documents were categorized into four primary categories: Fraud, Insider Trading, Misappropriation of Funds, and Unregistered Brokers.

The SEC website refers to the official website of the United States Securities and Exchange Commission (SEC). The Securities and Exchange Commission is a federal agency responsible for enforcing federal securities laws, proposing securities regulations, and regulating the securities industry. The SEC's website serves as the primary source of information regarding its regulatory activities and functions.

3.1.1 Exploratory Data Analysis (EDA) on the Dataset

A dataset consists of 2,500 text documents extracted from the SEC website. We examined the dataset to understand key features better with the help of EDA, as can be seen from Figure 2.

A comprehensive analysis of financial crimes from 2010 to 2020 is presented in Figure 2. As can be seen from the leftmost bar chart, titled "Distribution of Years," there appears to have been an increase in the frequency of financial crime incidents reported each year towards the latter part of the decade. In the upper right corner of the chart is a column entitled "Frequency of Crimes," which categorizes the crimes, with the most common being "Fraud," followed by "Misappropriated Funds" and "Unregistered Activities.".

The "Top 10 Violations" chart in the lower section summarizes the most frequently violated financial regulations. Among the most prevalent violations are those relating to Section 10(b) of the Securities Exchange Act of 1934 and Rule 10b-5 thereunder. The top ten fines chart presents the penalties associated with these crimes, including 'Penalty', 'Disgorgement', and 'Prejudgment Interest'. In addition, it provides specific financial figures for each category, such as a $269,187 penalty for non-compliance with financial regulations.

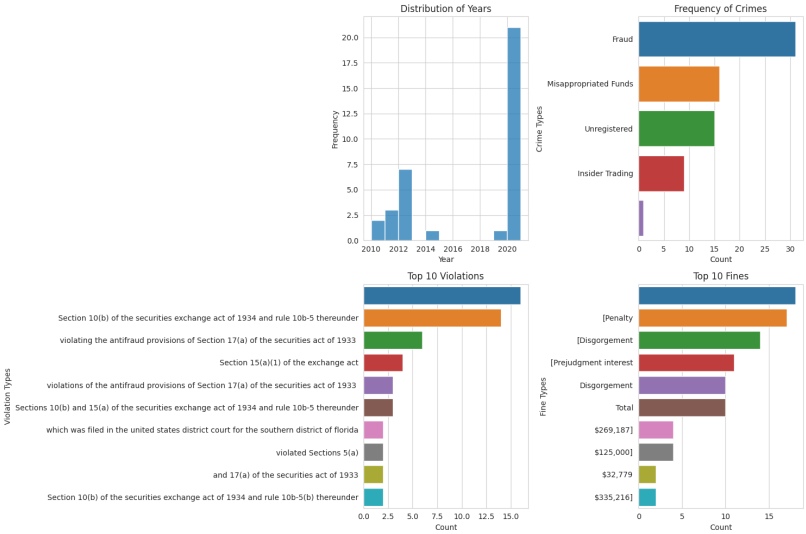


Figure 1
Exploratory Data Analysis Representation on the Dataset

3.1.2 Classification of Crimes

Multiple algorithms were employed to classify crimes in the dataset. Insider trading, misappropriation of funds, unregistered brokers, and fraud were the primary classifications. There were three main approaches to classification:

- LDA Unsupervised Topic Modelling: Initially, an unsupervised model was used, but it proved inaccurate and ineffective for classifying documents into multiple categories.
- BERT Text Classification: The BERT model struggled with low accuracy rates of 24.67%. Despite increasing training data size, accuracy stagnated.
- Regular Expressions: This method was the most effective, achieving a 95% accuracy rate in classifying the releases into various crimes.

Regular expressions were chosen in order to classify litigation releases.

3.1.3 Entity Extraction

To extract entities from the knowledge graph, the following methods were explored by the dataset developers:

- Identifying Subject, Object: The initial attempt focused on identifying the subject and object in documents, which was not very accurate.

- SVO Extraction: The approach was improved by identifying triplet phrases, which performed significantly better.

- Ontologies for Information Extraction: Ontology rules were constructed from litigation releases, including five main classes: Violator, Violation, Crime, Action Taken, Fine, and Date. This approach enabled a nested knowledge graph structure.

3.1.4 Specific Classes in the Ontology

- Violators: Extracted using regular expressions to identify patterns in the litigation texts.

- Laws: Three algorithms were developed to identify sections violated, refining the process for better accuracy and relevance.

- Fines: Categorized into disgorgement, penalty, prejudgment interest, and total fine, identified using specific keywords and patterns in the text.

- Action Taken: Determined by identifying keywords in the litigation releases that indicated legal actions.

- Dates: Extracted using regular expressions to determine the filing date of cases.

3.1.5 Knowledge Construction

The knowledge graph links various classes using relationships like: Violator → committed → Crime Violator → violated → Law Violator → face → Action Crime → ofValue → Fine

Figure 2 indicates the nested entities and relationships between the nodes of the law knowledge graph.

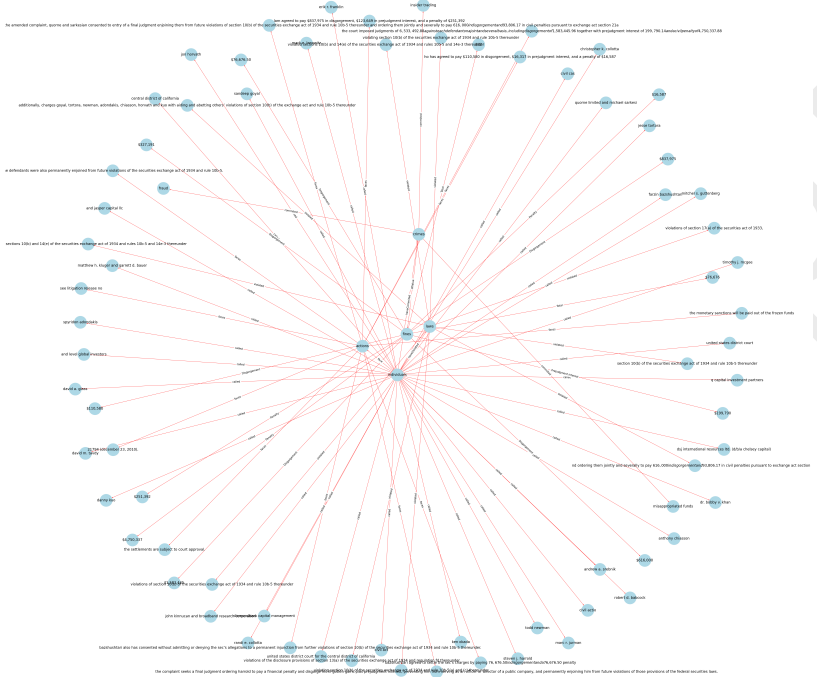


Figure 2
The nested entities and relationships

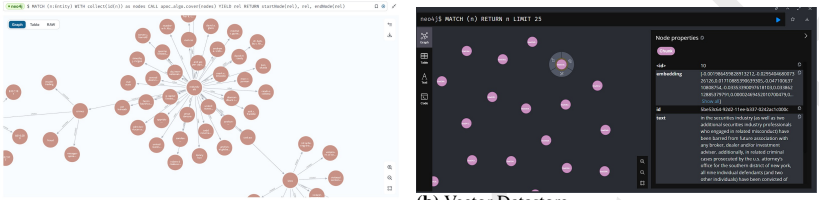
3.2 Neo4j

3.2.1 Vector Database

Due to its exceptional capabilities in managing data structured as graphs, Neo4j stands out as a highly acclaimed graph database management system. Neo4j uses nodes, relationships (edges), and properties instead of rows and tables to represent and store data, as is the case with relational databases. A more intuitive method of handling complex relationships between data points is provided by this approach, which is aligned with graph theory principles. It is particularly advantageous to use Neo4j for queries that involve intricate data relationships. It is particularly well-suited to environments in which data connections are as important as the data itself, such as social networking, complex network infrastructure, and intricate transactional data. We

chose Neo4j because its graph-centric data model facilitates the representation of complex hierarchies and interconnected networks. We sought to leverage Neo4j's robust features, such as its flexible schema, agile querying, and efficient storage of connected data. This system is designed to handle high-velocity data and provide real-time insights into the complex relationships between our data, which is important for the nuanced operations we anticipate in our queries and data transactions (8).

Representations of the vector datastore in our project can be seen below:



(a) Neo4J Knowledge Graph

(b) Vector Datastore

Figure 3
Neo4J KG and Vector Datastore Representations

3.2.2 Graph Database - Cypher Query Language and AuraDB

The Cypher Query Language (CQL) was used to interact with the Neo4j graph database. As part of its design philosophy, CQL focuses on pattern recognition and matching within the graph structure, which is expressed via an ASCII-Art syntax. By taking this approach, CQL becomes a declarative language that is derived from SQL but is far more expressive and straightforward to use when dealing with graph databases. CQL's benefits are manifold. The pattern-matching capabilities of this language are unmatched among graph query languages, allowing complex queries to be written in an elegant and simple manner. Our knowledge graph was constructed in CQL, which is optimized for graph processing thanks to features such as parameterized queries, subqueries, and composite operations. In addition, we adopted AuraDB's cloud service offerings. In order to alleviate the burdens associated with managing databases, Neo4j provides an AuraDB service that is fully managed. In addition to being easily deployable, the service ensures that our graph database is also scalable and accessible without our team having to manage infrastructure or perform administrative tasks. The managed service offered by AuraDB allows us to focus on leveraging our graph database's capabilities to the fullest extent without having to worry about the operational overhead normally associated with such complex systems (8).

3.3 The Language Model

We decided to use the Mistral 7b Model to evaluate the prompt and give the precise answer to the user due to many reasons. First, it is a 7-billion-parameter language model engineered for superior performance and efficiency. Second, Mistral 7B outperforms the best open 13B model (Llama 2) across all evaluated benchmarks and the best released 34B model (Llama 1) in reasoning, mathematics, and code generation. Mistral 7b model leverages grouped-query attention (GQA) for faster inference, coupled with sliding window attention (SWA) to effectively handle sequences of arbitrary length with a reduced inference cost (9)

As it is evident from Figure 4, Mistral 7B is completely dominant over the LLaMA models in terms of performance on language tasks and on diverse cognitive tasks, and thus, that was strong proof to choose this over other LLMs to get the better results in our work.

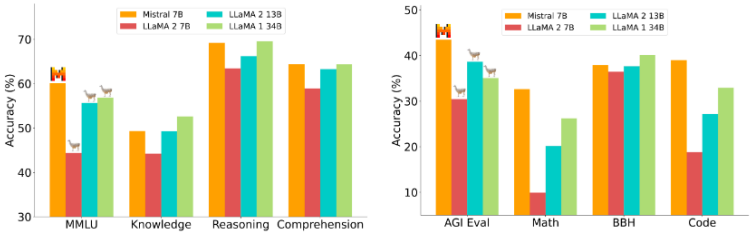


Figure 4
Performance Comparison Between Mistral 7B and other LLMs in terms of Performance on Language Tasks (Left plot) and on Diverse Cognitive Tasks (Right plot) (9)

3.4 Proposed Methodology

We propose a novel framework for a knowledge graph (KG)-based question-answering system tailored to legal documents, specifically focusing on the corpus of Securities and Exchange Commission (SEC) proceedings. As shown in Figure 5, The initial stage of our framework entails the extraction of a targeted subset of data from the SEC corpus. Through sophisticated natural language processing (NLP) techniques, this subset undergoes rigorous processing to discern relevant entities and their interrelations. These entities and relations are then meticulously organized into a high-level ontology, establishing a structured representation of intricate legal knowledge inherent in SEC documents. The core of our system is an entity and relation mapped to

an ontology. The graph is stored in AuraDB, a graph database that has been optimized to provide high retrieval efficiency, and this forms the basis for the response mechanism of our system. In order to facilitate retrieval, text chunks from the SEC corpus are tokenized and transformed into vector embeddings using OpenAI's language models. The embeddings are stored in a Neo4j vector datastore, which facilitates the process of matching similarity. When a search query is conducted, this is essential for identifying the most relevant documents. By handling both structured and unstructured data, we are enhancing the KG's richness and improving query precision. LangChain is used to convert natural language queries into actionable database queries. Using this method, these queries are efficiently converted into Cypher queries that are compatible with Neo4j. Mistral 7b is capable of processing natural language queries from users. As a result of this model, prompts are generated to navigate the KG in order to extract precise information. In addition, the entire system is powered by Azure's Virtual Machine Infrastructure, which provides the scalability, reliability, and computational power necessary to handle extensive data processing and query execution. Through this integrated framework, we aim to develop a system that can parse complex legal text and provide precise answers to user inquiries. Automated legal assistance tools will be enhanced as a result of this.

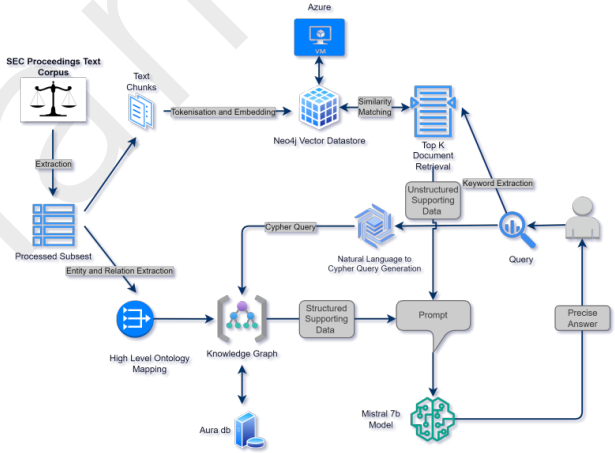


Figure 5
The Architecture of the Proposed Model

4. Experiments and Results

Our evaluation methodology centers on the utilization of semantic similarity scores to approximate traditional performance metrics due to the absence of a definitive list of correct answers. The key metrics calculated are Estimated Precision, Estimated Accuracy, and Estimated F1 Score.

The semantic similarity scores, ranging from 0 to 1, represent the closeness of the model's answers to the expected responses. A threshold value is set to categorize answers as "retrieved positives" or "retrieved negatives." Based on this threshold, we estimate precision and accuracy. The precision is estimated as the ratio of answers above the threshold (approximate true positives) to the total number of answers considered above a lower threshold. Accuracy is calculated as the ratio of approximate true positives to the total number of answers.

For the F1 score, which is the harmonic mean of precision and recall, we also estimate recall as the ratio of approximate true positives to an estimated total number of relevant answers in the dataset.

Our evaluation methodology involves both computational and human assessments. For the computational part, we use semantic similarity scores to approximate traditional performance metrics, given the absence of a definitive list of correct answers. The metrics calculated are Estimated Precision, Estimated Accuracy, Estimated Semantic Accuracy, and Estimated F1 Score. The semantic similarity scores, which range from 0 to 1, reflect the closeness of the model's answers to expected responses. Based on set thresholds, we estimate precision, accuracy, and F1 score.

In addition to computational evaluation, we conducted a human evaluation where five individuals rated the answers on a scale of 1 to 5. These ratings were then categorized as "Accurate," "Moderately Accurate," or "Ambiguous."

Computational Evaluation Results The results of the computational evaluation are as follows:

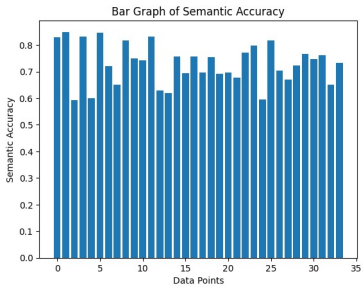
Metric	Value (%)
Estimated Precision	63%
Estimated Recall	73%
Estimated Semantic Similarity Score (Semantic Accuracy)	73%
Estimated F1 Score	68%

Table 1
Computational Evaluation Results

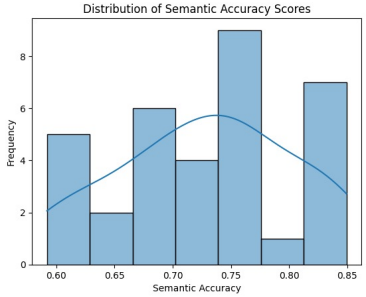
As can be seen from Figures 6 and 7, we plotted the semantic accuracy results in a couple of different forms to give you the readers' better overview of what is going on.

Histogram: The histogram shows a distribution with a single peak (unimodal), with the highest frequency of scores around 0.75. The shape of the distribution curve suggests a normal distribution but with a slight skew towards the lower scores. The tail on the left side of the distribution is a bit longer and fatter, which confirms a slight skewness to the lower values.

Bar Graph: The bar graph shows individual semantic accuracy scores for what appears to be 35 different data points. The scores fluctuate across the dataset, with no discernible pattern of increase or decrease. The scores range widely from below 0.4 to just above 0.8, indicating variability in the semantic accuracy of different data points.



(a) Bar Graph Representation of the Semantic Similarity (Accuracy)



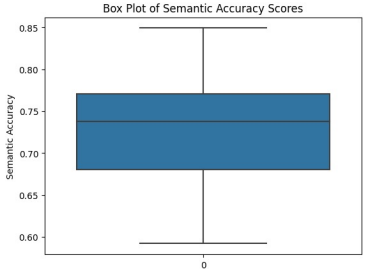
(b) Histogram Representation of the Semantic Similarity (Accuracy)

Figure 6
Different Representations of the Semantic Similarity (Accuracy)

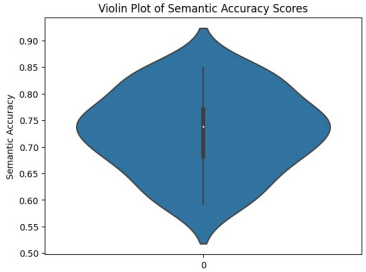
Violin Plot: The bulk of data points seems to be concentrated around a semantic accuracy score of approximately 0.75, as indicated by the widest part of the 'violin'. The median is also close to this value, marked by the white dot. The distribution is relatively symmetrical but slightly fatter on the lower end, suggesting a minor skew towards lower scores. The thinness at the top and bottom suggests fewer data points with very high or very low semantic accuracy scores.

Box Plot: The box plot indicates that the middle 50% of the data (the interquartile range, or IQR) is fairly tight, with scores mostly between approximately 0.72 and 0.78. The median line within the box appears to be slightly closer to the top of the box, indicating that the distribution might be slightly skewed towards the lower end. The "whiskers" extend from approximately 0.65 to 0.85, showing the range of the bulk of the data, with no individual

points indicating outliers.



(a) Box Plot Representation of the Semantic Similarity (Accuracy)



(b) Violin Plot Representation of the Semantic Similarity (Accuracy)

Figure 7
Violin and Box Plots of the Semantic Similarity (Accuracy)

Human Evaluation Results The human evaluation yielded the following summarized results:

Category	Total Count
Accurate	70%
Moderately Accurate	17%
Ambiguous	13%

Table 2
Human Evaluation Results

Here is the image of the results based on the Human Evaluation.

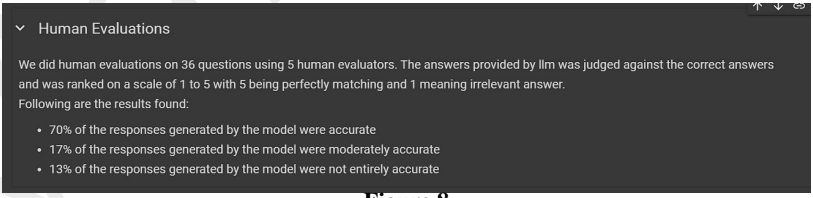


Figure 8
Human Evaluation Results

These results provide insights into the performance of our model, highlighting both computational estimations and human perceptions of the answers' accuracy. The combination of these methods offers a comprehensive understanding of the model's effectiveness.

5. Conclusion and Future Work

This study demonstrated that the Mistral 7B model played a significant role in the performance of the system, both in computing and in human assessments. In terms of precision, recall, and F1 score, it achieved an estimated 63% accuracy, 73% recall, and 68% F1 score. As part of future efforts, it is envisaged that the dataset will be expanded and the system's precision will be improved. A key objective of this project is to extend its application to a variety of fields and integrate it into real-world legal environments in order to revolutionize legal research and decision-making processes in the future.

References

[1] Hu, L., Liu, Z., Zhao, Z., Hou, L., Nie, L., Li, J. (2023). A survey of knowledge-enhanced pre-trained language models. IEEE Transactions on Knowledge and Data Engineering.

[2] Peng, C., Xia, F., Naseriparsa, M., Osborne, F. (2023). Knowledge graphs: Opportunities and challenges. Artificial Intelligence Review, 1-32.

[3] Chen, Z., Singh, A. K., Sra, M. (2023). LMExplainer: a Knowledge-Enhanced Explainer for Language Models. arXiv preprint arXiv:2303.16537.

[4] Qader, R., Portet, F., Labbé, C. (2019). Semi-supervised neural text generation by joint learning of natural language generation and natural language understanding models. arXiv preprint arXiv:1910.03484.

[5] github.com/AnjaneyaTripathi/knowledge_graph

[6] Wang, R., Tang, D., Duan, N., Wei, Z., Huang, X., Cao, G., ... Zhou, M. (2020). K-adapter: Infusing knowledge into pre-trained models with adapters. arXiv preprint arXiv:2002.01808.

[7] Qin, C., Kim, S., Zhao, H., Yu, T., Rossi, R. A., Fu, Y. (2022, August). External Knowledge Infusion for Tabular Pre-training Models with Dual-adapters. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 1401-1409).

[8] https://shorturl.at/kGJT9

[9] Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. D. L., ... Sayed, W. E. (2023). Mistral 7B. arXiv preprint arXiv:2310.06825.