Financial and Statistical Methods

Predicting Workers' Compensation Dispute Outcomes with Large Language Models

Vajira A. Manathunga[1a], Duyen Hai Doan[2b]

[1] Middle Tennessee State University, [2] Georgetown University

Keywords: Workers' compensation insurance, LLMs, Traditional NLP techniques, Random forest, Gradient boosting, XGB, BERT, Gemini, GPT, Actuarial science

Variance

Vol. 19, 2026

Workers' compensation insurance is one of the oldest social insurance programs in the United States, predating both Social Security and unemployment insurance. When disputes arise between employees and employers over benefit entitlements, most states require resolution through administrative boards. In this study, we evaluate whether large language models (LLMs) can predict the outcomes of workers' compensation cases more accurately than traditional, domain-specific natural language processing (NLP) techniques under the zero-shot learning paradigm. We compare performance under two input scenarios: using only the initial "Issues" filed and using the full "Findings of Fact" narrative of each case, and we measure predictive accuracy against actual board decisions. Our results show that, with access to a sufficiently large context window, LLMs match or surpass the performance of specialized NLP pipelines despite having no task-specific training on workers' compensation data. This finding underscores the practical utility of LLMs in case outcomes for the plaintiff, the employer, actuaries, and the insurance carrier.

1. INTRODUCTION

Large language models (LLMs), a subset of artificial intelligence (AI) systems, have revolutionized the way industry experts, as well as novices, perceive the future of models built on deep neural networks and trained on large corpora of datasets. As of now, LLMs are being utilized in a wide range of applications across various industries, including medicine, education, finance, and engineering (Hadi et al. 2023). ChatGPT and some other chatbots that use LLMs are becoming some of the most widely used LLM-based applications. In the middle of this revolution, the question of interest for us is how actuaries and insurance professionals can use them for their tasks. Although other industries have leveraged LLMs beyond simply generating text and answering questions, it remains unclear how actuaries can benefit from this growth. Can LLMs enable actuaries and industry professionals to conduct groundbreaking analyses of complex data streams, allowing for the discovery of hidden patterns while improving prediction accuracy and streamlining complex administrative functions throughout their systems? In this research, our motive is to answer whether LLMs can outperform standard machine learning techniques for classification tasks in the workers' compensation insurance arena. The question stemmed from our reading of "Financial Statement Analysis with Large Language Models" (Kim, Muhn, and Nikolaev 2024). In this paper, the authors demonstrate that LLMs exhibit a relative advantage over human analysts when it comes to analyzing financial statements. They also concluded that LLMs can perform at the same level as narrowly trained, state-of-the-art machine learning models specifically designed to analyze financial statements. The paper has been retracted temporarily to revalidate its findings; however, the core

a Vajira A. Manathunga is the undergraduate director of the actuarial science program and an associate professor of mathematics at Middle Tennessee State University. His research bridges actuarial science, data science, and AI, with recent projects using large language models for insurance analytics and workers' compensation. He leads curriculum innovation, develops university partnerships, and mentors students through research and teaching, while actively working to expand MTSU's national and international profile in actuarial education.

Address for correspondence: vajira.manathunga@mtsu.edu

b Duyen Hai Doan is a graduate student in mathematics and statistics department at Georgetown University, where she also serves as a data analysis intern for the medical center, using SAS for data manipulation and supporting research initiatives. She holds an MS in professional science–actuarial science from Middle Tennessee State University and a BS in finance from Lewis University. Her professional interests lie at the intersection of quantitative analysis, data science, and their applications in finance and public policy.

idea remains of interest to researchers. In our own research, we aimed to investigate the potential utility of using LLM in workers' compensation cases. In the next subsection, we will provide an overview of the workers' compensation insurance program, as it is one of the two main components of this research.

1.1. WORKERS' COMPENSATION INSURANCE

The United States launched Workers' Compensation as its initial social insurance program before developing Social Security and unemployment insurance systems. Workers' Compensation exists to provide specific benefits, as defined by law, to employees who suffer injuries or develop illnesses due to their work-related activities. Workers' compensation insurance policies cover medical treatment, temporary and permanent disability wage replacement, vocational rehabilitation services, supplemental job displacement assistance, and death benefits to dependents. The benefits a claimant can receive are determined by the specific details of their case.

The foundation of this system rests on the "exclusive remedy" doctrine. This rule establishes that employers are responsible for workplace injuries, regardless of who is at fault, and that workers' compensation benefits become the sole legal remedy available to injured employees against their employer. Under this framework, employees generally cannot bring separate tort actions against their employer to seek additional damages from the same workplace incident (Workers Compensation Insurance 2016). The payments received by workers under workers' compensation cases are known as indemnity payments and usually cover about two-thirds of lost wages. These payments depend on state regulations, which set minimum and maximum limits (Butler and Worrall 1983). The definition of a compensable injury has expanded beyond accidents to include long-term health issues from work. It's important to distinguish "impairment" (medical loss of function) from "disability" (how much the impairment affects work). Compensation typically covers partial lost wages during recovery and lump sums for lasting disability, plus medical and rehab costs. Employers often invest in extra rehab services to help employees recover and return to work, which can be more cost-effective than paying for permanent disability (Guyton 1999).

Workers' compensation claims are handled by state boards, but decisions can be appealed in court. However, the use of the state board process is more cost-effective than formal court litigation. Most of the time, states require employees to go through the state board or a judge designated for workers' compensation cases before moving to higher courts. The workers' compensation insurance workflow can vary from state to state, depending on local regulations. However, in general, the following process can be observed (New York City Bar Association 2025; California Department of Industrial Relations, Division of Workers' Compensation 2023; Texas Department of Insurance, Division of Workers' Compensation 2025). When job-related injuries occur, employees must inform their employers immediately. After the employer files the claim with the workers' compensation insurance carrier, the carrier investigates and decides whether to approve or deny it. If the claim is denied or disputed, the next step is dispute resolution, which is usually handled by the state board. After exhausting the state board process, the employee may have a narrow path to litigate the case in the state court (https://www.ic.nc.gov/faqs.html).

At this point, we can identify several stakeholders with vested interests in the outcomes of workers' compensation insurance claims. First, we have three primary stakeholders: the employee, the employer, and the insurance carrier. Beyond these, we have a state-designated regulatory authority that oversees claim disputes. Finally, the general public may also be interested in how these cases are handled. In this research, we take the perspective of an analyst in an insurance company and aim to understand how these claim disputes are ultimately resolved at the state regulatory board. This understanding would help the company decide whether denying a certain claim type is prudent by leveraging the insight gained from the predictive model.

The use of data analytics in the workers' compensation insurance arena is not new to many. Traditional statistical-based methods and machine learning techniques have been employed to identify claim frequency, severity, injury rates, and many other critical factors (Meyers et al. 2018; Mathews 2016). Predictive models are applied to understand the cause of occupational injury, identify the most efficient healthcare provider for injured employees, and forecast compensation (Moniz 2019; Vinit Patwa 2024). Having discussed the workers' compensation insurance process and the application of statistical and machine learning approaches therein, we will next proceed to discuss LLMs.

1.2. LLM MODELS

LLMs are a foundational technology in the area of AI. Initially developed as a subfield within natural language processing (NLP) to enhance natural language understanding (NLU) and the generation of natural language (NLG), LLMs have become a focal point of the modern AI revolution (Chang et al. 2024). LLMs are generally characterized by their substantial parameter counts and extensive training on large datasets. Nowadays, LLMs have moved from training on traditional text corpus to computer codes, images, and many other formats, introducing multimodal models. These LLMs have demonstrated considerable potential across various industries, particularly those that analyze and process vast quantities of textual information.

The insurance industry is characterized by its use of large amounts of text data, which stem from claim descriptions, doctors' notes, policy documents, customer communications, and various other sources. Traditionally, these unstructured text data were discarded, or used for qualitative understanding, or coded into categorical variables to use in traditional machine learning models. The traditional methods and techniques from NLP that can be used in the insurance industry were extensively discussed in Ly, Uthayasooriyar, and Wang (2020). Now, with the introduction of LLMs, the insurance industry is poised to reap the capabilities of new LLMs. Insurers can now utilize LLMs for

tasks like client interaction, claims processing, underwriting, and automating processes that were previously handled manually. The challenges of implementing LLMs in the insurance industry range from model fairness and transparency to possible legal and ethical consequences (Ferrer et al. 2021).

According to Zhao et al. (2025), language models can be divided into four types: statistical language models such as n-gram models, neural language models such as word2vec, pretrained language models such as BERT, and LLMs such as GPT-4. The authors of this article describe LLMs as scaled-up versions of pretrained language models. However, they emphasize the unexplained "emerging abilities" of LLMs due to a large number of parameters in these models. The authors also discuss how the research community would struggle to develop their own LLMs due to the cost associated with developing these models from the ground up. Key categories of pretrained language models and LLMs can be divided into four types: decoder-only models, encoder-only models, encoder-decoder models, and multimodal. In language models, encoders are used to understand the input text sequence, and decoders are used to generate the output text sequence. Examples of decoder-only models are GPT family models such as GPT-2 and -3. An example of an encoder-only model is BERT. T5 and BART are examples of encoder-decoder models. Finally, Google Gemini and newer versions of OpenAI models are examples of multimodal LLMs, which use non-text data in the training phase. The next subsection is devoted to understanding current applications of LLMs in workers' compensation insurance.

1.3. TEXT DATA, WORKERS' COMPENSATION INSURANCE, AND LLMS

Even though workers' compensation insurance products are rich with textual information, the use of LLMs is still scarce. This marks a significant departure from other fields, such as finance, engineering, and medicine, where LLMs have gained rapid traction. Nevertheless, here we summarize several research studies conducted at the crossroads of LLMs, insurance, and actuarial science.

A general framework for analyzing the consistency of text in insurance claim reports derived from various sources was discussed in D. Li et al. (2025). The authors used ChatGPT and distance metrics to measure the discrepancies in texts in insurance claim reports. In this paper, authors also mentioned the scarcity of literature in use of LLMs to solve practical needs and applications of insurance and actuarial science. Possible use of LLMs in insurance were discussed in C. Cao et al. (2024). It proposed that the integration of LLMs allows insurance companies to improve customer service response time, streamline the claim process, and enhance the accuracy of risk assessment. However, the authors also mentioned unique challenges the insurance industry faces primarily due to the sensitive nature of the data and the high standards set forth by regulators. A systematic exploration of the effectiveness of GPT-4 in various multimodal tasks in insurance was conducted by Lin et al. (2024). The research used four types of insurance products: auto, health, agriculture, and property, to analyze how effective GPT-4 was at various tasks in a given insurance type. For example, for auto insurance, the authors analyzed whether GPT-4 is effective in vehicle underwriting, detecting dangerous driving behavior, vehicle claim processing, and fraud detection. The researchers concluded that GPT-4 is remarkable in its robust and comprehensive understanding of insurance scenarios. However, they also mentioned that GPT-4 struggles with detailed risk assessment and suffers from hallucination in image understanding. Applications of LLM specifically in actuarial science were discussed by Balona (2024).

At this point, we come back to our own research questions, which we pose formally in the next section. As of now, it is clear that LLMs offer substantial opportunities for the insurance industry to reshape how it uses text data collected through various sources.

2. PROBLEM STATEMENT AND HYPOTHESES

An employee who suffers a workplace injury has the right to file a workers' compensation claim against their employer or the employer's designated insurance provider. After reviewing the claim, the employer may choose to reject it fully, offer a partial payment, or grant total compensation. The worker must decide whether to accept the employer's resolution or challenge it by filing a dispute with state regulatory bodies. Specialized state commissions handle workers' compensation disputes in most states. These commissions fulfill the role of dispute resolution bodies, but their proceedings follow a trial-like format with legal representatives present for both the employee filing the claim and the defending employer or insurance carrier. The hearing precedes the commission's examination of evidence from both sides, including expert witness testimony, to determine whether the injury relates to work activities or other factors. The commission delivers a binding resolution that determines the winning party and defines settlement terms when applicable. Employees have a narrow pathway to challenge unfavorable commission decisions by taking their case to the established court system if they believe the decision to be incorrect.

Our research aims to investigate how historical workers' compensation case records can aid in predicting outcomes for new cases using LLMs. We specifically focus on the following question: Can off-the-shelf, non-tuned, commercial LLMs outperform other specialized NLP techniques in predicting likely outcomes for workers' compensation cases?

The question comes under the text classification task. We will use the following traditional NLP-based models versus LLM models to predict the likely outcome of workers' compensation cases. For traditional NLP models, in order to convert text to numerical features, we used TF-IDF, word2vec, and BERT coupled with classification techniques such as random forest, gradient boosting, and XGBoost. For LLM models, we used eight: `deepseek-chat (deepseek-v3)`, `claude-3-haiku`, `gemini-1.5-pro`, `gemini-1.5-flash`, `gemini-2.0-flash`, `gpt-3.5-turbo`, `gpt-4.1-mini`, and `o4-mini`.

2.1. TEXT CLASSIFICATION OF WORKERS' COMPENSATION DATA

Text classification is one of the main topics in the NLP domain. The purpose of text classification under supervised learning is to determine the category or label to which a given piece of text belongs (Campesato 2022). This touches on a variety of topics: fraud detection (Y. Wang and Xu 2018), plagiarism detection (Barrón-Cedeño et al. 2013), opinion and sentiment classification (Abbasi, Chen, and Salem 2008), and topic modeling, to name a few. With the explosive proliferation of digital documents, traditional manual text classification has become prohibitively labor-intensive and costly (Q. Li et al. 2022). This has led to the development of various machine learning–based models to automate the text classification process. According to Q. Li et al. (2022), text classification models can be divided into two broad categories: traditional statistical and machine learning models, and deep learning models. However, recent advancements in LLMs necessitate a reclassification of text classification models into three distinct categories: traditional statistical and machine learning models, traditional deep neural network models, and pretrained LLMs.

Text classification research within the insurance domain, particularly in actuarial science, has been relatively limited compared to other fields. However, textual information is abundant in insurance, encompassing sources such as medical reports, police reports, accident narratives, various correspondence letters, and repair estimates, to name a few. As mentioned in GV et al. (2021), these documents serve as the foundation for insurance claim processing workflows and are integral to various other business operations. Automating the classification and extraction of pertinent information from these documents has the potential to substantially enhance the efficiency of numerous business processes, curtail manual operational expenses, and elevate both the quality and reusability of critical data.

The latent Dirichlet allocation (LDA) text analysis approach to identifying insurance fraud in automobile insurance was discussed in Y. Wang and Xu (2018). This approach combined traditional numerical and categorical features from claims data with topical features extracted from textual descriptions using LDA. These features were then input into a deep neural network to predict the likelihood of a claim being fraudulent. In Kindbom (2019), researchers compared a deep neural network called long short-term memory (LSTM) with a simpler, interpretable random forest model for classifying insurance-related customer messages as questions or non-questions. These messages were received by the Swedish insurance company Hedvig from their customers. To train the random forest classifier, the authors used two variants of the bag-of-words (BoW) method to convert messages into feature vectors and tuned several of the model's hyperparameters. For the LSTM model, they employed word2vec for word embedding and also tuned various hyperparameters. The study concluded that while the LSTM model marginally outperformed the simpler random forest model, both were outperformed by human classification. To establish a human baseline, the researchers enlisted eight randomly selected individuals.

Text classification in the worker's compensation domain is almost nonexistent. Yamin et al. (2016) used a narrative text analysis method and workers' compensation data to identify whether injuries are machine-related or not. Another study used Alaska workers' compensation data to classify nonfatal work-related injuries (Lucas et al. 2020). Even though this study mentioned the free-from-text data, these were limited to claimant occupation and resident city, along with other coded categorical and numerical variables. Y. Cao, Chen, and Quan (2024) analyzed the risk an insurer faces due to litigation. In this research, the authors used more than 300,000 litigation outcome documents from a Chinese court. These documents were preprocessed, tokenized, and then embedded into various dimensional vectors using methods such as BERT, RoBERTa, BGE, and LLaMA. Then the authors either used a neural network classifier to predict whether the plaintiff won or lost, or fine-tuned those pretrained LLM models. The method was applied to both the original Chinese language texts and their English translations. This research can be directly related to our research discussed in this paper.

Our research fills one of the main gaps in the workers' compensation domain by analyzing case records and predicting the likely outcome using LLMs. It will be equally valuable to actuaries, insurance companies, and legal professionals.

2.2. DATASET

A critical consideration for this research is the dataset. While some states make workers' compensation data publicly available, this study specifically utilizes data from the North Carolina Industrial Commission, where disputed workers' compensation claims undergo an administrative law process rather than traditional court litigation. The process includes:

1. Deputy commissioner hearing
2. Discovery and written arguments
3. Deputy commissioner's decision
4. Appeal to the full Commission
5. Appeal to the courts

The data are collected through the public government database available at https://www.ic.nc.gov/database.html. The dataset encompasses information pertaining to the first three stages mentioned above. The process of downloading data is given in Appendix B. Typically, case documents presented before the deputy commissioner comprise the following sections:

1. Introduction: This section typically provides an overview of the case, including the case number, names of the parties involved, and a summary of the legal proceedings. It outlines the dates of hearings and depositions, any changes in the presiding deputy commissioner, and the final closure date of the case record. This section serves to establish the procedural context before presenting the findings and decision.

2. Appearances: This section lists the legal representatives for both parties, including the names of the law firms and attorneys for the plaintiff and defendants. It records the parties' legal counsel as acknowledged in the pretrial agreement and during the hearing.
3. Stipulations: This section outlines the agreed-upon facts between the parties, confirming jurisdiction, the applicability of the Workers' Compensation Act, and the employment relationship. It details the incident date, injury, insurance coverage, employee wages, compensation rate, and medical expenses covered by the defendants. Relevant documents, medical records, and other evidentiary materials are also included.
4. Issues for Hearing: This section clarifies the matters to be resolved and focuses the hearing on these essential questions to facilitate a fair and thorough adjudication of the case.
5. Findings of Fact or Evidence Admitted: The "Evidence Admitted" section, sometimes titled "Findings of Fact" or appearing alongside it, lists the exhibits accepted into the case record, such as a pretrial agreement, Industrial Commission forms, medical records, medical bills, personnel files, and discovery responses. Regardless of the title, this section serves the same purpose: to establish and document the evidentiary basis for the case's findings and conclusions.
6. Conclusion of Law: This section clarifies the legal basis for the decision and ensures that the conclusions align with established legal precedents and statutory requirements.
7. Award: This section formalizes the financial and procedural outcomes based on the case's findings and legal conclusions, ensuring that the claimant receives the appropriate relief and compensation.

We have used 15,406 workers' compensation case records ranging from 1990 to 2024. However, these were reduced to 14,225 or 6,103 cases, depending on the independent variable. More details are given in the next section.

2.3. INDEPENDENT AND DEPENDENT VARIABLES

The workers' compensation case files from the North Carolina Industrial Commission lack straightforward indicators of "winner" or "award amount" in their decisions, which makes extraction difficult. A human analyst reviewed each file and extracted data from the "Award" section into a column labeled "Decision." The analyst recorded decisions as "1" for plaintiff win, "0" for plaintiff loss, and "2" for inconclusive. The cases that received dismissals for multiple reasons were recorded as a loss for the plaintiffs. In some cases, the court ordered both the plaintiff and defendant to pay different penalties to the state commission. At this stage, a subjective decision was made by the human analyst to determine if the plaintiff won, lost, or had an inconclusive result. A decision was recorded as a win when the state commission ruled in favor of most of the plaintiff's requests. This research studies how well new LLMs and traditional NLP methods predict "win" or "lose" outcomes in workers' compensation cases by using

1. "Issues" as the independent variable and "Decision" as the dependent variable, and
2. "Findings of Fact" as the independent variable and "Decision" as the dependent variable.

We believe issues are known to the plaintiff before they even contact a lawyer for their cases. Findings of fact also exist beforehand but may need extra effort to uncover, which is usually done during the hearing at the Industrial Commission through various testimony, such as by expert witnesses. However, for this study, we assume findings of fact also exist beforehand so that plaintiffs and employers both can make informed decisions.

Of the initial 15,406 records, 14,225 contained nonempty Findings of Fact and 6,103 contained nonempty Issues. Therefore, subsequent analysis contained two data frames, where one is of size 14,225 rows and the other is of size 6,103 rows. Summary statistics for the dependent variable are presented in Table B.1. The summary statistics for independent variables `Preprocessed_Issues` and `Preprocessed_Facts` in each training set are reported in Tables B.2, B.3, B.4, B.5, B.6, B.7, B.8, and B.9.

2.4. LIMITATIONS

A significant limitation of this research is the exclusive reliance on data from North Carolina. Given potential variations in legal regulations across states, the conclusions may not be directly applicable to other states without testing on their respective datasets. Another significant limitation is human decision-making. Most workers' compensation cases have multiple claims and counterclaims by the plaintiff and the defendant (employer or insurance carrier). Thus, deciding the prevailing (winning) party is not easy and may become a subjective decision. One such approach, according to Geary (2024) is to "define the prevailing party as the party that prevails on the central claims advanced and receives substantial relief in consequence thereof." But this is not an easy task, and the difficulty of deciding the result in workers' compensation cases becomes a limitation itself. There is no standard approach to overcoming this limitation.

3. METHODOLOGY

An overview of the research methodology used to compare traditional NLP techniques and LLMs in this study is shown in the flowchart in Figure 1.

3.1. TRADITIONAL NLP MODEL SETTINGS

The research incorporates traditional machine learning and NLP techniques as well as LLMs. The pipeline of text classification algorithms for traditional NLP models can be listed as follows: data preprocessing, feature extraction, classification, and evaluation (Kowsari et al. 2019).

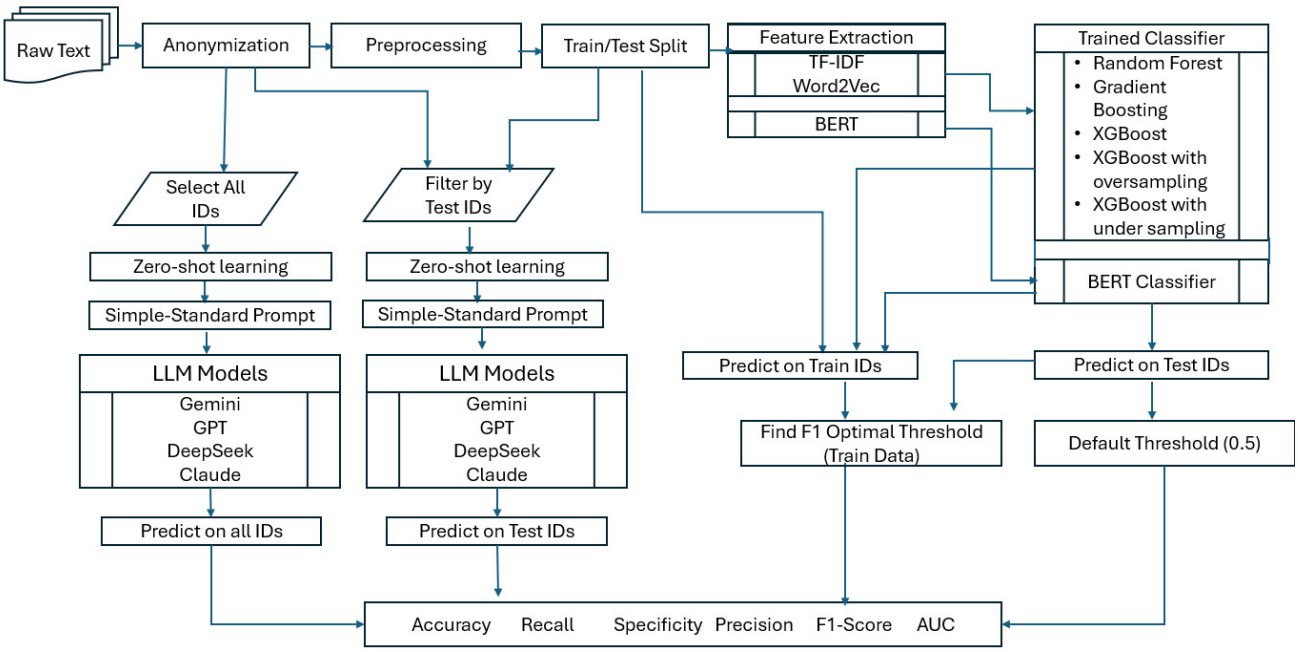

Figure 1. Research methodology to compare traditional NLP techniques vs. LLMs.

3.1.1. DATA PREPROCESSING

The purpose of text preprocessing is to enhance the quality of raw data for feature extraction, hence improving the effectiveness of NLP models and leading to better insights. Therefore, text preprocessing includes cleaning, noise reduction, standardization, and many other techniques. A survey of text preprocessing can be found in Kadhim (2018) and Kathuria, Gupta, and Singla (2021). In our research, we used the following preprocessing techniques in the given order: anonymizing the text, removing HTML tags and markups, removing URLs, removing digits, removing all punctuation, unifying multiple spaces to a single space, tokenizing, removing stop words, stemming, lemmatizing, and joining tokens back into a string. Preprocessing of text other than anonymization was never done for LLMs, since they have built-in mechanisms. Anonymizing text for traditional NLP models was done in order to have similar prompts for both LLMs and traditional NLP techniques. The need for anonymization is discussed in detail later in section 3.2.2. Although it is uncommon to use both lemmatizing and stemming together, our initial tests showed improved metric scores when both were applied compared to using lemmatizing alone, even though lemmatization is generally considered superior to stemming.

3.1.2. FEATURE EXTRACTION

Once the text is cleaned, the next step is to convert it into numeric vectors suitable for machine learning. There are three approaches for this: text encoding (traditional encoding), static word embedding, and contextual word embedding (Campesato 2022). In text encoding, encoded values are calculated directly from the text. Bag-of-Words (BoW), N-grams, and term frequency-inverse document frequency (TF-IDF) are examples of this approach. The downside of text encoding is that it does not capture the semantic or contextual meaning of words. Under the static word embedding approach, one assumes that languages have a distributional structure (Sezerer and Tekir 2021) and calculates dense vector representations of words that capture their semantic meaning. Some of the popular static word embedding approaches are word2vec (Mikolov et al. 2013) and GloVe (Pennington, Socher, and Manning 2014). These approaches are efficient in capturing general semantic relationships between words; however, they are unable to capture contextual meaning. For example, these approaches use the same embedding for the word "bank" in "riverbank" and "financial bank". Contextual word embedding, based on newly developed transformer architectures such as bidirectional encoder representations from transformers (BERT) (Devlin et al. 2019) embedding, considers the context of a word within a sentence before generating the embedding. For example, BERT would generate distinct embeddings for "bank" in "riverbank" and "financial bank" by considering the surrounding words. In this research, we utilized TF-IDF, word2vec, and BERT to convert text into numerical vectors. Each approach is then coupled with classification techniques at the next step.

3.1.3. BERT VS. LLM

In this research, we differentiate between BERT and other LLMs. BERT at heart is a language model; however, compared to today's language models such as GPT-4 and Gemini, BERT is relatively small. It is an open question whether one can surpass the results from LLMs by fine-tuning small language models such as BERT (Bosley et al. 2023). Another big difference between current LLM models and BERT is architecture. BERT is an encoder-only transformer model with the base model using about 110 million parameters.

Table 1. Dataset split (training/test) sizes for each model.

Model Type	Independent Variable	Dependent Variable	N	Training Set Size	Test Set Size
Traditional NLP Technique	Preprocessed_Facts	Decision	14225	11380	2845
Traditional NLP Technique	Preprocessed_Issues	Decision	6103	4882	1221
LLM Models	Anonymized_Facts	Decision	14225	0	14225
LLM Models	Anonymized_Issues	Decision	6103	0	6103

Conversely, most LLM models are decoder-only or encoder-decoder models with parameters surpassing 100 billion. Also, BERT is trained using a masked language modeling approach, while newer LLMs are modeled for generation of the next word. Thus, we separated BERT from other LLMs models used in this study. Also, we used a pretrained version of BERT, `bert-base-uncased`, for our task.

3.1.4. TRAINING VS. TESTING

Under traditional NLP techniques, we split data into training and test sets using the 80:20 rule. However, when we used LLMs, we took the entire dataset for testing. Train and test data sizes are given in Table 1.

3.1.5. HYPERPARAMETER TUNING

To optimize classification performance, we conducted hyperparameter tuning for random forest, gradient boosting, and XGBoost models using the `RandomizedSearchCV` framework. For each algorithm, a specific parameter search space was established to assess key hyperparameters impacting predictive power and generalization. The random forest optimization focused on the number of trees (`n_estimators`), minimum leaf size (`min_samples_leaf`), and the proportions of features and samples (`max_features`, `max_samples`). For gradient boosting, the tuning process considered the number of boosting stages (`n_estimators`), learning rate (`learning_rate`), sample and feature fractions (`subsample`, `max_features`), and maximum tree depth (`max_depth`). The XGBoost model's tuning explored the number of boosting rounds (`n_estimators`), learning rate, regularization (`gamma`), subsampling ratios, and class weighting (`scale_pos_weight`) to address data imbalance.

For all models, `RandomizedSearchCV` was configured with 10 random parameter iterations and fivefold cross-validation. Recall was employed as the scoring metric to prioritize the correct identification of positive cases (employee winning the dispute case). Through parallel computation, the optimal parameter sets and their corresponding recall scores were determined, yielding tuned models that balance complexity, interpretability, and predictive performance. To address class imbalance, the XGBoost model's hyperparameters were tuned separately for two resampling strategies. For oversampling, the model was optimized on a dataset balanced using the synthetic minority oversampling technique (SMOTE). In parallel, another optimization was performed on a dataset that was balanced through undersampling via `RandomUnderSampler`. This dual approach yielded two distinct models, each specifically tuned for its respective data balancing method.

For computing classification metrics, two approaches were used. The first approach used a default probability of 0.5. In the second approach, the trained model was used to predict again on the training dataset, and the F1-maximizing threshold was calculated. This threshold was then used on the predicted test data values to calculate evaluation metrics. The best approach would have been to use a validation dataset to find an F1-optimizing threshold, but, in this case, we have split the data only between training and testing without a holdout dataset for validation.

3.1.6. NON-DETERMINISM IN WORD2VEC AND BERT EMBEDDINGS

Neither BERT nor word2vec embeddings are strictly deterministic, as both the training and the inference processes of these methods have many sources of randomness. Random initialization, stochastic optimizers, and GPU or multi-threaded computations that update model parameters in a different order all contribute small amounts of noise. In practice, this means that rerunning these methods on the same input data can result in slightly different embeddings. To get an estimate of this uncertainty, we repeated the training of word2vec and BERT embeddings 20 times with different random seeds. This results in 20 different representations using word2vec and BERT. The resulting embedding is used as input for the classifier, and we then record the classification metrics: accuracy, recall, specificity, precision, F1-score, and AUC (area under the curve). For both word2vec and BERT, these metrics are computed in two cases: (1) with the default decision threshold of 0.5 and (2) with the F1-maximizing threshold estimated from the training data. The results from 20 runs are averaged for a more robust estimate of the performance for each embedding method. For TF-IDF embeddings, since the representation is deterministic, we do not need to rerun the embedding generation with different random seeds 20 times. We only need to compute the classification metrics with the default threshold of 0.5 and the F1-maximizing threshold. The above calculation was done separately for two data frames where "`Preprocessed_Issues`" and "`Preprocessed_Facts`" serve as independent variables.

3.2. LLM MODEL SETTINGS

3.2.1. REPRODUCIBILITY AND CONSISTENCY

Reproducibility, which is defined as the ability to achieve identical results upon repeating experiments under comparable conditions, is a crucial element for valid scientific research outcomes. However, with LLMs, obtaining reproducible results is a complex endeavor due to the probabilistic nature of the algorithm. Generating consistent outputs from LLMs remains a difficult task because of the many interacting elements involved. The probabilistic framework of LLMs stands as one of their most important characteristics. LLMs produce variations of human-like text through probabilistic predictions of subsequent words, unlike deterministic algorithms which always generate identical results for the same input. The inherent randomness of this process means that the same prompt can produce various outputs. Parameters such as temperature, top P, and top K manage this randomness.

In traditional machine learning algorithms, random seed plays a crucial role in reproducibility. However, even with a fixed random seed, LLMs may produce different outputs due to architectural differences, such as the use of CPU versus GPU. For example, PyTorch, a framework for building deep learning and LLMs, mentions in its documentation that "completely reproducible results are not guaranteed across PyTorch releases, individual commits, or different platforms. Furthermore, results may not be reproducible between CPU and GPU executions, even when using identical seeds" (PyTorch Documentation 2024). The other aspect of LLMs findings is consistency. In other words, how stable or uniform the result is when tests are conducted multiple times. If the answer shows less variation over time, we can conclude that the experiment produced consistent results. In one of the first comprehensive analyses of LLM output consistency and reliability (J. J. Wang and Wang 2025), the authors found that reproducibility and consistency vary by task, showing near-perfect stability in binary classification and sentiment analysis but greater variability in more complex tasks. They also noted that more advanced models do not consistently achieve higher consistency, with task-specific patterns emerging. However, in another study, researchers concluded that the results of Boolean query generation in LLMs are not reproducible (Staudinger et al. 2024), hence bringing no conclusive answer to the question of reproducibility and consistency in LLMs.

In our effort to make this research reproducible, we provided respective code and data sources, set random seeds for libraries, and set certain LLM parameters close to a deterministic level. We also provided the exact prompts we used for several models. We also measured the variability and consistency of predictions by choosing 20% of a sample and then running two LLMs 20 times. Some parameters that introduce variability in LLM are given in Table 2.

3.2.2. LLM AND MEMORY

One of the main questions that arises about LLM models is whether their performance is due to their memory. In other words, did the LLM already see the data we are using in our research beforehand, hence the better result than from a traditional model? In order to address this issue, Kim, Muhn, and Nikolaev (2024) proposed three techniques: anonymizing the text prompt, using the Sarkar and Vafa test (Sarkar and Vafa 2024), and conducting a clean out-of-sample test. Under the anonymization of the text prompt, we remove any contextual information, such as names, dates, location names, monetary units, etc., that the LLM can use to infer the outcome of the worker's compensation case from its training data. By doing so, the LLM has to rely on the general language and context it has learned to infer the outcome rather than any specific historical information about the particular case it may have seen. A sample of anonymized Findings of Fact is given in Table A.1 in Appendix A. Under the Sarkar and Vafa test, given the anonymized workers' compensation case data, we asked the LLM to predict the case number, year, plaintiff name, and defendant name. For conducting this research on a clean out-of-sample, which means finding workers' compensation cases that happened after LLMs versions were released, we were unsuccessful, primarily due to the frequency of LLM updating.

3.2.3. LLM, PROMPTS, AND ZERO-SHOT LEARNING

The performance of LLMs highly depends on the provided prompt. Because the result of an LLM can change dramatically with a different input, we examined the sensitivity of our final predictions by using a diverse set of 17 prompts from three different prompt families. The architecture of this strategy is illustrated in Figure 2. In this context, a prompt refers to the text input that is used to guide the output of the model for a specific task. Thus, designing effective prompts is essential for achieving good performance, and this has attracted a lot of attention (Santu and Feng 2023). Influential works have explored strategies for "designing appropriate" prompts (Ahn et al. 2022; Kojima et al. 2022; Webson and Pavlick 2022), while detailed surveys of the field can be found in Schulhoff et al. (2024) and Huttula (2025).

The first set of prompt families we examine are taken from Ahn et al. (2022). Their paper explores the use case of an LLM providing high-level linguistic instructions to an agent (a robot acting as the "hands and eyes" of the LLM). In order to provide a controlled comparison of their instructions to PaLM-SayCan system, the authors created a benchmark of 101 prompts organized into seven distinct prompt families. We have created one prompt for each of the seven families in Table A.13 for our experiments. The second set of prompts we examine are taken from Kojima et al. (2022). In this paper, they discovered that the simple addition of the phrase "Let's think step by step" resulted in a huge accuracy gain for LLM predictions even when using zero-shot learning. They also attempted the prompt families from Ahn et al. (2022) but found that "Let's think step by step" significantly outperformed them. We used five "Let's think..." variations for this research, and these are given in Table A.14. Finally, we look at Webson and Pavlick (2022). The authors pose the question "Do prompt-based

Table 2. Parameters for fine-tuning LLMs.

Parameter	Description	Reproducibility
Model	LLM model used.	Variations of each model and versions and configurations can lead to different behavior of reproducibility.
Temperature	Controls the randomness of token sampling. Higher values increase randomness and creativity.	Set to 0 for maximum determinism.
Top P	Filters tokens based on cumulative probability. Higher values allow more diverse outputs.	Set to 0 for maximum focused output.
Top K	Limits token selection to the top K most probable tokens. Higher values allow more options.	Set to 1 for highly predictable output.
Seed	Initial value for the pseudorandom number generator.	Set to a fixed seed value for consistent results across runs.

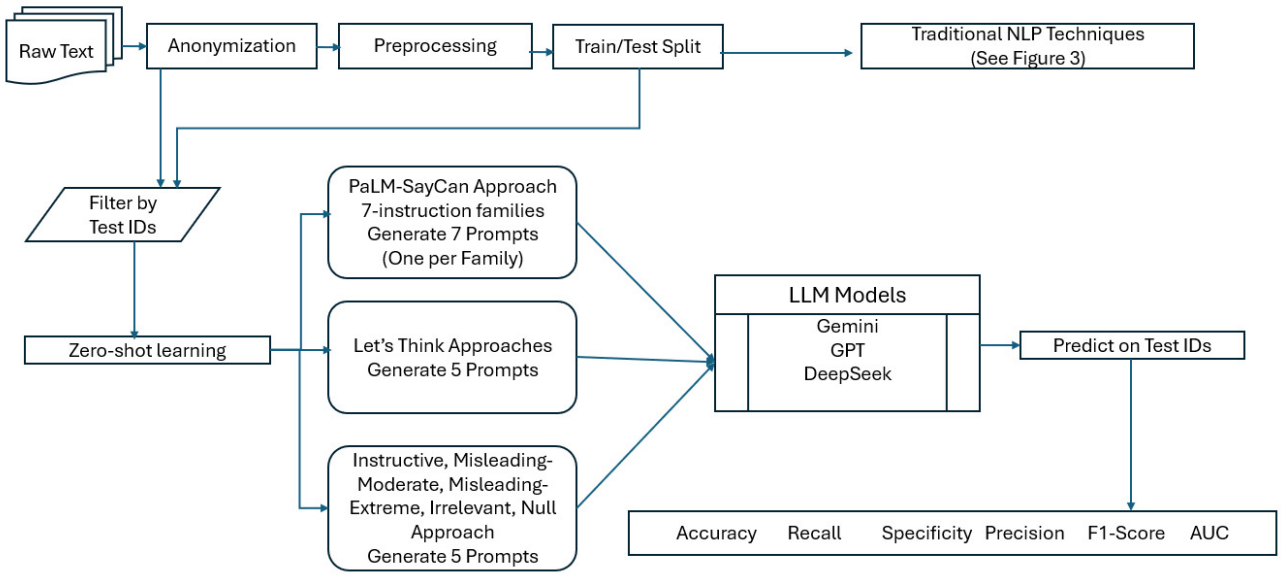

Figure 2. Prompting strategies for measuring robustness.

models really understand the meaning of their prompts?" and discover that models can learn just as quickly with a large number of prompts that are irrelevant or misleading as with similar "good" prompts. They created five distinct prompt templates, and we used five prompts, one from each template. See Table A.15 for prompts we created from this family.

Prompts are commonly categorized into "simple prompts" and "chain-of-thought (CoT)" prompts. A simple prompt is a single instruction or question to the LLM that requests the final answer and provides no context for the reasoning that led to that answer. By contrast, CoT prompting guides the model to articulate the series of logical steps taken to reach a conclusion (Wei et al. 2022). In a highly cited publication by Wei et al. (2022), researchers from the Google Brain team have demonstrated how CoT prompting can improve LLM performance. CoT prompting without step-by-step examples, using only a lead-in phrase like "Let's think step by step," can also lead to significant increases in accuracy (Kojima et al. 2022; Jeoung et al. 2025). We have categorized each prompt in Tables A.13, A.14, and A.15 as either a simple prompt or a CoT prompt.

Zero-shot learning (simply referred to as zero-shot in the LLM context) tasks the model to do a job without giving it any examples to learn. Few-shot learning gives the model several examples of how to do a task, complete with step-by-step reasoning to arrive at the correct answer. CoT prompting is usually done in combination with a few-shot example, but it also works in a zero-shot setting. This is achieved by simply adding instructional phrases like "go step-by-step through each fact" to the prompt, which triggers the model's reasoning process without explicit examples. In this research, we exclusively used zero-shot strategy, employing either zero-shot simple prompts or zero-shot CoT prompts. Unless we check for prompt robustness, all other LLM-based calculations are done under the unique standard simple prompt given in Table 3.

Table 3. Standard simple prompt.

Simple Prompt (Standard)	Analyze the following legal case facts. Based solely on these facts, predict whether the plaintiff likely won or lost the case. 1. Respond ONLY with the number 1 if the plaintiff likely won. 2. Respond ONLY with the number 0 if the plaintiff likely lost. Do NOT provide any explanation, commentary, or any text other than 1 or 0.

3.2.4. *LLMS AND MODEL TYPE*

In this research, we used the following LLM models: `deepseek-chat` (`deepseek-v3`), `claude-3-haiku`, `gemini-1.5-pro`, `gemini-1.5-flash`, `gemini-2.0-flash`, `gpt-3.5-turbo`, `gpt-4.1-mini`, and `o4-mini`. Model characteristics are given in Table A.2.

4. MAIN RESULTS

This section is dedicated to presenting the primary findings derived from this research. Given the intrinsic nature of the problem as a classification task, a comprehensive set of evaluative metrics has been employed. Specifically, the performance of the proposed model is assessed using accuracy, recall (sensitivity), specificity, precision, F1 score, and the AUC score. The interpretation of these metrics under the research context is provided in Table A.24.

4.1. PERFORMANCE COMPARISON OF LLMS AGAINST TRADITIONAL NLP TECHNIQUES

Tables A.3 and A.4 display how traditional NLP technique approaches predict "Decision" results in workers' compensation cases through analysis of "`Preprocessed_Issues`" or "`Preprocessed_Facts`" variables at default threshold of 0.5. Under the default threshold of 0.5, it is surprising to see simpler TF-IDF embedding approach coupled classifiers outperforming the other two advanced embedding-based classifiers most of the time. The gradient boosting method achieves a perfect recall score of 1.0 when combined with TF-IDF embedding techniques. In general we can see very high recall value for word2vec embedding coupled classifiers as well as most of the TF-IDF coupled classifiers. Recall answers the question: "Of all the actual plaintiff-winning workers' compensation disputes, how many did the model correctly identify?" A high recall value means the model is capturing most of these cases.

However, because our dataset is imbalanced, with more plaintiff-winning cases than losing ones, recall alone can be misleading. For example, a model that always predicts the plaintiff wins would achieve a perfect recall of 1.0 but provide no useful discrimination. Yet, in practice, the stakeholders of this research: employees, employers, insurance companies, and analysts need balanced insights about both wins and losses. Here, specificity complements recall by answering: "Of all the actual plaintiff-losing cases, how many did the model correctly identify as losses?" High specificity ensures the model is not simply overpredicting wins and helps prevent unnecessary payouts by ensuring insurance companies do not mistakenly assume the employee will win the case.

Recall and specificity cannot both be maximized at the same time, because as the decision threshold changes, improving one typically reduces the other. Additionally, precision may be important to ensure predicted wins are reliable. Thus, metrics such as AUC (capturing the trade-off between recall and specificity across thresholds, hence overall discrimination) and F1 score (balancing recall and precision) are more informative. With this in mind, when we looked into Tables A.3 for `Preprocessed_Issues` and A.4 for `Preprocessed_Facts`, TF-IDF coupled with random forest always gives the highest AUC. For F1 score, TF-IDF coupled with XGB and TF-IDF coupled with random forest are highest. However, when performance is considered from a broader recall–specificity balance perspective, the more reliable choices appear to be TF-IDF with XGB (with undersampling applied) and BERT embeddings with SequenceClassification, regardless of the independent variable we used. Instead of using the default threshold, if one uses the F1 maximizing threshold, calculated from the training data to evaluate metrics on the test data, the summary metrics can be found in Tables A.9 and A.10. Even though AUC does not change in this scenario, other metrics do differ from those calculated at the default threshold. However, TF-IDF–based classifiers still show the highest values for specificity, precision, and AUC when `Preprocessed_Facts` are used. In this scenario, word2vec-based classifiers outperform other classifiers in terms of accuracy and recall.

The comparison of the effect of the independent variable and the thresholding method on the performance of the traditional NLP technique is shown in Figure 3. It indicates that `Preprocessed_Facts` outperforms `Preprocessed_Issues` when focusing on the AUC metric. For the other metrics, in general, `Preprocessed_Facts` performs slightly better than `Preprocessed_Issues`. Regarding the thresholding method, the F1-maximizing threshold outperforms the default threshold when the concern is accuracy, specificity, and precision. But for the other metrics, the result is less clear cut.

LLMs demonstrate significantly reduced predictive power when `Anonymized_Issues` are used as the independent variable; in fact, none of the tested LLM models perform well in this scenario. However, LLM models show a significant improvement in performance when `Anonymized_Facts` are used as input data. Of the eight LLM models tested, the `deepseek-chat` model performs better than all others in every category, except for specificity. For specificity, the `claude-3-haiku-20240307` model performs better than all other models considered. It's worth noting that LLMs perform better than traditional NLP techniques when `Anonymized_Facts` are used with the standard simple prompt. This is evident by comparing the average of model metrics for LLMs given in Tables A.5 and A.6 against the average of model metrics for traditional NLP techniques under different thresholds given in Tables A.3, A.9, A.4, A.10, as well as through Figure 4.

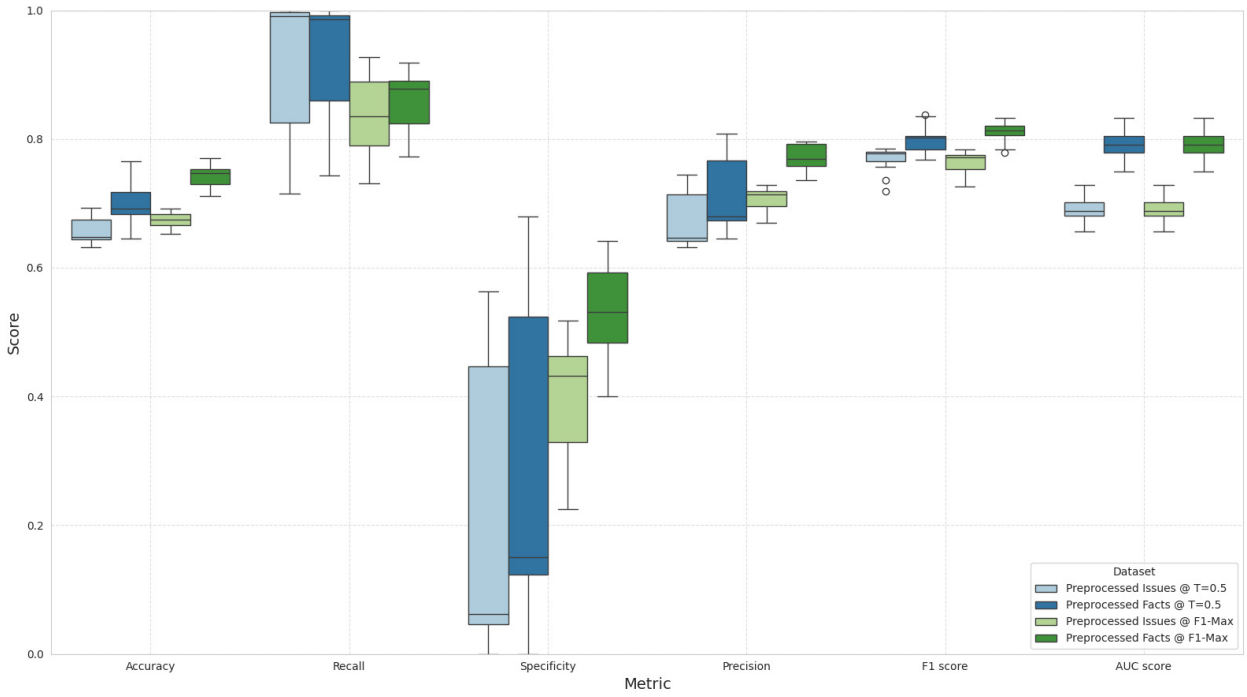

Figure 3. Impact of the independent variable (Issues vs. Facts) and thresholding on traditional NLP model performance.

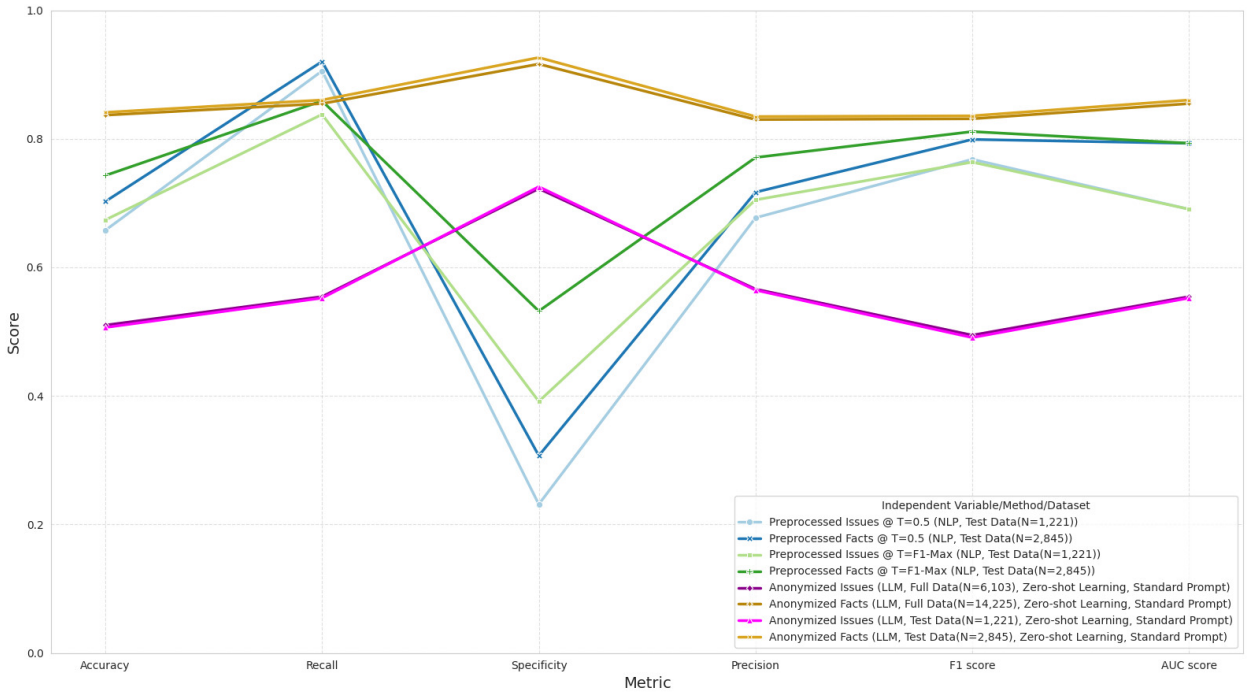

Figure 4. Overall mean performance comparison of NLP and LLM techniques.

The outcomes on the test data for the traditional methods may exhibit greater variability compared to those for the LLMs, since in Tables A.5 and A.6 the LLMs are evaluated on the entire dataset under the zero-shot learning approach, as explained in Figure 1. In contrast, the traditional NLP methods rely on a smaller test subset, which introduces more random variation in their results. The simplest way to address this concern is to show the results of the LLMs on the same test dataset as the traditional NLP methods. Thus, as explained in Figure 1, we tested the LLMs again using only the same test dataset used by the NLP techniques. The results are given in Tables A.11 and A.12 and Figure 4. This indicates that the LLMs can perform at the same level regardless of the dataset size.

4.2. PROMPT ROBUSTNESS

As explained in Section 3.2.3, different prompts may yield different results. However, for actuaries and professionals in related fields, consistent and reliable outputs are essential. This requires two types of investigation. On the one hand, we need to understand how much the results deviate when using different prompts with the same LLM. On the other hand, we need to examine how much the results deviate when using the same prompt and the same LLM but providing the same data entry repeatedly.

To answer the first question, we used three different prompt families and 17 different prompts drawn from those families as shown in Tables A.13, A.14, and A.15 on LLM models `gemini-1.5-flash-002`, `gpt-4.1-mini`, and `deepseek-chat`. Since we have observed that, regardless of dataset size, LLM models produce nearly identical results (see Figure 4), for this test, we used only the anonymized test dataset for Issues and Facts, respectively. The results are given in Tables A.16–A.21. Figures 5 and 6 present boxplots comparing the distributions of performance metrics across prompts for LLMs on `Anonymized_Facts` and `Anonymized_Issues`. These results are evaluated on the same test set as the traditional NLP techniques; however, unlike the NLP models, the LLMs are tested using anonymized data only (not anonymized + preprocessed). From these figures, it is evident that the `gemini-1.5-flash-002` and `gpt-4.1-mini` models exhibit less variation in performance metrics when `Anonymized_Facts` are used as the independent variable. The standard prompt we employed also appears to be reasonably optimal for both `gemini-1.5-flash-002` and `gpt-4.1-mini`. In contrast, for `deepseek-chat`, the standard prompt performance often falls within the first quartile, and in some cases at the minimum, illustrating that there is no universal prompt that performs well across all LLMs. Interestingly, the LLMs simultaneously achieve higher recall and specificity, which contrasts with the trade-off typically observed in traditional NLP techniques. Furthermore, when `Anonymized_Issues` are used as the predictor, model performance declines sharply and exhibits greater variability across the 17 different prompts for accuracy, specificity, and F1 score.

To compare LLM model performance across 17 different prompts against 11 traditional NLP techniques, we present boxplots of each LLM model alongside the NLP techniques. The results are shown in Figures 7 and 8. It is evident that, despite the variability introduced by 17 different prompts, LLM models consistently outperform traditional NLP techniques in accuracy, specificity, precision, F1, and AUC metrics when `Anonymized_Facts` are used as predictors. In contrast, when `Anonymized_Issues` serve as predictors, traditional NLP techniques consistently achieve superior performance.

To answer the second question, whether model performance changes when the same LLM, prompt, and entries are run multiple times, we used 20% of anonymized Findings of Fact (2,845 records) and 20% of anonymized Issues (1,220 records) to simulate case outcomes 20 times. We selected `gemini-1.5-flash-002` and `gpt-4.1-mini` for this experiment due to their cost-effectiveness and speed, along with the standard prompt. We performed sampling once, then ran models 20 times on the same dataset repeatedly. Results are shown in Tables A.7 and A.8. The tables indicate that the standard deviation for `gemini-1.5-flash-002` is 0.0000, showing identical results when the same question is asked repeatedly with the same prompt. For `gpt-4.1-mini`, the standard deviation is nonzero but very small, again implying consistent outputs for the same entry, same prompt.

4.3. PROMPTING STRATEGY COMPARISON: SIMPLE VS. CHAIN-OF-THOUGHT IN LLM PREDICTIONS

Prompts we created in Tables A.13, A.14, and A.15 can be divided into CoT prompts and simple (direct) prompts. We have indicated which prompts we consider CoT prompts and which ones are simple prompts. The logic behind this categorization is whether the prompt explicitly asks LLMs to go step by step to achieve its conclusion in some form. If not, then the prompt was classified as a simple prompt. We understand that some of these prompts may feel like CoT prompts. This categorization resulted in 11 simple prompts and six CoT prompts. When comparing the metrics for CoT versus simple prompts, it appears that there is little difference, yielding an unexpected result. This can be seen by comparing the boxplots in Figures 9 and 10.

One possible reason is that the models' internal reasoning doesn't align with human judgments. In other words, what LLMs interpret as a "win" or "loss" through internal "thought" process may differ from the official case outcomes labeled by humans. This poses an interesting question: Who is right? Did the human label these cases correctly, or are LLM models correct? Given the cost constraint, we were only able to use `deepseek-chat`, `gemini-1.5-flash-002`, and `gpt-4.1-mini` for the prompt robustness analysis in the previous section and the CoT versus simple prompt comparison here. The result shows that CoT may not be a suitable prompting strategy for workers' compensation case studies. Another reason there is little difference in the metrics for CoT versus simple prompts may be that the CoT prompt used by the researchers is not suitable and may need tweaking to achieve good results. It should be noted that the `04-mini` model internally uses the CoT approach to make a decision. Thus, even though we used a simple prompt externally, some form of CoT was involved to produce the output.

4.4. MODEL PREDICTIONS OF CASE OUTCOMES: ISSUES VERSUS FINDINGS OF FACT

In this research, we used two different independent variables to predict the case outcome. Interestingly, with traditional NLP techniques, on average, there was little difference in accuracy, recall, specificity, precision, F1 score, and AUC score regardless of whether we used issues or finding of facts as the independent variable. However, for LLM models, when findings of fact were used, on average all metrics improved across models. One possible reason is that findings of fact contain more information. They are more formal and imitative of the legal findings presented

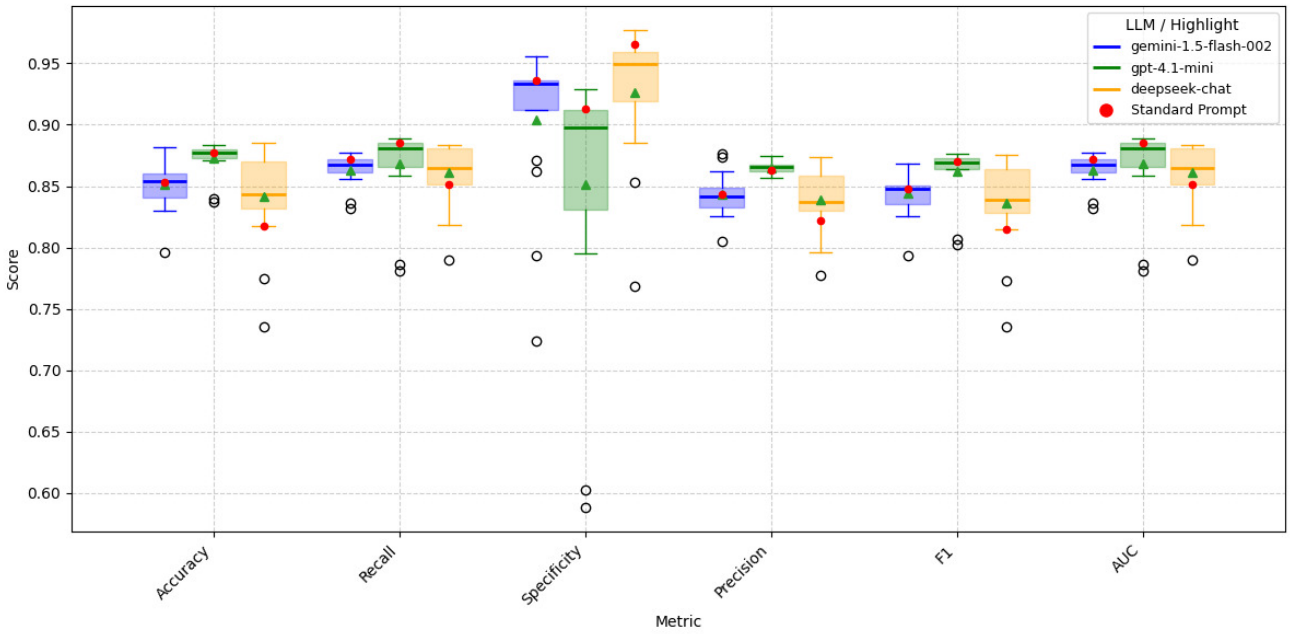

Figure 5. Performance metric distributions across prompts by LLMs for Facts.

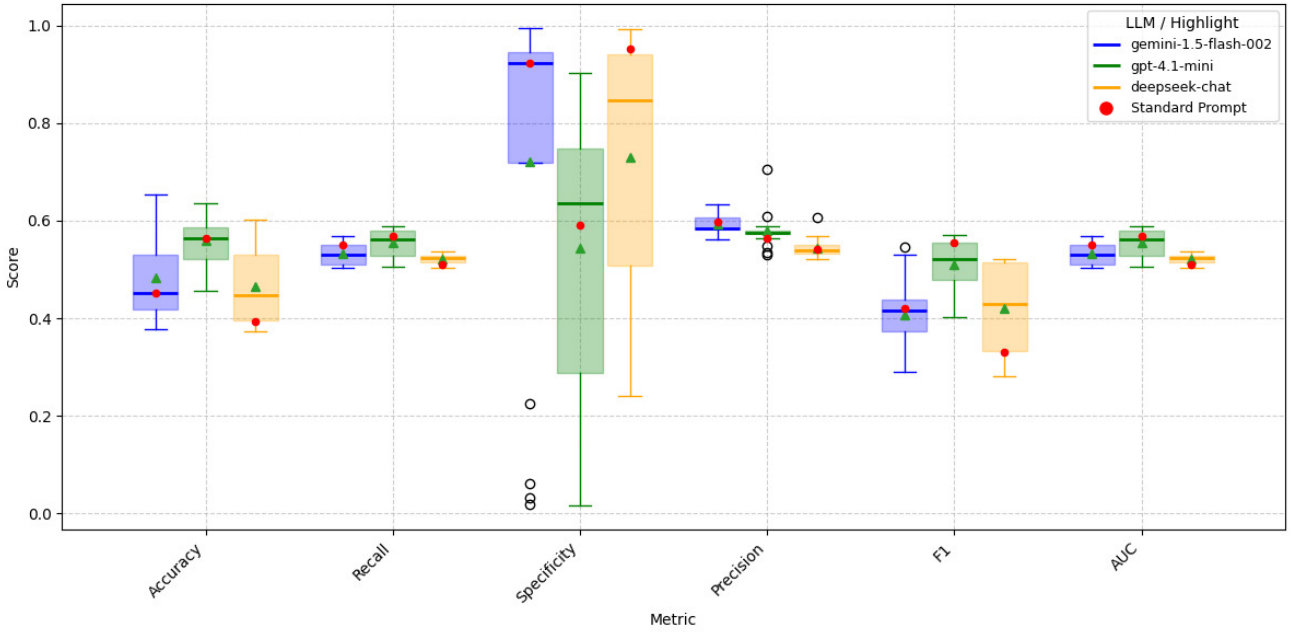

Figure 6. Performance metric distributions across prompts by LLMs for Issues.

in court. Thus, when one reads these findings of fact, they may get a sense of what is likely to happen during the final decision phase. In contrast, issues are items brought up at a very early stage of the case and may not reveal what the next step will be. In essence, the rich textual context in findings of fact may enable LLM models to predict outcomes more accurately than when using issues.

4.5. THE LLMS TRAINING MEMORY REGARDING THE DATASET

One key question about LLM performance is whether it stems from the model having seen the data during its training phase. Thus, for `gemini-1.5-flash-002` and `gpt-4.1-mini`, we used the following prompt to verify whether the model could infer the case number, year, plaintiff's name, and defendant's name.

```
You are an expert at extracting case information from
legal summaries. Based solely on your internal
training data and knowledge (no web search),
```

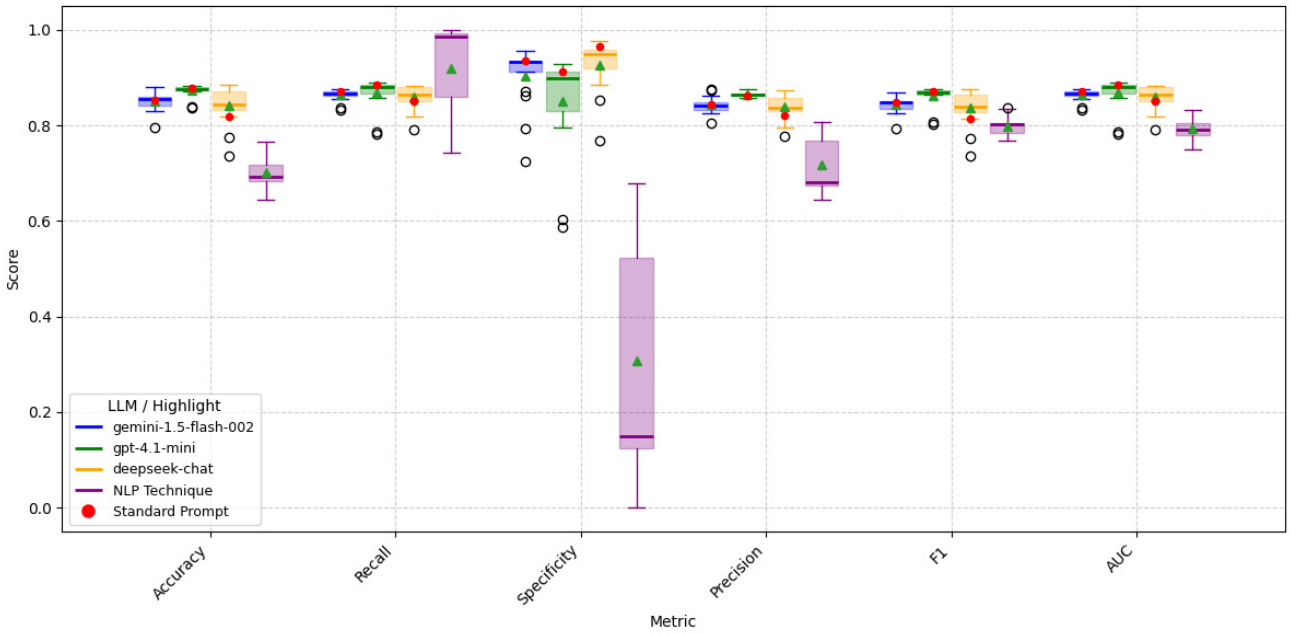

Figure 7. Performance metric distributions across prompts by LLMs for Facts, compared with traditional NLP techniques.

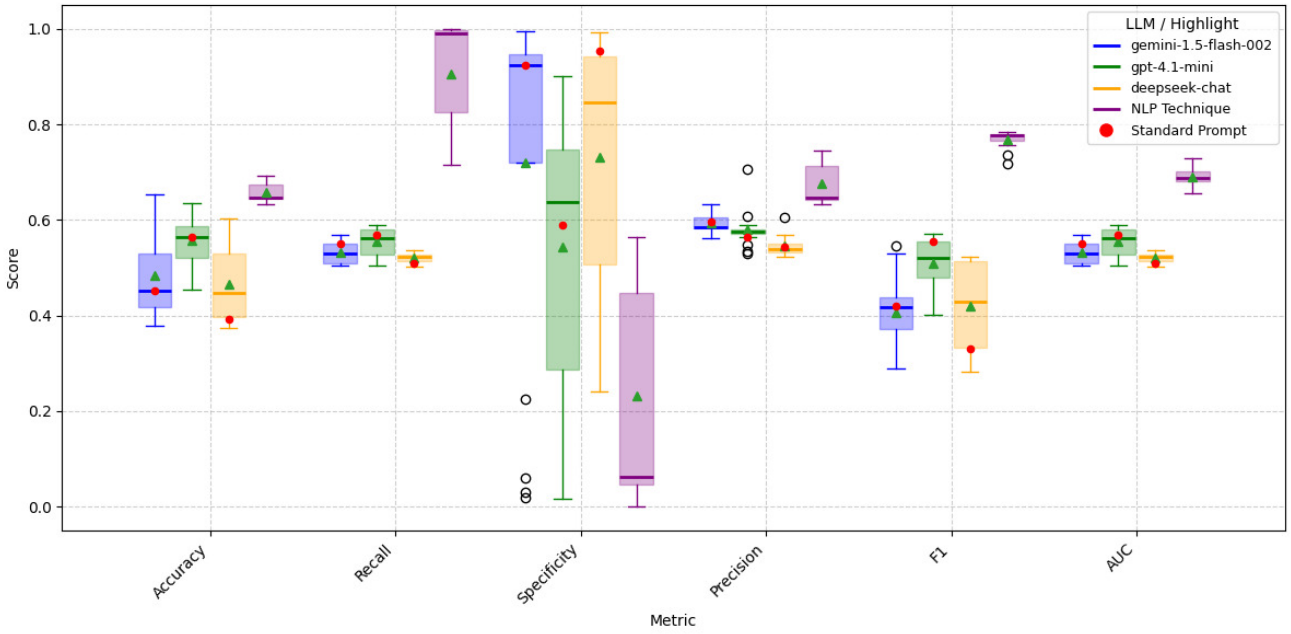

Figure 8. Performance metric distributions across prompts by LLMs for Issues, compared with traditional NLP techniques.

```
identify the following for each anonymized
workers' compensation case:
''Case_ID'': (string, predicted Case ID, or ''Unknown''
    if not found)
''Year'': (string, predicted Year the case was heard,
    or  ''Unknown''  if not found)
''Plaintiff_Name'': (string, predicted Plaintiff's
    Name,  or ''Unknown'' if not found)
''Defendant_Name'': (string, predicted Defendant's
    Name,  or ''Unknown'' if not found)

If a piece of information is explicitly stated as
anonymized or cannot be confidently extracted, use
''Unknown'' for that specific key. Your  response MUST
be a JSON object and contain ONLY the JSON object.
Do NOT include any other text or explanation.
```

We used 20% of the data for this test (Sarkar and Vafa test) and concluded that `gemini-1.5-flash-002` and `gpt-4.1-mini` do not retain memory of the workers' compensation cases used in this research. Hence, we generalize to other models, without testing them, due to cost constraints, that LLMs' memory is not a significant contributing factor to the findings of this research. The Sarkar and

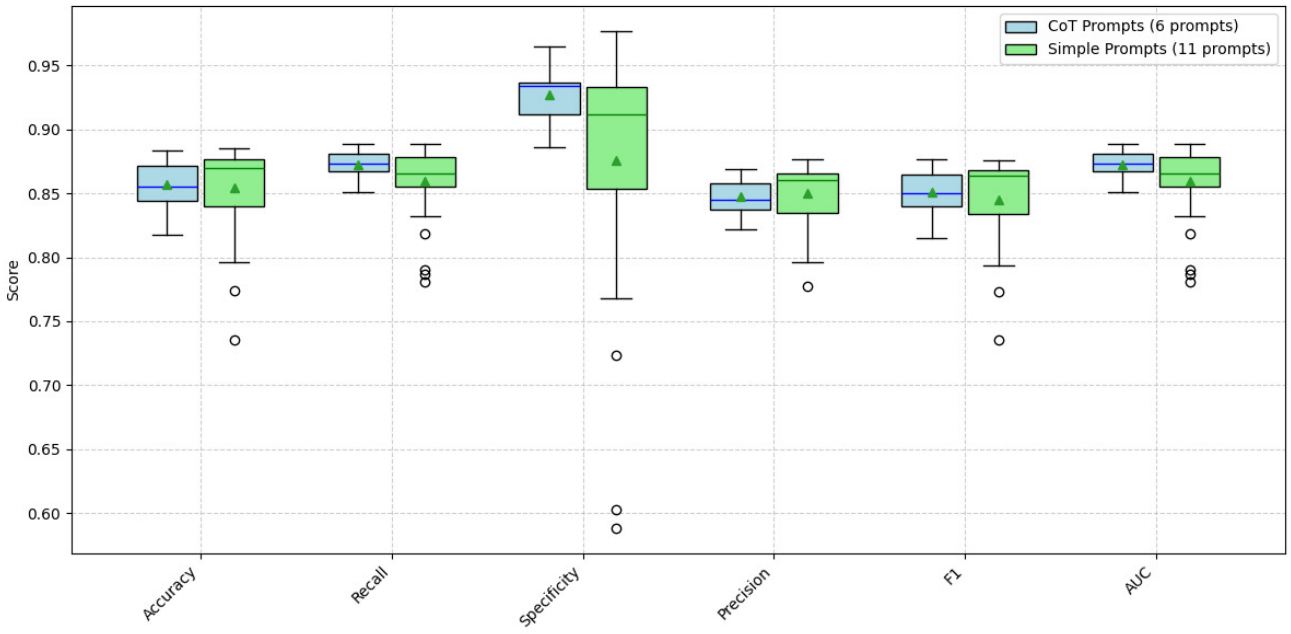

Figure 9. Metric distributions: CoT vs. simple prompts for Facts.

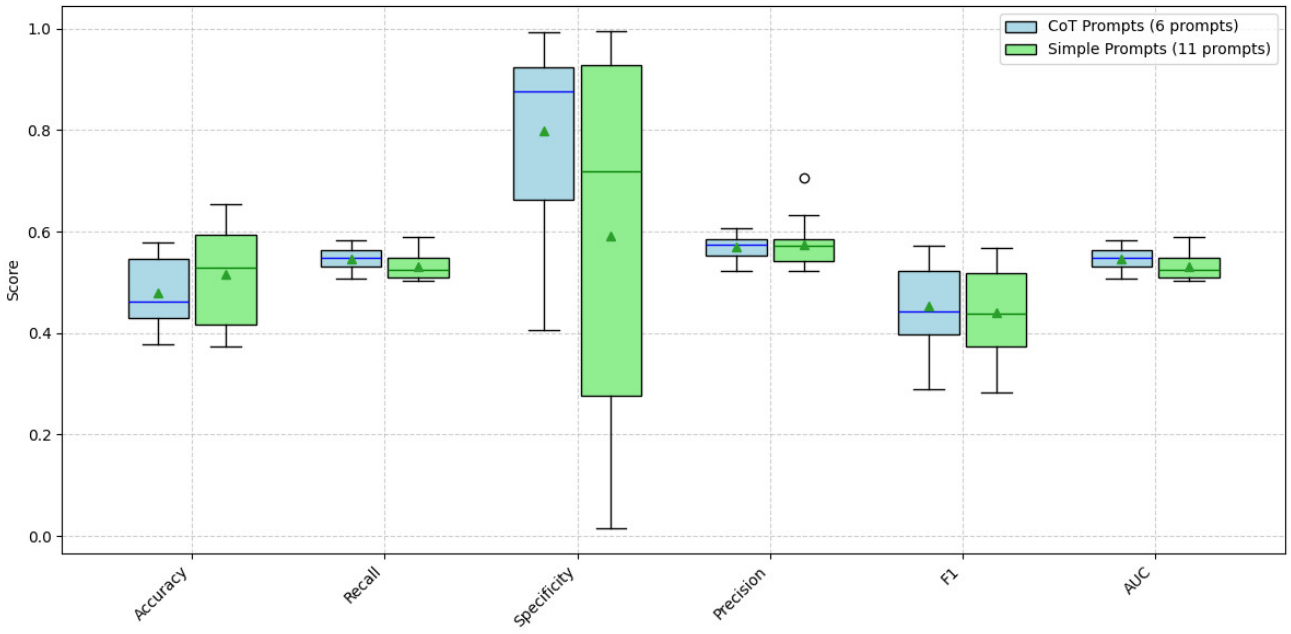

Figure 10. Metric distributions: CoT vs. simple prompts for Issues.

Vafa test was applied to 2,845 (20%) randomly chosen rows of anonymized Findings of Fact and 1,220 (20%) randomly chosen rows of anonymized Issues. The outcome revealed that, given anonymized Findings of Fact or Issues, LLMs could not predict the Case_ID, year, plaintiff name, or defendant name.

4.6. ANONYMIZED VERSUS PREPROCESSED PREDICTORS

In this research, we used anonymization only for LLMs and anonymization followed by preprocessing for the traditional NLP technique. A question may arise whether doing additional preprocessing before we enter workers' compensation dispute data into LLMs would yield superior results. To test that, we followed the methodology explained in Figure 11.

The results can be found in Tables A.22 and A.23. They show that, in general, except for specificity, anonymization performs better than anonymization followed by preprocessing for LLMs. This surprising result indicates that LLMs may utilize extra information that we remove during the preprocessing step. Figures 12 and 13 clearly indicate that, except for specificity, all other metrics perform better when just anonymization is used. Smoothing/cleaning of text

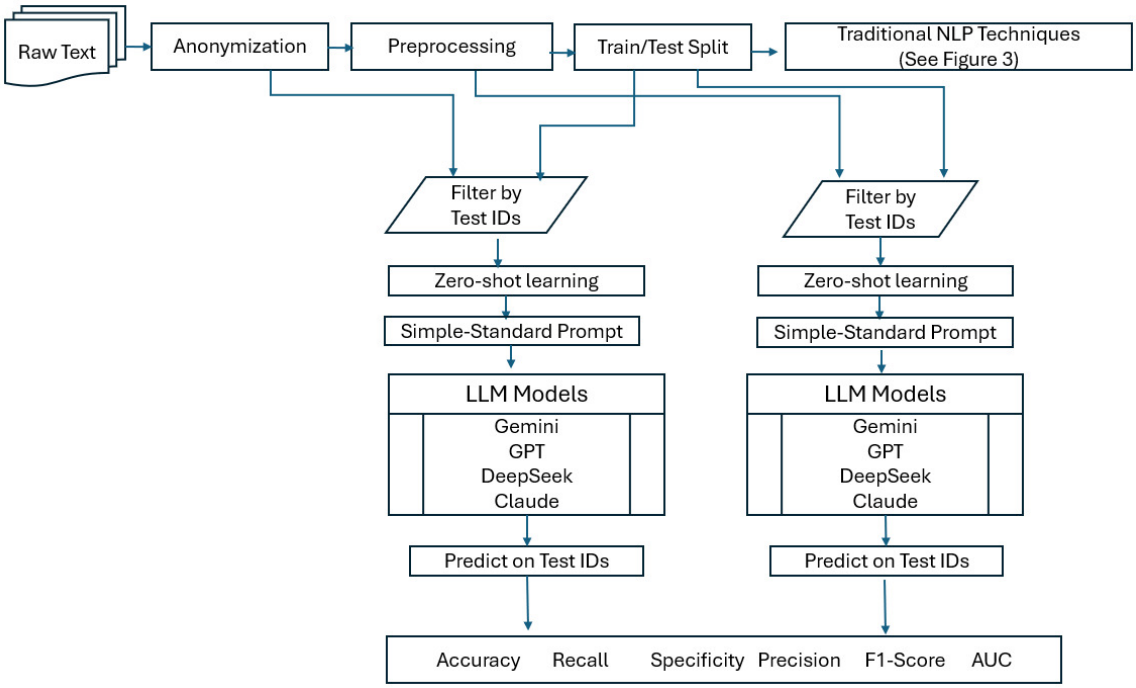

Figure 11. Methodology for testing the impact of preprocessing on LLMs.

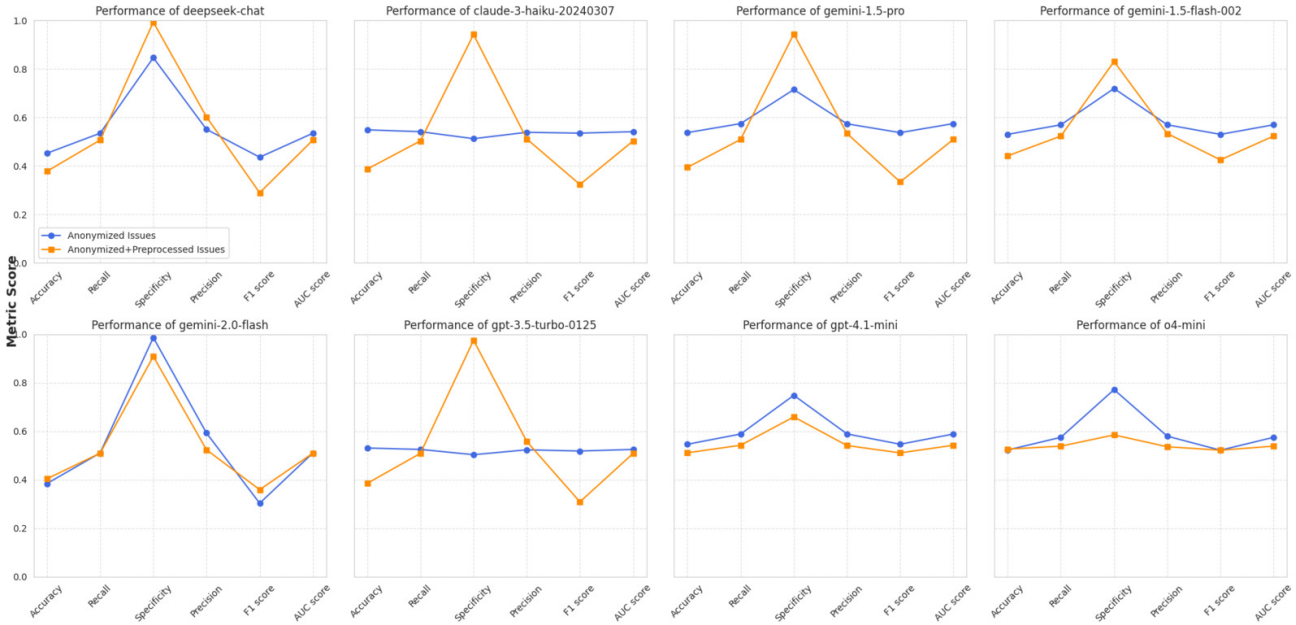

Figure 12. Model performance comparison for anonymization versus (anonymization + preprocessing) for Issues.

through preprocessing may not work well in the workers' compensation case arena, according to this result.

Anonymization presents a major obstacle for legal LLM research. Legal text must be sanitized to mitigate reidentification risks. It must also retain contextual and semantic information for meaningful and accurate analysis, and it must also use the US legal system's reliance on case law. Therefore, it requires more sophisticated techniques than the simple text redaction or masking used in this research. To understand why anonymization degrades LLM performance so significantly, it is crucial to understand the two primary knowledge sources an LLM utilizes: parametric knowledge (PK), which is stored and learned during pretraining, and contextual knowledge (CK), which is supplied at inference time through the prompt or context window (Cheng et al. 2024). Even though LLMs contain massive PK, empirical research found that LLMs overwhelmingly prioritize CK, even to the extent of suppressing their own parametric knowledge (Tao et al. 2025). This leads to a phenomenon known as PK suppression, where the model ignores its internal knowledge when CK is available, even if that CK is

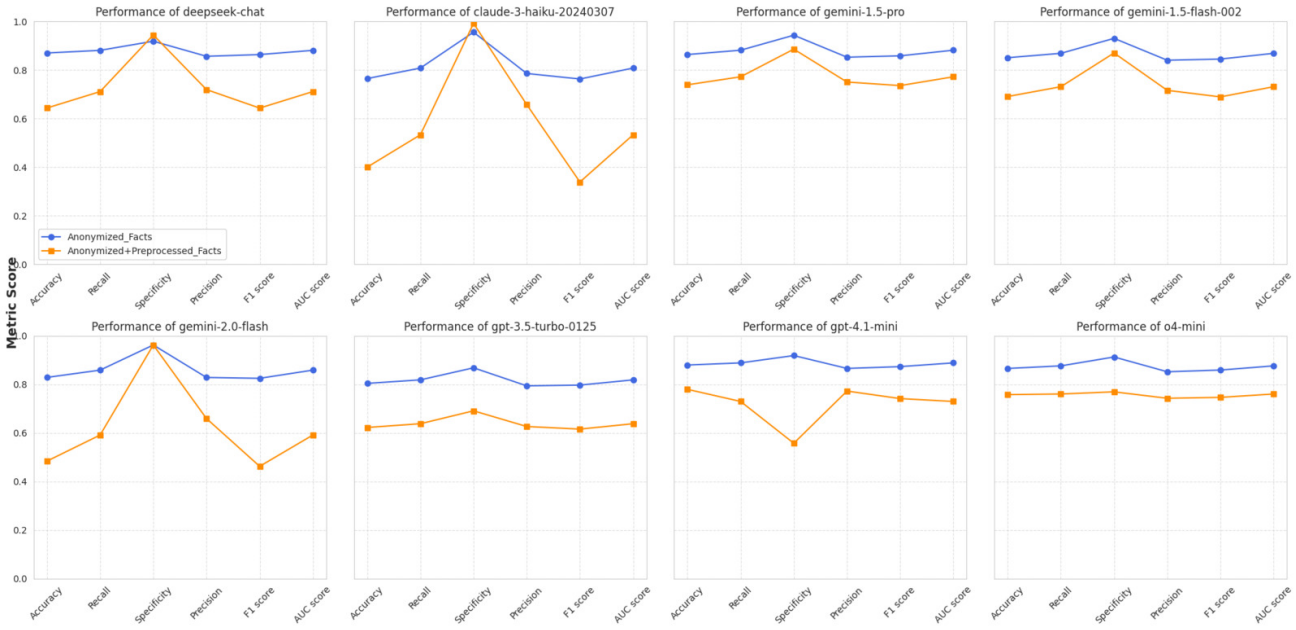

Figure 13. Model performance comparison for anonymization versus (anonymization + preprocessing) for Facts.

merely complementary or irrelevant, according to Tao et al. (2025).

For our research, this means the model relies on the immediate context of a workers' compensation case file provided through prompt rather than its broader, pretrained understanding of legal principles. The problem is especially acute in US law, where case names, citations, courts, and factual details are not just metadata but essential components of legal reasoning. Under stare decisis (the legal principle that courts should follow past judicial decisions, or precedents, when making future rulings on similar cases), no precedent can be judged binding or persuasive without knowing its jurisdiction and court level. Thus, naive anonymization that redacts parties or case identifiers severs the logical links needed for reasoning, making it harder for LLMs to assess outcomes meaningfully.

4.7. WHO IS CORRECT?

Another question we can raise is whether humans made the correct labels for workers' compensation cases. Since we assume human decisions as absolute truth and evaluate traditional NLP techniques and LLMs against them, it is crucial to know whether the human decision is correct. However, in many workers' compensation cases, deciding whether the plaintiff won or lost is not straightforward and may be subjective. Thus, the human decision may not be correct. If we assume each AI model acts like a human, we can determine the majority decision across AI models. Once we have this majority decision, we can compare it to the human decision and measure the difference. For this, we used anonymized Findings of Facts and the simple prompting strategy, with model outcomes in Table A.6. Using model predictions for these models, we calculated the majority decision for each case and then checked it against the human decision. Only 2,024 cases out of 14,225 had a different LLM-majority decision than the human decision, equivalent to 14.23%, where the majority of eight LLMs (`deepseek-chat`, `claude-3-haiku`, `gemini-1.5-pro`, `gemini-1.5-flash`, `gemini-2.0-flash`, `gpt-3.5-turbo`, `gpt-4.1-mini`, `o4-mini`) disagreed with the human decision on who won the workers' compensation. When we checked the actual awards for a few of these mismatched cases, it appears that the human decision was correct. However, we have not reviewed all 2,024 mismatched cases individually.

5. CONCLUSION

In this research, we were interested in finding whether off-the-shelf LLMs can outperform other specialized NLP techniques in predicting likely outcomes for workers' compensation cases. We used two different independent variables, namely "Findings of Fact" and "Issues," to predict the case outcome "Decision." These outcomes were then compared against the human decisions. We have used TF-IDF, word2vec, and BERT as our embedding approaches for traditional NLP techniques. These text-embedding approaches, coupled with random forest, gradient boosting, XGBoost, and a BERT Classifier (based on neural networks), serve as our traditional NLP techniques. For LLMs we have used eight models: `deepseek-chat`, `claude-3-haiku`, `gemini-1.5-pro`, `gemini-1.5-flash`, `gemini-2.0-flash`, `gpt-3.5-turbo`, `gpt-4.1-mini`, and `o4-mini`.

We first found that deterministic, simple TF-IDF classifiers either outperform or are on par with modern embedding–based classifiers, regardless of the predictor used. This is surprising but evident in Table A.3, Table A.4, and Figure 3. Next, our research shows that when `Anonymized_Issues` is used, LLMs' performance becomes worse than that of traditional NLP techniques. However, when we use `Anonymized_Facts`, LLMs' performance significantly improves and is either on par with or better than the

traditional NLP techniques. LLM models show a better balance between recall and specificity compared to the traditional NLP model; Figure 4 illustrates this clearly. We have also noticed that the BERT model coupled with the BERT classifier did not outperform other traditional NLP techniques, except for accuracy, when "Issues" was used as the independent variable.

We have also found that the CoT prompting strategy did not improve the results as we expected. This is a curious outcome and needs further research to determine whether our CoT prompt requires modification or should move from zero-shot prompting to few-shot prompting, so LLMs have more avenues to learn the CoT approach. We have used 17 different prompts from three different popular prompt families. Of these prompts, 11 can be considered as simple, direct prompts, and the other six can be considered as CoT prompts. Figures 5 and 6 show the prompt robustness across three different LLMs from three different manufacturers. These figures show tight boxplots for each metric except for specificity. They also show that the standard prompt we used is either in the third or the fourth quartile for the Gemini and GPT models, performance-wise, but not for the DeepSeek model. Given the tight nature of the boxplots for each metric except specificity, we conclude that LLMs can be used by actuaries for workers' compensation dispute outcome determination at a high level of accuracy, recall, precision, F1, and AUC (generally above 0.8) when coupled with `Anonymized_Facts`. When prompts are divided between CoT and simple prompts, it is clear that CoT did not improve the model metrics. This is clear from Figures 9 and 10.

Using the Sarkar and Vafa test, we found that these LLM models do not retain memory of the cases we tested, which lends significant validity to our research outcomes. It also shows that the models' predictions are consistent, making them potentially useful for predicting future workers' compensation cases. We found that LLMs produce consistent outcome with very small variability when the same prompt, LLM model, and entry are used. Thus LLMs decisions on workers' compensation do not change with different runs. This is evident from the results given in Tables A.7 and A.8. We also tested whether anonymization alone or anonymization followed by preprocessing performs better for LLMs. From Figures 12 and 13 it can be seen that anonymization outperforms anonymization followed by preprocessing. This allows practical use of LLMs in industry scenarios, where less data preparation is needed to use off-the-shelf LLMs. We also observed that the performance of the LLM model remains unchanged with varying dataset sizes. This was evident when we tested LLM models on smaller test datasets, with results shown in Tables A.11 and A.12. This is promising for actuaries, indicating that LLMs can be leveraged for these tasks off the shelf.

Next, we tested whether humans made incorrect decisions on the cases. It turned out that only 14.23% of cases had decisions that did not agree with the majority of LLM decisions. Upon reviewing a few such cases, it seems that the human decision was correct. Thus, human decision-making still surpasses LLM decisions in determining who won workers' compensation cases.

This research highlights that LLMs can be used proactively by actuaries, insurance companies, plaintiffs, and defendants to decide the likelihood of winning their cases if they have sufficient information on Findings of Fact. However, if they know only a few facts such as Issues, then the use of LLMs is not desirable. Finally, we conclude that LLMs outperform traditional NLP techniques in deciding who likely won workers' compensation disputes when a large context window is provided to the LLMs. The finding is significant given that a layperson cannot use traditional NLP techniques without domain-specific knowledge, but LLMs do not require such knowledge. Also, LLMs are not trained to predict workers' compensation case outcomes, whereas specialized NLP techniques are. Hence, the fact that just the base LLMs—without any modification—can perform on par with traditional NLP techniques specially trained to predict workers' compensation cases proves our hypothesis that LLMs outperform traditional NLP techniques when the context window is sufficiently large and enriched.

We agree that, in some instances, the most immediate benefit appears to be ex post, that is after a claim is disputed and filings are made. However, even in this scenario, our work shows that LLMs could serve as a decision support tool. For example, an actuary, claims adjuster, or a legal staff person could use LLMs to assess case strength before taking it further, helping to triage disputes, manage resources, or make settlement offers. One may also ask about the endogenous cost related to our research outcome: must the user draft a hefty legal brief in order to use LLMs and see predictions? We view this less as a cost and more as a feature. Most of these documents are going to exist already as a matter of the routine processing of claims. The LLM isn't looking for a fully realized and polished legal brief; in most cases it will work with factual summaries or incident reports that are quite routine in the early stages of the claims cycle. The LLM is thus a way to use these texts earlier in the cycle, before a formal legal review might be warranted.

The larger goal of our work, though, is to experiment with LLMs and show that they can be used in actuarially relevant applications, where at least some of the data is unstructured text. We chose workers' compensation disputes as a use case not because it is the only use case or the final one, but because it is complex enough, with rich data and significant legal shading, that modeling the outcomes traditionally has been difficult to do from a structured actuarial perspective. The results presented in the paper are promising in this regard and suggest that LLMs can outperform traditional NLP approaches at predicting outcomes, especially with context. This suggests that LLMs are useful for uncovering latent features of legal language that may not be transparent or intuitive even to experienced actuaries or underwriters. Further, they may have even broader applications in the actuarial and underwriting world, especially where there are processes involving claims man-

agement, risk assessment, or policy review with substantial amounts of textual narrative content.

ACKNOWLEDGMENTS

We gratefully acknowledge the assistance of the following Middle Tennessee State University students in determining the outcome of several workers' compensation cases: Jiyao Luo, Aocheng Wang, Danlei Zhu, Jennifer Rody, Clifford Jones, Lala Yamazaki, and Derek Nehring.

FINANCIAL DISCLOSURE

This research was supported by the Middle Tennessee State University Data Science Seed Grant.

DATA AND PYTHON CODE AVAILABILITY

Raw data are publicly available at the North Carolina Industrial Commission website: https://www.ic.nc.gov/database.html. Python codes are publicly available at https://github.com/cvajira/Workers_Compensation_Case_Studies. The anonymized dataset, which includes Anonymized Issues, Anonymized Findings of Fact, and Decisions, is available at: https://data.mendeley.com/datasets/b6n2vn2d69/1.

CONFLICT OF INTEREST

The authors declare that they have no financial or personal relationships that could inappropriately influence or be perceived to influence the work reported in this paper. No external commercial entities provided funding, materials, or in-kind support that could constitute a competing interest. All affiliations and sources of support are disclosed in the Acknowledgments and Funding sections.

AI STATEMENT

The authors acknowledge the use of ChatGPT and Gemini to produce Python codes used in this research. Python codes are publicly available.

Published: January 23, 2026 EDT.

REFERENCES

Abbasi, Ahmed, Hsinchun Chen, and Arab Salem. 2008. "Sentiment Analysis in Multiple Languages: Feature Selection for Opinion Classification in Web Forums." *ACM Transactions on Information Systems (TOIS)* 26 (3): 1–34. https://doi.org/10.1145/1361684.1361685.

Ahn, Michael, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, et al. 2022. "Do as i Can, Not as i Say: Grounding Language in Robotic Affordances." *arXiv Preprint arXiv:2204.01691*.

Balona, Caesar. 2024. "ActuaryGPT: Applications of Large Language Models to Insurance and Actuarial Work." *British Actuarial Journal* 29:e15. https://doi.org/10.1017/S1357321724000102.

Barrón-Cedeño, Alberto, Marta Vila, M Antònia Martí, and Paolo Rosso. 2013. "Plagiarism Meets Paraphrasing: Insights for the next Generation in Automatic Plagiarism Detection." *Computational Linguistics* 39 (4): 917–47. https://doi.org/10.1162/COLI_a_00153.

Bosley, Mitchell, Musashi Jacobs-Harukawa, Hauke Licht, and Alexander Hoyle. 2023. "Do We Still Need BERT in the Age of GPT? Comparing the Benefits of Domain-Adaptation and in-Context-Learning Approaches to Using LLMs for Political Science Research."." In *2023 Annual Meeting of the Midwest Political Science Association (MPSA)*. https://mbosley.github.io/papers/bosley_harukawa_licht_hoyle_mpsa2023.pdf.

Butler, Richard J., and John D. Worrall. 1983. "Workers' Compensation: Benefit and Injury Claims Rates in the Seventies." *The Review of Economics and Statistics* 65 (4): 580–89. https://doi.org/10.2307/1935926.

California Department of Industrial Relations, Division of Workers' Compensation. 2023. "How Is My Case Resolved?" 2023. https://www.dir.ca.gov/dwc/CaseResolved.htm.

Campesato, Oswald. 2022. *Natural Language Processing Using r Pocket Primer*. Berlin, Boston: Mercury Learning and Information. https://doi.org/10.1515/9781683927297.

Cao, Chunling, Xiuling Yao, Wenying Gong, Yutong Li, Yang Pan, Jinghan Zhang, Yuechen Yang, and Chaoran Zhao. 2024. "LLMs for Insurance: Opportunities, Challenges and Concerns." https://www.preprints.org/frontend/manuscript/84aa2603cc247b348fd60874e037ac25/download_pub.

Cao, Yuan, Ze Chen, and Zhiyu Quan. 2024. "Assessing Insurer's Litigation Risk: Claim Dispute Prediction with Actionable Interpretations Using Machine Learning Technique." *Available at SSRN 5085185*. https://doi.org/10.2139/ssrn.5085185.

Chang, Yupeng, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. "A Survey on Evaluation of Large Language Models." *ACM Transactions on Intelligent Systems and Technology* 15 (3): 1–45. https://doi.org/10.1145/3641289.

Cheng, Sitao, Liangming Pan, Xunjian Yin, Xinyi Wang, and William Yang Wang. 2024. "Understanding the Interplay between Parametric and Contextual Knowledge for Large Language Models." *arXiv Preprint arXiv:2410.08414*. https://doi.org/10.48550/arXiv.2410.08414.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding." In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, edited by Jill Burstein, Christy Doran, and Thamar Solorio, 4171–86. Minneapolis, Minnesota: Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423.

Ferrer, Xavier, Tom Van Nuenen, Jose M. Such, Mark Coté, and Natalia Criado. 2021. "Bias and Discrimination in AI: A Cross-Disciplinary Perspective." *IEEE Technology and Society Magazine* 40 (2): 72–80. https://doi.org/10.1109/MTS.2021.3056293.

Geary, Kelly A. 2024. "Analyzing Who Is the Prevailing Party?" Adams Leclair LLP. September 3, 2024. https://adamsleclair.law/published-articles/analyzing-who-is-the-prevailing-party/.

Guyton, Gregory P. 1999. "A Brief History of Workers' Compensation." *The Iowa Orthopaedic Journal* 19:106. https://pmc.ncbi.nlm.nih.gov/articles/PMC1888620/pdf/IowaOrthopJ-19-106.pdf.

GV, Ananth Raj, Qian You, Dan Dickinson, Eric Bunch, and Glenn Fung. 2021. "Document Classification and Information Extraction Framework for Insurance Applications." In *2021 Third International Conference on Transdisciplinary AI (TransAI)*, 8–16. https://doi.org/10.1109/TransAI51903.2021.00010.

Hadi, Muhammad Usman, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. 2023. "A Survey on Large Language Models: Applications, Challenges, Limitations, and Practical Usage." *Authorea Preprints* 3. https://doi.org/10.36227/techrxiv.23589741.v3.

Huttula, Arttu. 2025. "Advanced Prompt Engineering: Systematic Approaches to Enhance LLM Performance." *Master's Thesis*. https://urn.fi/URN:NBN:fi:amk-2025061823320.

Jeoung, Sullam, Yueyan Chen, Yi Zhang, Shuai Wang, Haibo Ding, and Lin Lee Cheong. 2025. "PromptPrism: A Linguistically-Inspired Taxonomy for Prompts." *arXiv Preprint arXiv:2505.12592*. https://doi.org/10.48550/arXiv.2505.12592.

Kadhim, Ammar Ismael. 2018. "An Evaluation of Preprocessing Techniques for Text Classification." *International Journal of Computer Science and Information Security (IJCSIS)* 16 (6): 22–32. https://www.academia.edu/36998792/An_Evaluation_of_Preprocessing_Techniques_for_Text_Classification.

Kathuria, Abhinav, Anu Gupta, and R. K. Singla. 2021. "A Review of Tools and Techniques for Preprocessing of Textual Data." *Computational Methods and Data Engineering: Proceedings of ICMDE 2020, Volume 1*, 407–22. https://doi.org/10.1007/978-981-15-6876-3_31.

Kim, Alex, Maximilian Muhn, and Valeri Nikolaev. 2024. "Financial Statement Analysis with Large Language Models." *arXiv Preprint arXiv:2407.17866*. https://doi.org/10.48550/arXiv.2407.17866.

Kindbom, Hannes. 2019. "LSTM vs Random Forest for Binary Classification of Insurance Related Text." *Industrial Engineering and Management Thesis, KTH Royal Institute of Technology*. https://www.diva-portal.org/smash/get/diva2:1334069/FULLTEXT01.pdf.

Kojima, Takeshi, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. "Large Language Models Are Zero-Shot Reasoners." In *Advances in Neural Information Processing Systems*, edited by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, 35:22199–213. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2022/file/8bb0d291acd4acf06ef112099c16f326-Paper-Conference.pdf.

Kowsari, Kamran, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. "Text Classification Algorithms: A Survey." *Information* 10 (4): 150. https://doi.org/10.3390/info10040150.

Li, Dongchen, Zhuo Jin, Linyi Qian, and Hailiang Yang. 2025. "Textual Analysis of Insurance Claims with Large Language Models." *Journal of Risk and Insurance* 92 (2): 505–35. https://doi.org/10.1111/jori.70004.

Li, Qian, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. 2022. "A Survey on Text Classification: From Traditional to Deep Learning." *ACM Trans. Intell. Syst. Technol.* 13 (2). https://doi.org/10.1145/3495162.

Lin, Chenwei, Hanjia Lyu, Jiebo Luo, and Xian Xu. 2024. "Harnessing Gpt-4v (Ision) for Insurance: A Preliminary Exploration." *arXiv Preprint arXiv:2404.09690*.

Lucas, Devin L., Jennifer R. Lee, Kyle M. Moller, Mary B. O'Connor, Laura N. Syron, and Joanna R. Watson. 2020. "Using Workers' Compensation Claims Data to Describe Nonfatal Injuries among Workers in Alaska." *Safety and Health at Work* 11 (2): 165–72. https://doi.org/10.1016/j.shaw.2020.01.004.

Ly, Antoine, Benno Uthayasooriyar, and Tingting Wang. 2020. "A Survey on Natural Language Processing (Nlp) and Applications in Insurance." *arXiv Preprint arXiv:2010.00462*.

Mathews, David. 2016. "Data Mining and Machine Learning Algorithms for Workers' Compensation Early Severity Prediction." PhD thesis, Middle Tennessee State University. http://jewlscholar.mtsu.edu/handle/mtsu/5015.

Meyers, Alysha R., Ibraheem S. Al-Tarawneh, Steven J. Wurzelbacher, P. Timothy Bushnell, Michael P. Lampl, Jennifer L. Bell, Stephen J. Bertke, David C. Robins, Chih-Yu Tseng, Chia Wei, et al. 2018. "Applying Machine Learning to Workers' Compensation Data to Identify Industry-Specific Ergonomic and Safety Prevention Priorities: Ohio, 2001 to 2011." *Journal of Occupational and Environmental Medicine* 60 (1): 55–73. https://doi.org/10.1097/JOM.0000000000001162.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. "Distributed Representations of Words and Phrases and Their Compositionality." In *Advances in Neural Information Processing Systems*, edited by C. J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger. Vol. 26. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf.

Moniz, Francisco Fernandes Correia do Canto. 2019. "Healthcare Provider Effciency in Workers' Compensation: An Approach with Machine Learning." PhD thesis, Instituto Superior de Economia e Gestão. https://www.proquest.com/dissertations-theses/healthcare-provider-efficiency-workers/docview/2585931188/se-2.

New York City Bar Association. 2025. "Workers' Compensation Process." 2025. https://www.nycbar.org/get-legal-help/article/workers-comp/workers-compensation-process%E2%80%AF/.

Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. "GloVe: Global Vectors for Word Representation." In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, edited by Alessandro Moschitti, Bo Pang, and Walter Daelemans, 1532–43. Doha, Qatar: Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1162.

PyTorch Documentation. 2024. "Randomness in PyTorch." 2024. https://pytorch.org/docs/stable/notes/randomness.html.

Santu, Shubhra Kanti Karmaker, and Dongji Feng. 2023. "Teler: A General Taxonomy of Llm Prompts for Benchmarking Complex Tasks." *arXiv Preprint arXiv:2305.11430*.

Sarkar, Suproteem K., and Keyon Vafa. 2024. "Lookahead Bias in Pretrained Language Models." *Available at SSRN*.

Schulhoff, Sander, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, et al. 2024. "The Prompt Report: A Systematic Survey of Prompt Engineering Techniques." *arXiv Preprint arXiv:2406.06608*. https://doi.org/10.48550/arXiv.2406.06608.

Sezerer, Erhan, and Selma Tekir. 2021. "A Survey on Neural Word Embeddings." *arXiv Preprint arXiv:2110.01804*.

Staudinger, Moritz, Wojciech Kusa, Florina Piroi, Aldo Lipani, and Allan Hanbury. 2024. "A Reproducibility and Generalizability Study of Large Language Models for Query Generation." In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, 186–96. https://doi.org/10.1145/3673791.3698432.

Tao, Yufei, Adam Hiatt, Rahul Seetharaman, and Ameeta Agrawal. 2025. "'lost-in-the-Later': Framework for Quantifying Contextual Grounding in Large Language Models." *arXiv Preprint arXiv:2507.05424*.

Texas Department of Insurance, Division of Workers' Compensation. 2025. "Dispute Resolution for Injured Employees." 2025. https://www.tdi.texas.gov/wc/employee/dispute.html.

Vinit Patwa, Pooja. 2024. "Unveiling Patterns in Employee Compensation: A Feature-Driven Analysis Using Machine Learning Algorithms." PhD thesis, Dublin Business School. https://hdl.handle.net/10788/4569.

Wang, Julian Junyan, and Victor Xiaoqi Wang. 2025. "Assessing Consistency and Reproducibility in the Outputs of Large Language Models: Evidence across Diverse Finance and Accounting Tasks." *arXiv Preprint arXiv:2503.16974*.

Wang, Yibo, and Wei Xu. 2018. "Leveraging Deep Learning with LDA-Based Text Analytics to Detect Automobile Insurance Fraud." *Decision Support Systems* 105:87–95. https://doi.org/10.1016/j.dss.2017.11.001.

Webson, Albert, and Ellie Pavlick. 2022. "Do Prompt-Based Models Really Understand the Meaning of Their Prompts?" In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2300–2344. https://doi.org/10.18653/v1/2022.naacl-main.167.

Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models." In *Advances in Neural Information Processing Systems*, edited by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, 35:24824–37. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

Workers Compensation Insurance. 2016. "Workers Compensation Insurance." *Ruhl Insurance*. https://www.iruhl.com/business-insurance/workers-comp.

Yamin, Samuel C., Anca Bejan, David L. Parker, Min Xi, and Lisa M. Brosseau. 2016. "Analysis of Workers' Compensation Claims Data for Machine-Related Injuries in Metal Fabrication Businesses." *American Journal of Industrial Medicine* 59 (8): 656–64. https://doi.org/10.1002/ajim.22603.

Zhao, Wayne Xin, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, et al. 2025. "A Survey of Large Language Models." https://doi.org/10.48550/arXiv.2303.18223.

APPENDICES

APPENDIX A. RESULTS

Table A.1. Anonymization of Findings of Fact.

Before Anonymization	After Anonymization
Dr. Moore's June 2014 evaluation indicated that the nodules on Plaintiff's hands were unrelated to the December 2010 incident and that Plaintiff had advised that he was treating privately for this condition with a hand specialist. Dr. Moore confirmed these two facts during his deposition. Dr. Moore's June 2014 evaluation and his testimony are identical on this issue.	Dr. [PERSON]'s [DATE] evaluation indicated that the nodules on Plaintiff's hands were unrelated to the [DATE] incident and that [PERSON] had advised that he was treating privately for this condition with a hand specialist. Dr. [PERSON] confirmed these [CARDINAL] facts during his deposition. Dr. [PERSON]'s [DATE] evaluation and his testimony are identical on this issue.

Table A.2. Panel A: Model characteristics.

Model Name	Context Window	Max Output Tokens	Knowledge Cutoff
`gpt-3.5-turbo-0125`	16,385 tokens	4,096 tokens	Sep 2021
`gpt-4o-mini`	128,000 tokens	16,384 tokens	Oct 2023
`o4-mini`	128,000 tokens	16,384 tokens	Oct 2023
`gemini-1.5-pro`	1 million tokens	8,192 tokens	Aug 2024
`gemini-1.5-flash-002`	1 million tokens	8,192 tokens	Aug 2024
`gemini-2.0-flash`	1 million tokens	8,192 tokens	Aug 2024
`deepseek-chat`	64,000 tokens	8,000 tokens	July 2024
`claude-3-haiku-20240307`	200,000 tokens	4,096 tokens	August 2023

Table A.3. Performance metrics for traditional NLP models using Preprocessed_Issues ($N = 1221$, threshold = 0.5).

Embedding	Classifier	Accuracy	Recall	Specificity	Precision	F1	AUC
TF-IDF							
TF-IDF	Random Forest	**0.6929**	0.8821	0.3675	0.7057	0.7841	**0.7281**
TF-IDF	Gradient Boosting	0.6323	**1.0000**	0.0000	0.6323	0.7747	0.6560
TF-IDF	XGB	0.6536	0.9974	0.0624	0.6465	**0.7845**	0.7052
TF-IDF	XGB (oversample)	0.6454	0.7150	0.5256	0.7216	0.7183	0.6887
TF-IDF	XGB (undersample)	0.6691	0.7306	**0.5635**	0.7421	0.7363	0.7057
Word2Vec (20 runs mean ± std)							
Word2Vec	Random Forest	0.6794 ± 0.0065	0.8885 ± 0.0093	0.3197 ± 0.0154	0.6919 ± 0.0045	0.7780 ± 0.0048	0.6985 ± 0.0075
Word2Vec	Gradient Boosting	0.6465 ± 0.0140	0.9905 ± 0.0226	0.0551 ± 0.0735	0.6437 ± 0.0144	0.7800 ± 0.0042	0.6667 ± 0.0129
Word2Vec	XGB	0.6369 ± 0.0049	0.9993 ± 0.0020	0.0138 ± 0.0157	0.6354 ± 0.0034	0.7768 ± 0.0021	0.6809 ± 0.0108
Word2Vec	XGB (oversample)	0.6437 ± 0.0106	0.9962 ± 0.0075	0.0378 ± 0.0409	0.6404 ± 0.0085	0.7796 ± 0.0041	0.6869 ± 0.0095
Word2Vec	XGB (undersample)	0.6474 ± 0.0130	0.9916 ± 0.0161	0.0556 ± 0.0606	0.6439 ± 0.0119	0.7806 ± 0.0043	0.6800 ± 0.0098
BERT (20 runs mean ± std)							
BERT	Sequence Classification	0.6873 ± 0.0127	0.7699 ± 0.0489	0.5452 ± 0.0560	**0.7451** ± 0.0127	0.7562 ± 0.0187	0.6988 ± 0.0115
Overall Summary							
Mean ± std		0.6577 ±0.0210	0.9056±0.1163	0.2315 ±0.2346	0.6771 ±0.0448	0.7681 ± 0.0219	0.6905 ± 0.0199

Table A.4. Performance metrics for traditional NLP models using `Preprocessed_Facts` ($N = 2845$, threshold $= 0.5$).

Embedding	Classifier	Accuracy	Recall	Specificity	Precision	F1	AUC
TF-IDF							
TF-IDF	Random Forest	0.7641	0.9450	0.4356	0.7526	**0.8379**	**0.8330**
TF-IDF	Gradient Boost	0.6450	**1.0000**	0.0000	0.6450	0.7842	0.7906
TF-IDF	XGB	0.6840	0.9918	0.1248	0.6731	0.8019	0.7834
TF-IDF	XGB (oversample)	0.7062	0.7548	0.6178	0.7820	0.7682	0.7670
TF-IDF	XGB (undersample)	0.7202	0.7428	**0.6792**	**0.8079**	0.7740	0.7825
Word2Vec (20 runs, mean ± std)							
Word2Vec	Random Forest	**0.7649** ± 0.0043	0.9238 ± 0.0046	0.4762 ± 0.0111	0.7622 ± 0.0037	0.8352 ± 0.0029	0.8320 ± 0.0031
Word2Vec	Gradient Boosting	0.6887 ± 0.0274	0.9864 ± 0.0196	0.1477 ± 0.1107	0.6791 ± 0.0265	0.8038 ± 0.0115	0.7749 ± 0.0177
Word2Vec	XGB	0.6634 ± 0.0163	0.9987 ± 0.0026	0.0543 ± 0.0499	0.6576 ± 0.0116	0.7929 ± 0.0077	0.8006 ± 0.0068
Word2Vec	XGB (oversample)	0.6836 ± 0.0282	0.9927 ± 0.0109	0.1220 ± 0.0982	0.6736 ± 0.0236	0.8022 ± 0.0129	0.8063 ± 0.0110
Word2Vec	XGB (undersample)	0.6920 ± 0.0288	0.9903 ± 0.0125	0.1499 ± 0.1025	0.6802 ± 0.0250	0.8061 ± 0.0132	0.8036 ± 0.0104
BERT (20 runs, mean ± std)							
BERT	Sequence Classification	0.7158 ± 0.0149	0.7956 ± 0.0520	0.5707 ± 0.0574	0.7721 ± 0.0137	0.7823 ± 0.0207	0.7496 ± 0.0151
Overall Summary							
Mean ± std		0.7025 ±0.0375	0.9202±0.1035	0.3071 ±0.2501	0.7169 ±0.0584	0.7990 ± 0.0224	0.7930 ± 0.0256

Table A.5. Performance metrics of LLM models using `Anonymized_Issues` ($N = 6103$) as predictor with the simple standard prompt.

Model Name	Accuracy	Recall	Specificity	Precision	F1 Score	AUC Score
deepseek-chat	0.4673	0.5452	0.8391	0.5616	0.4551	0.5452
claude-3-haiku-20240307	0.5431	0.5330	0.4946	0.5309	0.5279	0.5330
gemini-1.5-pro	**0.5552**	**0.5851**	0.6976	0.5817	**0.5544**	**0.5851**
gemini-1.5-flash-002	0.5373	0.5805	0.7436	0.5814	0.5372	0.5805
gemini-2.0-flash	0.3844	0.5100	**0.9839**	0.5833	0.3046	0.51
gpt-3.5-turbo-0125	0.5238	0.5135	0.4744	0.5126	0.5088	0.5135
gpt-4.1-mini	0.5415	0.5841	0.7445	0.5846	0.5415	0.5841
o4-mini	0.5279	**0.5851**	0.8018	**0.5933**	0.5261	**0.5851**
Mean	0.5101	0.5545	0.7224	0.5662	0.4945	0.5545
Std	0.0573	0.0330	0.1704	0.0292	0.0825	0.0330

Table A.6. Performance metrics of LLM models using `Anonymized_Facts` ($N = 14,225$) as predictor with the simple standard prompt.

Model Name	Accuracy	Recall	Specificity	Precision	F1 Score	AUC Score
deepseek-chat	**0.8712**	**0.8800**	0.9105	**0.8572**	**0.8644**	**0.8800**
claude-3-haiku-20240307	0.7557	0.7986	**0.9465**	0.7777	0.754	0.7986
gemini-1.5-pro	0.8625	0.8798	0.9396	0.8515	0.857	0.8798
gemini-1.5-flash-002	0.8470	0.8654	0.9289	0.8371	0.8414	0.8654
gemini-2.0-flash	0.8344	0.8620	0.9570	0.8317	0.8304	0.8620
gpt-3.5-turbo-0125	0.7890	0.8007	0.8408	0.7779	0.7810	0.8007
gpt-4.1-mini	0.8699	0.8766	0.8994	0.8555	0.8626	0.8766
o4-mini	0.8647	0.8744	0.9075	0.8509	0.8578	0.8744
Mean	0.8368	0.8547	0.9163	0.8299	0.8311	0.8547
Std	0.0426	0.0346	0.0366	0.0334	0.0415	0.0346

Table A.7. Average performance metrics of LLM models from a fixed 20% random sample of `Anonymized_Issues` ($N = 1220$), evaluated over 20 runs using the simple standard prompt.

Model Name	Accuracy	Recall	Specificity	Precision	F1 Score	AUC Score
gemini-1.5-flash-002: Mean	0.5393	0.5840	0.7414	0.5832	0.5393	0.5840
gemini-1.5-flash-002: Std	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
gpt-4.1-mini: Mean	0.5555	0.5979	0.7475	0.5958	0.5553	0.5979
gpt-4.1-mini: Std	0.0063	0.0069	0.0130	0.0069	0.0063	0.0069

Table A.8. Average performance metrics of LLM models from a fixed 20% random sample of `Anonymized_Facts` ($N = 2845$), evaluated over 20 runs using the simple standard prompt.

Model Name	Accuracy	Recall	Specificity	Precision	F1 Score	AUC Score
gemini-1.5-flash-002: Mean	0.8488	0.8695	0.9373	0.8379	0.8427	0.8695
gemini-1.5-flash-002: Std	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
gpt-4.1-mini: Mean	0.8732	0.8806	0.9047	0.8573	0.8652	0.8806
gpt-4.1-mini: Std	0.0024	0.0023	0.0036	0.0024	0.0024	0.0023

Table A.9. Performance metrics for traditional NLP models using `Preprocessed_Issues` ($N = 1221$) at the F1-maximizing threshold.

Embedding	Classifier	Accuracy	Recall	Specificity	Precision	F1	AUC
TF-IDF							
TF-IDF	Random Forest	0.6871	0.8355	0.4321	0.7167	0.7715	**0.7281**
TF-IDF	Gradient Boost	0.6618	0.9158	0.2249	0.6701	0.7739	0.6560
TF-IDF	XGB	0.6863	0.9003	0.3185	0.6943	**0.7840**	0.7047
TF-IDF	XGB (oversample)	0.6806	0.8782	0.3408	0.6961	0.7766	0.6887
TF-IDF	XGB (undersample)	0.6724	0.7681	0.5078	**0.7285**	0.7478	0.7057
Word2Vec (20 runs, mean ± std)							
Word2Vec	Random Forest	0.6747 ± 0.0076	0.8118 ± 0.0191	0.4389 ± 0.0276	0.7134 ± 0.0069	0.7593 ± 0.0075	0.6985 ± 0.0075
Word2Vec	Gradient Boosting	0.6699 ± 0.0101	**0.9271** ± 0.0239	0.2277 ± 0.0634	0.6742 ± 0.0130	0.7803 ± 0.0038	0.6667 ± 0.0129
Word2Vec	XGB	0.6758 ± 0.0084	0.8518 ± 0.0176	0.3732 ± 0.0317	0.7004 ± 0.0080	0.7686 ± 0.0067	0.6809 ± 0.0108
Word2Vec	XGB (oversample)	0.6586 ± 0.0117	0.7632 ± 0.0366	0.4788 ± 0.0483	0.7162 ± 0.0116	0.7384 ± 0.0149	0.6869 ± 0.0095
Word2Vec	XGB (undersample)	0.6529 ± 0.0090	0.7315 ± 0.0387	**0.5177** ± 0.0493	0.7235 ± 0.0114	0.7267 ± 0.0153	0.6800 ± 0.0098
BERT (20 runs, mean ± std)							
BERT	Sequence Classification	**0.6913** ± 0.0109	0.8341 ± 0.0399	0.4456 ± 0.0585	0.7219 ± 0.0136	0.7733 ± 0.0127	0.6988 ± 0.0115
Overall Summary							
Mean ± std		0.6738 ±0.0123	0.8379±0.0650	0.3915 ±0.1033	0.7050 ±0.0197	0.7637 ± 0.0185	0.6905 ± 0.0199

Table A.10. Performance metrics for traditional NLP models using `Preprocessed_Facts` $(N = 2845)$ at the F1-maximizing threshold.

Embedding	Classifier	Accuracy	Recall	Specificity	Precision	F1	AUC
TF-IDF							
TF-IDF	Random Forest	0.7645	0.9084	0.5030	0.7686	**0.8327**	**0.8330**
TF-IDF	Gradient Boost	0.7501	0.8703	0.5317	0.7715	0.8179	0.7906
TF-IDF	XGB	0.7378	0.8850	0.4703	0.7522	0.8132	0.7844
TF-IDF	XGB (oversample)	0.7114	0.7902	0.5683	0.7688	0.7794	0.7670
TF-IDF	XGB (undersample)	0.7258	0.7722	**0.6416**	**0.7965**	0.7842	0.7825
Word2Vec (20 runs, mean ± std)							
Word2Vec	Random Forest	**0.7706** ± 0.0035	0.8776 ± 0.0153	0.5763 ± 0.0268	0.7903 ± 0.0079	0.8315 ± 0.0037	0.8320 ± 0.0031
Word2Vec	Gradient Boosting	0.7350 ± 0.0131	**0.9190** ± 0.0165	0.4006 ± 0.0581	0.7364 ± 0.0160	0.8174 ± 0.0066	0.7749 ± 0.0177
Word2Vec	XGB	0.7534 ± 0.0060	0.8948 ± 0.0184	0.4964 ± 0.0368	0.7638 ± 0.0102	0.8239 ± 0.0047	0.8006 ± 0.0068
Word2Vec	XGB (oversample)	0.7517 ± 0.0090	0.8299 ± 0.0213	0.6096 ± 0.0401	0.7948 ± 0.0141	0.8116 ± 0.0075	0.8063 ± 0.0110
Word2Vec	XGB (undersample)	0.7473 ± 0.0077	0.8181 ± 0.0269	0.6187 ± 0.0409	0.7964 ± 0.0135	0.8066 ± 0.0087	0.8036 ± 0.0104
BERT (20 runs, mean ± std)							
BERT	Sequence Classification	0.7240 ± 0.0104	0.8826 ± 0.0342	0.4359 ± 0.0542	0.7403 ± 0.0126	0.8047 ± 0.0106	0.7496 ± 0.0151
Overall Summary							
Mean ± std		0.7429 ±0.0180	0.8589±0.0488	0.5320 ±0.0783	0.7709 ±0.0219	0.8112 ±0.0171	0.7931 ± 0.0256

Table A.11. Performance metrics of LLM models using `Anonymized_Issues` as predictor with the simple standard prompt, evaluated on the same test set ($N = 1221$) as traditional NLP techniques.

Model Name	Accuracy	Recall	Specificity	Precision	F1 Score	AUC Score
deepseek-chat	0.4521	0.5346	0.8463	0.5507	0.4357	0.5346
claude-3-haiku-20240307	0.5487	0.5411	0.5122	0.5385	0.5350	0.5411
gemini-1.5-pro	0.5373	0.5744	0.7149	0.5735	0.5372	0.5744
gemini-1.5-flash-002	0.5299	0.5695	0.7194	0.5695	0.5299	0.5695
gemini-2.0-flash	0.3841	0.5102	0.9866	0.5925	0.3028	0.5102
gpt-3.5-turbo-0125	0.5307	0.5250	0.5033	0.5233	0.5183	0.5250
gpt-4.1-mini	0.5463	0.5885	0.7483	0.5890	0.5463	0.5885
o4-mini	0.5226	0.5749	0.7718	0.5796	0.5217	0.5749
Mean	0.5065	0.5523	0.7254	0.5646	0.4908	0.5523
Std	0.0581	0.0282	0.1606	0.0247	0.0835	0.0282

Table A.12. Performance metrics of LLM models using `Anonymized_Facts` as predictor with a simple standard prompt, evaluated on the same test set ($N = 2845$) as traditional NLP techniques.

Model Name	Accuracy	Recall	Specificity	Precision	F1 Score	AUC Score
deepseek-chat	0.8703	0.8812	0.9188	0.8567	0.8638	0.8812
claude-3-haiku-20240307	0.7650	0.8082	0.9560	0.7858	0.7632	0.8082
gemini-1.5-pro	0.8639	0.8818	0.9435	0.8530	0.8586	0.8818
gemini-1.5-flash-002	0.8509	0.8689	0.9306	0.8405	0.8453	0.8689
gemini-2.0-flash	0.8287	0.8588	0.9623	0.8285	0.8250	0.8588
gpt-3.5-turbo-0125	0.8042	0.8186	0.8683	0.7940	0.7971	0.8186
gpt-4.1-mini	0.8798	0.8886	0.9188	0.8658	0.8732	0.8886
o4-mini	0.8657	0.8763	0.9129	0.8521	0.8591	0.8763
Mean	0.8411	0.8603	0.9264	0.8346	0.8357	0.8603
Std	0.0394	0.0304	0.0269	0.0298	0.0382	0.0304

Table A.13. PaLM-SayCan prompt families with corresponding prompts.

Prompt ID	Family	Prompt	Simple or CoT
$PF1\_1$	NL Single Primitive	Predict winner of this legal case. Respond ONLY with the number 0 if the plaintiff likely lost. Do NOT provide any explanation, commentary, or any text other than 1 or 0.	Simple
$PF1\_2$	NL Nouns	Review the following workers' compensation dispute scenario. Based on this information, determine the likely result for the claimant. Respond ONLY with the number 0 if the plaintiff likely lost. Do NOT provide any explanation, commentary, or any text other than 1 or 0.	Simple
$PF1\_3$	NL Verb	Adjudicate the provided workers' compensation dispute scenario. Based on the details, determine if the plaintiff prevailed or failed. Respond ONLY with the number 0 if the plaintiff likely lost. Do NOT provide any explanation, commentary, or any text other than 1 or 0.	Simple
$PF1\_4$	Embodiment	You will be given a block of text. First, confirm the text contains facts from a legal case. If it does, proceed to predict the plaintiff's outcome. If the plaintiff's position appears stronger, output 1. If it appears weaker, output 0. Your final response must be only the single digit.	Simple
$PF1\_5$	Crowd-Sourced	Hey, I've got this legal stuff here, a bunch of facts from a case. Could you take a look and just tell me who you think won? Just give me a 1 if you think the plaintiff got it, or a 0 if they probably lost. And please, just the number, nothing else, thanks!	Simple
$PF1\_6$	Long Horizon	You are a legal expert in workers' compensation claim disputes, reading the facts of a case. Internally, go step by step through each fact to determine whether the plaintiff likely won or lost, carefully considering each relevant detail and legal principle. Form your conclusion based solely on the facts presented. However, do not reveal your reasoning or any internal thoughts in your final answer. Respond ONLY with the number 1 if the plaintiff likely won. Respond ONLY with the number 0 if the plaintiff likely lost. Do NOT provide any explanation, commentary, or any text other than 1 or 0.	CoT
$PF1\_7$	Structured Language (**standard, simple prompt in this research**)	Analyze the following legal case facts. Based solely on these facts, predict whether the plaintiff likely won or lost the case. 1. Respond ONLY with the number 1 if the plaintiff likely won. 2. Respond ONLY with the number 0 if the plaintiff likely lost. Do NOT provide any explanation, commentary, or any text other than 1 or 0.	Simple (and standard in this research)

Table A.14. "Let's Think" prompt families with corresponding prompts.

Prompt ID	Family	Prompt	Simple or CoT
$PF2\_1$	Let's think step by step.	Let's think step by step. Predict the winner of this legal case. Respond ONLY with the number 1 if the plaintiff likely won. Respond ONLY with the number 0 if the plaintiff likely lost. Do NOT provide any explanation, commentary, or any text other than 1 or 0.	CoT
$PF2\_2$	Let's think about this logically.	Let's think about this logically. Predict the winner of this legal case. Respond ONLY with the number 1 if the plaintiff likely won. Respond ONLY with the number 0 if the plaintiff likely lost. Do NOT provide any explanation, commentary, or any text other than 1 or 0.	CoT
$PF2\_3$	Let's solve this problem by splitting it into steps.	Let's solve this problem by splitting it into steps. Predict the winner of this legal case. Respond ONLY with the number 1 if the plaintiff likely won. Respond ONLY with the number 0 if the plaintiff likely lost. Do NOT provide any explanation, commentary, or any text other than 1 or 0.	CoT
$PF2\_4$	Let's be realistic and think step by step.	Let's be realistic and think step by step. Respond ONLY with the number 1 if the plaintiff likely won. Respond ONLY with the number 0 if the plaintiff likely lost. Do NOT provide any explanation, commentary, or any text other than 1 or 0.	CoT
$PF2\_5$	Let's think like a prosecutor step by step.	Let's think like a prosecutor step by step. Respond ONLY with the number 1 if the plaintiff likely won. Respond ONLY with the number 0 if the plaintiff likely lost. Do NOT provide any explanation, commentary, or any text other than 1 or 0.	CoT

Table A.15. Premise-hypothesis prompt families (instructive, misleading-moderate, misleading-extreme, irrelevant, null approach).

Prompt ID	Family	Prompt	Simple or CoT
		Premise (P): The legal case facts as presented in the input. **Hypothesis (H):** The plaintiff won the case.	
$PF3\_1$	Instructive	The legal case facts as presented in the input. Is it correct to conclude that the plaintiff won? Respond ONLY with the number 1 if the plaintiff likely won. Respond ONLY with the number 0 if the plaintiff likely lost. Do NOT provide any explanation, commentary, or any text other than 1 or 0.	Simple
$PF3\_2$	Misleading-moderate	The legal case facts as presented in the input. Do most of the above words appear in the following passage? The plaintiff won the case. Respond ONLY with the number 1 if the plaintiff likely won. Respond ONLY with the number 0 if the plaintiff likely lost. Do NOT provide any explanation, commentary, or any text other than 1 or 0.	Simple
$PF3\_3$	Misleading-extreme	The legal case facts as presented in the input. Is the sentiment positive? The plaintiff won the case. Respond ONLY with the number 1 if the plaintiff likely won. Respond ONLY with the number 0 if the plaintiff likely lost. Do NOT provide any explanation, commentary, or any text other than 1 or 0.	Simple
$PF3\_4$	Irrelevant	The legal case facts as presented in the input. Inflections are annoying and thank god that Middle English got rid of most of them. The plaintiff won the case. Respond ONLY with the number 1 if the plaintiff likely won. Respond ONLY with the number 0 if the plaintiff likely lost. Do NOT provide any explanation, commentary, or any text other than 1 or 0.	Simple
$PF3\_5$	Null	The legal case facts as presented in the input. The plaintiff won the case. Respond ONLY with the number 1 if the plaintiff likely won. Respond ONLY with the number 0 if the plaintiff likely lost. Do NOT provide any explanation, commentary, or any text other than 1 or 0.	Simple

Table A.16. Performance metrics of `gemini-1.5-flash-002` across prompt families (PF1, PF2, PF3) using `Anonymized_Issues` as predictor, evaluated on the same test set ($N = 1221$) as traditional NLP techniques.

Prompt ID	Accuracy	Recall	Specificity	Precision	F1	AUC
PF1_1	0.3784	0.5070	0.9933	0.6066	0.2904	0.5070
PF1_2	0.3825	0.5103	0.9933	0.6238	0.2973	0.5103
PF1_3	0.3800	0.5088	0.9955	0.6333	0.2922	0.5088
PF1_4	0.4767	0.5484	0.8196	0.5614	0.4682	0.5484
PF1_5	0.4070	0.5227	0.9599	0.5824	0.3489	0.5227
PF1_6	0.4201	0.5303	0.9465	0.5845	0.3723	0.5303
PF1_7	0.5299	0.5695	0.7194	0.5695	0.5299	0.5695
PF2_1	0.4513	0.5502	0.9243	0.5974	0.4210	0.5502
PF2_2	0.4521	0.5509	0.9243	0.5982	0.4221	0.5509
PF2_3	0.4251	0.5328	0.9399	0.5835	0.3812	0.5328
PF2_4	0.4554	0.5479	0.8976	0.5816	0.4315	0.5479
PF2_5	0.4893	0.5690	0.8708	0.5946	0.4773	0.5690
PF3_1	0.4177	0.5246	0.9287	0.5622	0.3736	0.5246
PF3_2	0.6339	0.5139	0.0601	0.5791	0.4387	0.5139
PF3_3	0.6536	0.5639	0.2249	0.6204	0.5452	0.5639
PF3_4	0.6331	0.5044	0.0178	0.5838	0.4040	0.5044
PF3_5	0.6364	0.5098	0.0312	0.6228	0.4170	0.5098
Mean	0.4837	0.5332	0.7204	0.5932	0.4065	0.5332
Std	0.0974	0.0228	0.3726	0.0218	0.0751	0.0228

Table A.17. Performance metrics of `gpt-4.1-mini` across prompt families (PF1, PF2, PF3) using `Anonymized_Issues` as predictor, evaluated on the same test set $(N = 1221)$ as traditional NLP techniques.

Prompt ID	Accuracy	Recall	Specificity	Precision	F1	AUC
PF1_1	0.4988	0.5626	0.8040	0.5730	0.4944	0.5626
PF1_2	0.5651	0.5857	0.6637	0.5807	0.5625	0.5857
PF1_3	0.4554	0.5488	0.9020	0.5845	0.4307	0.5488
PF1_4	0.5938	0.5273	0.2762	0.5347	0.5206	0.5273
PF1_5	0.5143	0.5675	0.7684	0.5725	0.5131	0.5675
PF1_6	0.4685	0.5503	0.8597	0.5721	0.4538	0.5503
PF1_7	0.5463	0.5885	0.7483	0.5890	0.5463	0.5885
PF2_1	0.5635	0.5691	0.5902	0.5642	0.5560	0.5691
PF2_2	0.5676	0.5784	0.6192	0.5730	0.5621	0.5784
PF2_3	0.5790	0.5837	0.6013	0.5778	0.5710	0.5837
PF2_4	0.5209	0.5508	0.6637	0.5492	0.5204	0.5508
PF2_5	0.5574	0.5796	0.6637	0.5752	0.5553	0.5796
PF3_1	0.5733	0.5866	0.6370	0.5808	0.5686	0.5866
PF3_2	0.5872	0.5245	0.2873	0.5299	0.5193	0.5245
PF3_3	0.6331	0.5263	0.1225	0.5751	0.4797	0.5263
PF3_4	0.6339	0.5046	0.0156	0.6089	0.4024	0.5046
PF3_5	0.6364	0.5065	0.0156	0.7065	0.4034	0.5065
Mean	0.5585	0.5553	0.5434	0.5792	0.5094	0.5553
Std	0.0536	0.0284	0.2877	0.0379	0.0570	0.0284

Table A.18. Performance metrics of `deepseek-chat` across prompt families (PF1, PF2, PF3) using `Anonymized_Issues` as predictor, evaluated on the same test set ($N = 1221$) as traditional NLP techniques.

Prompt ID	Accuracy	Recall	Specificity	Precision	F1	AUC
PF1_1	0.3735	0.5031	0.9933	0.5692	0.2819	0.5031
PF1_2	0.3964	0.5147	0.9621	0.5643	0.3317	0.5147
PF1_3	0.4013	0.5144	0.9421	0.5477	0.3457	0.5144
PF1_4	0.5299	0.5253	0.5078	0.5236	0.5181	0.5253
PF1_5	0.4464	0.5282	0.8374	0.5412	0.4300	0.5282
PF1_6	0.3784	0.5070	0.9933	0.6066	0.2904	0.5070
PF1_7	0.4521	0.5346	0.8463	0.5507	0.4357	0.5346
PF2_1	0.3931	0.5103	0.9532	0.5426	0.3295	0.5103
PF2_2	0.4439	0.5374	0.8909	0.5659	0.4179	0.5374
PF2_3	0.4210	0.5174	0.8820	0.5336	0.3893	0.5174
PF2_4	0.4840	0.5361	0.7327	0.5390	0.4825	0.5361
PF2_5	0.5536	0.5226	0.4053	0.5224	0.5225	0.5226
PF3_1	0.3923	0.5064	0.9376	0.5227	0.3334	0.5064
PF3_2	0.4881	0.5365	0.7194	0.5386	0.4873	0.5365
PF3_3	0.5897	0.5274	0.2918	0.5333	0.5225	0.5274
PF3_4	0.5790	0.5185	0.2895	0.5220	0.5139	0.5185
PF3_5	0.6028	0.5270	0.2405	0.5383	0.5148	0.5270
Mean	0.4662	0.5216	0.7309	0.5448	0.4204	0.5216
Std	0.0784	0.0112	0.2722	0.0218	0.0877	0.0112

Table A.19. Performance metrics of `gemini-1.5-flash-002` across prompt families (PF1, PF2, PF3) using `Anonymized_Facts` as predictor, evaluated on the same test set ($N = 2845$) as traditional NLP techniques.

Prompt ID	Accuracy	Recall	Specificity	Precision	F1	AUC
PF1_1	0.7961	0.8319	0.9554	0.8049	0.7933	0.8319
PF1_2	0.8400	0.8611	0.9337	0.8321	0.8348	0.8611
PF1_3	0.8298	0.8556	0.9444	0.8258	0.8253	0.8556
PF1_4	0.8542	0.8680	0.9158	0.8419	0.8479	0.8680
PF1_5	0.8377	0.8608	0.9404	0.8312	0.8328	0.8608
PF1_6	0.8541	0.8673	0.9129	0.8415	0.8477	0.8673
PF1_7	0.8509	0.8689	0.9306	0.8405	0.8453	0.8689
PF2_1	0.8534	0.8719	0.9356	0.8433	0.8480	0.8719
PF2_2	0.8562	0.8739	0.9347	0.8456	0.8507	0.8739
PF2_3	0.8601	0.8769	0.9347	0.8489	0.8545	0.8769
PF2_4	0.8478	0.8675	0.9356	0.8386	0.8425	0.8675
PF2_5	0.8408	0.8617	0.9337	0.8327	0.8356	0.8617
PF3_1	0.8555	0.8682	0.9119	0.8427	0.8490	0.8682
PF3_2	0.8763	0.8731	0.8624	0.8623	0.8670	0.8731
PF3_3	0.8724	0.8722	0.8713	0.8578	0.8636	0.8722
PF3_4	0.8685	0.8360	0.7238	0.8734	0.8496	0.8360
PF3_5	0.8815	0.8617	0.7931	0.8766	0.8682	0.8617
Mean	0.8515	0.8633	0.9041	0.8435	0.8445	0.8633
Std	0.0199	0.0124	0.0612	0.0174	0.0176	0.0124

Table A.20. Performance metrics of `gpt-4.1-mini` across prompt families (PF1, PF2, PF3) using `Anonymized_Facts` as predictor, evaluated on the same test set ($N = 2845$) as traditional NLP tech-niques.

Prompt ID	Accuracy	Recall	Specificity	Precision	F1	AUC
PF1_1	0.8794	0.8883	0.9188	0.8655	0.8729	0.8883
PF1_2	0.8822	0.8658	0.8089	0.8748	0.8699	0.8658
PF1_3	0.8794	0.8787	0.8762	0.8653	0.8709	0.8787
PF1_4	0.8731	0.8787	0.8980	0.8586	0.8657	0.8787
PF1_5	0.8749	0.8870	0.9287	0.8616	0.8688	0.8870
PF1_6	0.8805	0.8824	0.8891	0.8662	0.8726	0.8824
PF1_7	0.8798	0.8886	0.9188	0.8658	0.8732	0.8886
PF2_1	0.8773	0.8853	0.9129	0.8632	0.8705	0.8853
PF2_2	0.8721	0.8810	0.9119	0.8580	0.8653	0.8810
PF2_3	0.8714	0.8778	0.9000	0.8569	0.8640	0.8778
PF2_4	0.8837	0.8889	0.9069	0.8694	0.8766	0.8889
PF2_5	0.8815	0.8826	0.8861	0.8674	0.8735	0.8826
PF3_1	0.8752	0.8808	0.9000	0.8608	0.8678	0.8808
PF3_2	0.8749	0.8649	0.8307	0.8627	0.8638	0.8649
PF3_3	0.8770	0.8586	0.7950	0.8699	0.8637	0.8586
PF3_4	0.8397	0.7865	0.6030	0.8667	0.8070	0.7865
PF3_5	0.8369	0.7810	0.5881	0.8682	0.8021	0.7810
Mean	0.8729	0.8680	0.8514	0.8648	0.8617	0.8680
Std	0.0135	0.0329	0.1037	0.0047	0.0218	0.0329

Table A.21. Performance metrics of `deepseek-chat` across prompt families (PF1, PF2, PF3) using `Anonymized_Facts` as predictor, evaluated on the same test set ($N = 2845$) as traditional NLP techniques.

Index	Accuracy	Recall	Specificity	Precision	F1	AUC
PF1_1	0.7357	0.7900	0.9772	0.7774	0.7352	0.7900
PF1_2	0.7743	0.8184	0.9703	0.7958	0.7727	0.8184
PF1_3	0.8190	0.8512	0.9624	0.8218	0.8156	0.8512
PF1_4	0.8692	0.8466	0.7683	0.8639	0.8539	0.8466
PF1_5	0.8401	0.8651	0.9515	0.8349	0.8356	0.8651
PF1_6	0.8320	0.8606	0.9594	0.8303	0.8281	0.8606
PF1_7	0.8703	0.8812	0.9188	0.8567	0.8638	0.8812
PF2_1	0.8179	0.8511	0.9653	0.8217	0.8147	0.8511
PF2_2	0.8330	0.8612	0.9584	0.8309	0.8291	0.8612
PF2_3	0.8432	0.8671	0.9495	0.8370	0.8386	0.8671
PF2_4	0.8548	0.8734	0.9376	0.8447	0.8494	0.8734
PF2_5	0.8696	0.8809	0.9198	0.8561	0.8632	0.8809
PF3_1	0.8380	0.8633	0.9505	0.8331	0.8335	0.8633
PF3_2	0.8717	0.8832	0.9228	0.8583	0.8654	0.8832
PF3_3	0.8780	0.8807	0.8901	0.8636	0.8702	0.8807
PF3_4	0.8808	0.8818	0.8851	0.8666	0.8727	0.8818
PF3_5	0.8854	0.8782	0.8535	0.8734	0.8757	0.8782
Mean	0.8420	0.8608	0.9259	0.8392	0.8363	0.8608
Std	0.0397	0.0249	0.0529	0.0258	0.0372	0.0249

Table A.22. Performance metrics of LLM models using `Preprocessed_Issues` as predictor with the simple standard prompt, evaluated on the same test set ($N = 1221$) as traditional NLP techniques.

Model	Accuracy	Recall	Specificity	Precision	F1	AUC
deepseek-chat	0.3776	0.5064	0.9933	0.6020	0.2890	0.5064
claude-3-haiku-20240307	0.3866	0.5028	0.9421	0.5112	0.3231	0.5028
gemini-1.5-pro	0.3939	0.5091	0.9443	0.5337	0.3337	0.5091
gemini-1.5-flash-002	0.4414	0.5229	0.8307	0.5335	0.4249	0.5229
gemini-2.0-flash	0.4038	0.5094	0.9087	0.5236	0.3589	0.5094
gpt-3.5-turbo-0125	0.3849	0.5085	0.9755	0.5580	0.3085	0.5085
gpt-4.1-mini	0.5111	0.5421	0.6592	0.5410	0.5107	0.5421
o4-mini	0.5262	0.5385	0.5848	0.5360	0.5218	0.5385
Mean	0.4282	0.5174	0.8548	0.5424	0.3838	0.5174
Std	0.0593	0.0153	0.1531	0.0276	0.0913	0.0153

Table A.23. Performance metrics of LLM models using `Preprocessed_Facts` as predictor with the simple standard prompt, evaluated on the same test set ($N = 2485$) as traditional NLP techniques.

Model	Accuracy	Recall	Specificity	Precision	F1	AUC
deepseek-chat	0.6436	0.7112	0.9446	0.7195	0.6433	0.7112
claude-3-haiku-20240307	0.4000	0.5331	0.9921	0.6577	0.3387	0.5331
gemini-1.5-pro	0.7392	0.7722	0.8861	0.7506	0.7360	0.7722
gemini-1.5-flash-002	0.6910	0.7313	0.8703	0.7164	0.6894	0.7313
gemini-2.0-flash	0.4838	0.5913	0.9624	0.6593	0.4624	0.5913
gpt-3.5-turbo-0125	0.6225	0.6379	0.6911	0.6264	0.6158	0.6379
gpt-4.1-mini	0.7796	0.7297	0.5574	0.7725	0.7415	0.7297
o4-mini	0.7578	0.7604	0.7693	0.7428	0.7465	0.7604
Mean	0.6397	0.6834	0.8342	0.7056	0.6217	0.6834
Std	0.1353	0.0863	0.1506	0.0520	0.1481	0.0863

Table A.24. Interpretation of classification metrics used in this study.

Metric	Formula	Interpretation Under Workers' Compensation Context
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	Overall proportion of correctly predicted cases (both employee wins and losses). Can be misleading if most cases result in employee losses (class imbalance).
Precision	$\frac{TP}{TP+FP}$	Of all cases predicted as employee wins, how many actually resulted in employee winning? High precision means few false claims of employee success.
Recall	$\frac{TP}{TP+FN}$	Of all cases where the employee actually won, how many did the model correctly identify? High recall means the model rarely misses potential employee victories.
Specificity	$\frac{TN}{TN+FP}$	Of all cases where the employee actually lost, how many did the model correctly predict as losses? Important for insurers/employers to avoid predicting unnecessary payouts.
F1 Score	$\frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$	Harmonic mean of precision and recall. Useful when employee victories are rare and a balance between missing wins (FN) and falsely predicting wins (FP) is needed.
AUC	Measures the area under the Receiver Operating Characteristic (ROC) curve, which plots True Positive Rate (Recall) versus False Positive Rate (1 – Specificity).	Represents the model's ability to distinguish between employee wins (1) and employee losses (0). AUC = 0.5 means random guessing; AUC = 1.0 means perfect separation. Higher AUC indicates better overall discrimination across all possible decision thresholds.

APPENDIX B. DATASET EXPLANATIONS

The database was accessed using four full-text searchable databases via the "OpenText Content Server." To access the server, use "public" as both the username and password.

The search form used is "Public" with the following settings:

1. Look For: "All Words"
2. Modifier: "None"
3. Within: "All"
4. Search Text: "workers compensation"
5. Slices: "Deputy Commissioner [CS]"

Press "Search" to process the request and access the filtered data. Once the results are displayed, click on "Microsoft Word" under the "OTFileType" filter to narrow down the format. Finally, download the documents. The entire process was conducted between May 8, 2024, and May 10, 2024. Figure B.1 shows the input window, and Figure B.2 shows the output window of the North Carolina "OpenText Content Server."

Table B.1. Distribution of the dependent variable "Decision" according to the data frame.

Decision	Preprocessed_Facts			Preprocessed_Issues		
	Overall	Test	Train	Overall	Test	Train
0	0.3551	0.3550	0.3551	0.3675	0.3677	0.3675
1	0.6449	0.6450	0.6449	0.6325	0.6323	0.6325

Table B.2. Basic summary statistics for preprocessed Issues.

Statistic	Value
Number of documents	4,882
Total tokens	240,002
Vocabulary size	6,373
Words per document (avg | median)	49.16 | 37.00
Average word length	6.08
Type–Token Ratio	0.027
Herdan's C	0.707
Yule's K	118.3
Hapax legomena	3,068
Dis-legomena	834
Stopword ratio	2.80%
Total sentences	4,882
Average sentences per document	1.00
Average words per sentence	49.16
Characters (total)	1,694,504
Alphabetic / Digits / Spaces / Punctuation	1,459,196 / 0 / 235,120 / 188

Table B.3. Top 1-grams for preprocessed Issues.

Rank	1-grams	Frequency
1	plaintiff	15,249
2	compens	7,154
3	whether	6,128
4	entitl	5,651
5	date	5,323
6	cardin	4,778
7	person	4,638
8	follow	4,569
9	injuri	4,532
10	undersign	3,899

Table B.4. Top 2-grams for preprocessed Issues.

Rank	2-grams	Frequency
1	plaintiff entitl	3,893
2	whether plaintiff	3,811
3	undersign make	2,508
4	base upon	2,457
5	make follow	2,418
6	gpe gen	2,281
7	gen person	2,264
8	compens injuri	1,997
9	injuri accid	1,917
10	record undersign	1,858

Table B.5. Top 3-grams for preprocessed Issues.

Rank	3-grams	Frequency
1	undersign make follow	2,408
2	gpe gen person	2,258
3	record undersign make	1,649
4	gen person cardin	1,400
5	pursuant gpe gen	1,274
6	benefit plaintiff entitl	1,243
7	compens injuri accid	1,169
8	view entir record	1,130
9	base upon compet	1,130
10	evid view entir	1,118

Table B.6. Basic summary statistics for preprocessed Facts.

Statistic	Value
Number of documents	11,380
Total tokens	8,624,528
Vocabulary size	68,981
Words per document (avg | median)	757.87 | 625.00
Average word length	5.69
Type–Token Ratio	0.008
Herdan's C	0.698
Yule's K	90.8
Hapax legomena	36,858
Dis-legomena	8,952
Stopword ratio	3.59%
Total sentences	11,380
Average sentences per document	1.00
Average words per sentence	757.87
Characters (total)	57,659,113
Alphabetic / Digits / Spaces / Punctuation	49,039,176 / 3 / 8,613,148 / 6,786

Table B.7. Top 1-grams for preprocessed Facts.

Rank	1-grams	Frequency
1	plaintiff	503,068
2	person	359,917
3	date	331,866
4	dr	212,897
5	org	131,551
6	work	129,959
7	pain	81,961
8	cardin	78,613
9	time	67,248
10	injuri	66,731

Table B.8. Top 2-grams for preprocessed Facts.

Rank	2-grams	Frequency
1	dr person	205,544
2	date plaintiff	65,717
3	on date	40,778
4	mr person	34,897
5	date date	26,810
6	person testifi	24,760
7	date dr	22,777
8	m person	19,654
9	plaintiff return	16,661
10	return work	16,417

Table B.9. Top 3-grams for preprocessed Facts.

Rank	3-grams	Frequency
1	on date plaintiff	22,456
2	date dr person	22,287
3	dr person testifi	12,228
4	dr person date	11,650
5	dr person note	9,890
6	dr person recommend	7,855
7	person dr person	7,563
8	return dr person	7,376
9	on date dr	7,341
10	dr person opin	7,331

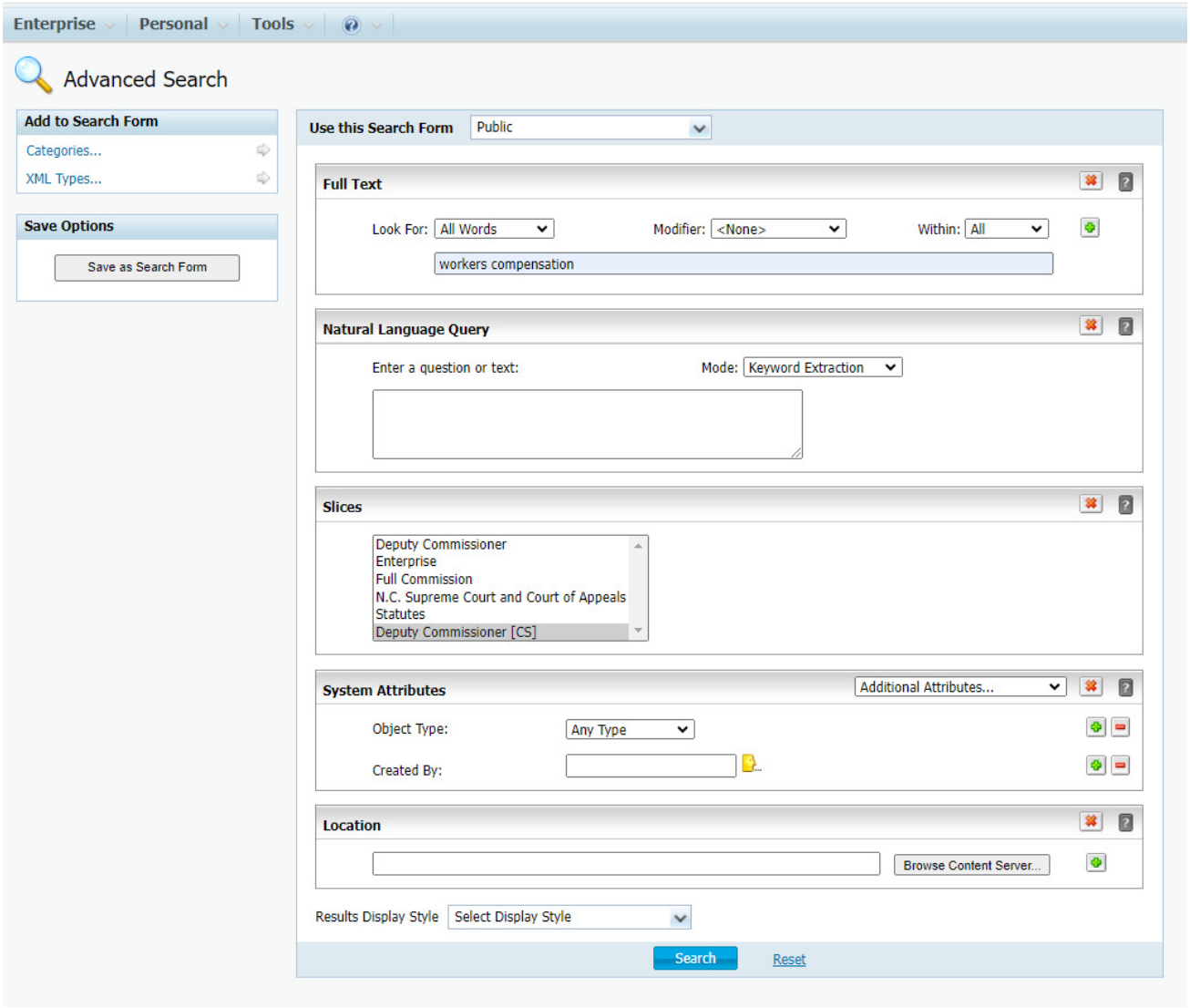

Figure B.1. Input search criteria for "workers compensation" using the "Public" search form and "Deputy Commissioner [CS]" slice.

Figure B.2. Displayed results for "workers compensation" search query filtered by the specified criteria.