

Using Large Language Models to Generate New Features from Text Data for Loss Prediction

Guojun Gan^{1a}, Christopher Shultz^{2b}

¹ Department of Mathematics, University of Connecticut, ² The Hartford, USA

Keywords: Large language model, Text data, Loss prediction, Feature engineering, Prompt engineering

<https://doi.org/10.66573/001c.162423>

Variance

Vol. 19, 2026

Insurance companies collect large volumes of unstructured text data, such as claims descriptions, adjuster notes, and customer feedback. While these data contain valuable context, their unstructured nature leads to underutilization in traditional actuarial models. This paper investigates the use of large language models (LLMs) to extract low-dimensional structured features from claims descriptions to improve loss prediction models. With structured prompts, we apply GPT-4o and Llama-3.2-3B to classify incidents into ordinal risk categories. The resulting labels are semantically meaningful, ordinally consistent, and predictive of average loss. Incorporating them into generalized linear models improves both in-sample fit and out-of-sample accuracy, with five-category labels from GPT-4o yielding the best performance. These results demonstrate that LLMs can effectively augment traditional actuarial risk models by extracting meaningful information from unstructured text data.

1. INTRODUCTION

Text data are unstructured data that are stored and written in text format. Insurance companies collect an enormous amount of text data from multiple sources such as customer feedback, claims adjuster notes, underwriter notes, accident descriptions, etc. Text data provide more context than numerical data and can be analyzed to extract valuable insights.

Several methods are available for analyzing text data. One method applies pretrained language models, which can be used for a number of traditional natural language tasks, including summarization, sentiment analysis, etc. [Table 1](#) shows a timeline of language model development over the past three decades as they progressed through four major stages, from the initial statistical language models to the current stage of large language models (LLMs). Statistical language models use probability distributions to model word sequences. Neural language models use neural networks to capture complex patterns and representations of

language. Pretrained language models use large corpora and self-supervised learning to capture general linguistic knowledge. LLMs extend pretrained language models by processing massive amounts of data.

LLMs are language models (LMs) with massive parameter sizes that have exceptional capabilities to understand and generate human language (Chang et al. 2024). LLMs are built using deep learning techniques, are trained on vast amounts of text data, and can perform a variety of natural language processing tasks, such as text generation, translation, summarization, and question answering. LLMs have been applied in various fields, such as software engineering, computational biology, medicine, law, and education, to name just a few. Well-known LLMs include OpenAI's GPT-3 and GPT-4, Google's BERT, and Meta's Llama 1 and Llama 2. Wang et al. (2024) and Raiaan et al. (2024) have conducted comprehensive surveys of LLMs.

Two paradigms are used to apply pretrained LMs (Liu et al. 2023), (1) the pretrain and fine-tune paradigm, and (2) the pretrain, prompt, and predict paradigm. In the first,

a Dr. Gan is an associate professor in the Department of Mathematics at the University of Connecticut, where he has served since August 2014. Prior to his academic appointment, he spent six years at a major life insurance company in Toronto and one year at a hedge fund in Oakville, Canada. Dr. Gan holds a BS degree from Jilin University (2001) and MS and PhD degrees from York University (2003 and 2007, respectively). He is a Fellow of the Society of Actuaries (FSA). His research lies at the intersection of actuarial science and data science. Address for correspondence: guojun.gan@uconn.edu

b Dr. Shultz is the director of data science at The Hartford, which is a large property and casualty insurance company in the US, where he leads research and development efforts for developing and deploying predictive models for risk management, pricing, and process optimization. His research interests are in the application of machine learning and econometric techniques to address questions related to insurance and financial risk management. His work has been split between publishing in the applied economics literature and proprietary industry research and development.

Table 1. A timeline of language model development.
Source: Wang et al. (2024).

Era	Year - Milestone
Statistical Language Models	1990 - N-grams
Neural Language Models	2000 - Neural probabilistic LM
	2013 - Word2Vec
	2014 - RNN/LSTM
	2015 - Attention mechanism
Pretrained Language Models	2017 - Transformers
	2018 - BERT/GPT
	2019 - GPT-2
Large Language Models	2020 - GPT-3
	2022 - ChatGPT
	2023 - GPT-4
	2024 - GPT-4o

a pretrained LM is fine-tuned for downstream tasks with objective functions including additional parameters. This paradigm focuses on objective engineering that consists of designing the training objectives used at both the pretraining and fine-tuning stages. In the second paradigm, a pretrained LM is used for downstream tasks directly with the help of textual prompts. This paradigm focuses on prompt engineering that reformulates downstream tasks to look like those solved during the original LM training.

Our research investigated the use of pretrained LLMs to create features from text data with the help of textual prompts and studied the effectiveness of incorporating such features for predicting insurance losses. Our method has the following advantages: first, using pretrained LLMs circumvents a supervised learning condition, which requires large sets of labeled text data; second, the features produced by pretrained LLMs have lower dimensions than those produced by traditional text analysis techniques such as topic modeling; third, pretrained LLMs can understand complex texts and concepts in different languages through their analytical and machine translation capabilities.

The remainder of the paper is organized as follows. Section 2 provides a brief review of existing approaches to text data analysis as applied to insurance analytics and discusses potential applications of LLMs. Section 3 describes the public dataset containing the text descriptions that provided the basis for our study. Section 4 introduces the principles of prompt engineering. Section 5 presents the numerical results obtained by applying two LLMs to the dataset. Finally, Section 6 provides a summary and closing remarks.

2. LITERATURE REVIEW

Researchers in the actuarial community have used text data in their studies. For example, text mining techniques, from

topic modeling to vector embeddings, have been used to analyze text data in the insurance setting (Liao et al. 2019; Baillargeon et al. 2020; Zappa et al. 2021; Manski et al. 2021). While many studies have analyzed text data with traditional natural language processing (NLP) methods, the use of LLMs has been limited. This section provides a brief summary of the text-related explorations that have been published, as well as a foray into LLMs where applications exist.

Lee et al. (2019) considered how to enhance insurance analytics using textual data via the concept of word similarities. Word or sentence embedding models provide numerical representation of text data that retains semantic meaning. In the context of property insurance, for example, the embedding vector of a word like “fire” will be closer to the embedding of a word like “arson” than it will be to a word like “theft.” Actuarial applications can benefit from this encoding scheme. Lee et al. (2019) leveraged this approach by defining a procedure for extracting new numeric variables from text data and incorporating them into claims analytics. The authors demonstrated that the ability of models to predict and quantify risk is enhanced through the use of NLP methods.

Liao et al. (2019) investigated customer service practices of US personal lines insurance carriers using a variety of techniques, including word clouds, latent Dirichlet allocation for topic modeling, and a selection of pretrained sentiment analysis algorithms. Their study was an exploratory unsupervised analysis, demonstrating the potential for text data to enhance analytical capabilities in the actuarial space. The authors’ core finding was that unsupervised analysis of text data can identify frequent call topics, how customer sentiment differs between calls, and other factors in a highly automated fashion.

In the supervised analysis setting, Baillargeon et al. (2020) expanded the focus beyond traditional statistical actuarial models by exploring the potential of using neural architectures with unstructured text data to mine risk predictors from accident descriptions. The authors used features identified as important for assessing the underlying relationships as inputs to a Poisson regression model, from which they estimated the number of cars involved in an accident using the text description of the accident itself. In a similar setting, Zappa et al. (2021) used text narratives of accident occurrences from the US National Highway Traffic Safety Administration to extract semantic information that can be used in a regression context for fine-tuning premium decisions.

Manski et al. (2021) used text descriptions from a property claims dataset to predict risk. Using Word2Vec to condense the text descriptions into dense embeddings, they used a nonlinear transformation of the captured embeddings as inputs to a gamma double generalized linear model (GLM) (Smyth and Verbyla 1999) that regularizes for mean and dispersion. These studies demonstrate strong potential for using NLP techniques to supplement the rating and risk-determination process, but none of the models leverage LLMs.

Balona (2024) focused specifically on LLMs, aiming to summarize the existing and potential use cases for LLMs in actuarial work. The author defined the contributions of LLMs as directly beneficial to actuarial work as well as helping to make the process more efficient. Further, the author provided guidelines for when LLMs should be considered for actuarial work, considering factors such as data structure (e.g., if the data are highly structured and numeric, traditional approaches may be more suitable). The author noted that LLMs are technically best when the data are less structured, require context and pattern recognition, and/or require the generation of new content.

Li et al. (2025) proposed a comprehensive framework for examining discrepancies in textual content using LLMs. The authors focused on using the OpenAI interface to embed text and project it into categories, using distance metrics to evaluate discrepancies. The approach is based on three types of relationships: identical information, logical relationships, and potential relationships.

Some research on using LLMs to generate new features for predictive models exists outside the insurance field. Malberg et al. (2024) proposed an approach to use pre-trained LLMs to generate a set of human-interpretable features from text data. Their methodology generates human-interpretable features and outperforms traditional methods like term frequency-inverse document frequency (Salton and Buckley 1988) and vector embeddings. Hollmann et al. (2023) proposed an automated feature development method by using LLMs to iteratively generate meaningful features for tabular datasets.

Text embeddings play a central role in transforming unstructured text into numerical representations suitable for modeling and analysis. One of the earlier approaches—the term frequency-inverse document frequency (Salton and Buckley 1988)—represents documents using sparse vectors that capture word occurrence patterns; however, it lacks semantic meaning. Subsequent developments introduced static word embeddings, such as Word2Vec (Mikolov et al. 2013) and GloVe (Pennington et al. 2014), which learn dense vector representations based on word co-occurrence statistics and encode semantic relationships between words. However, these embeddings are context-independent, assigning the same vector to a word regardless of its usage. The introduction of contextual embeddings, such as Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al. 2019), addressed this limitation by generating dynamic representations that vary with surrounding text, enabling more accurate capture of linguistic nuance. Modern LLMs extend these ideas further—their deep transformer architectures produce high-dimensional embeddings that encode not only local syntax and semantics but also broader conceptual and relational information.

3. DESCRIPTION OF THE DATA

We used data from the Wisconsin Local Government Property Insurance Fund (LGPIF) for this study. The LGPIF was created to provide insurance to government properties not owned by the state of Wisconsin. The program made insurance available for local government properties like municipal buildings, schools, libraries, vehicles, etc. The dataset includes claims collected from 2006 to 2010 as a training set ($n = 4,991$) and claims from 2011 as a test set ($n = 1,039$).¹ The dataset includes nine binary variables (Vandalism, Fire, Lightning, Wind, Hail, Vehicle, WaterNW, WaterW, and Misc), one text variable (Description), and a continuous variable (Loss).

The LGPIF dataset has been used extensively in actuarial literature. Frees et al. (2016) used it for a case study that extends the literature on multivariate frequency-severity regression using copulas. The data have also been used to propose a multivariate framework for pricing property insurance contracts with multiperil coverage using a two-part model with heavy tails and Gaussian copulas (Yang and Shi 2018). A similar assessment is presented by Lee and Shi (2019). In another study, Lee et al. (2019) conducted a text-based analysis of the same dataset using the concept of word similarities to generate new features for regression modeling. The authors used the pretrained GloVe word embeddings framework to convert words in the LGPIF dataset claims descriptions into semantic embedding vectors. To extract features for modeling, they computed the cosine similarity between these word vectors and a curated set of keywords representing claims-related concepts (e.g., vandalism, fire, etc.). The similarity scores were fed into generalized additive models and GLMs to develop models that retain semantic information from text in the loss prediction framework.

Manski et al. (2021) used the LGPIF text data to predict loss amounts by applying a gamma double GLM approach. The authors first used GloVe to create sentence-level embeddings of each claim description by averaging the GloVe vectors of all words in the sentence. Next, they applied principal component analysis to the average embeddings and retained the first 10 principal components as summary features of the text. The 10 principal components were fed into a gamma double GLM model, which included a mean component (for expected loss) and a dispersion component (for variance), both modeled as functions of the text-derived features. Their analysis showed that text features significantly improve predictive performance and interoperability.

Other applications of the dataset include diagnostic testing (Li et al. 2023), dependency modeling (Lee 2023), hazard type prediction (Wüthrich and Merz 2023), and auto machine learning (Dong and Quan 2025). We identified a

¹ The dataset can be downloaded from the website of the paper by (Manski et al. 2021): <https://sites.google.com/view/doubleglm/home>. Accessed on October 27, 2025.

Table 2. Summary of the target variable loss by category.

(a) Training set.							
	Count	Min	1st Q	Mean	Median	3rd Q	Max
Hail	76	124.3	8,086.0	145,487.6	17,819.0	56,118.7	6,615,117.2
Fire	171	100.0	2,443.7	81,762.2	8,963.5	48,918.5	1,570,618.8
WaterW	426	1.0	3,009.0	80,432.4	8,898.3	33,019.5	12,922,217.8
Misc	362	1.0	1,641.2	29,149.5	3,928.7	12,072.4	2,633,822.4
WaterNW	202	1.0	2,000.8	23,973.8	6,306.3	19,688.0	1,114,594.5
Wind	296	1.0	2,156.7	18,125.4	4,960.1	12,245.8	492,477.9
Lightning	832	1.0	2,131.4	11,087.3	4,430.7	11,067.6	655,092.4
Vehicle	852	37.1	1,281.3	3,904.5	2,500.0	4,336.4	111,739.8
Vandalism	1774	6.0	274.5	2,084.4	586.6	1,368.4	207,565.0
Overall	4991	1.0	750.0	19,297.1	2,243.3	6,654.2	12,922,217.8

(b) Test set.							
	Count	Min	1st Q	Mean	Median	3rd Q	Max
Hail	18	7,886.0	18,563.2	103,674.1	49,184.0	170,570.8	332,412.0
Fire	46	125.4	5,123.9	83,767.0	11,355.4	43,656.4	964,150.0
WaterNW	67	544.0	4,246.1	60,538.5	6,739.1	15,187.1	2,672,184.2
WaterW	38	55.5	5,155.3	51,607.6	19,336.8	57,753.4	411,641.1
Wind	107	325.0	3,669.2	46,303.6	9,010.0	28,302.7	1,048,683.1
Lightning	123	500.0	2,937.7	11,622.7	5,000.0	10,895.1	88,603.0
Misc	103	70.0	1,045.5	9,723.0	3,025.0	9,223.2	242,918.4
Vandalism	310	1.0	314.7	6,190.4	500.0	1,485.5	981,598.7
Vehicle	227	1.0	1,515.1	5,662.0	3,000.0	5,126.1	135,268.0
Overall	1039	1.0	882.2	21,488.4	3,240.0	9,279.8	2,672,184.2

number of slight variations in the dataset across these sources and focused in particular on the property insurance claims across different loss types, as in Manski et al. (2021), with a particular focus on the text descriptions of claims.

Tables 2(a) and 2(b) provide a summary of the loss by category for the training set and the test set, respectively. The categories are sorted by the average loss. From the two tables, we see that hail caused the highest loss and fires caused the second highest loss. However, the average loss ranks of other categories differ for the training and test sets.

The summary statistics shown in Table 2 show that the loss variable is highly skewed because the overall mean is several times greater than the third quantile. Figures 1(a) and 1(b) show the histograms of the log-transformed loss for the training set and the test set, respectively. Let Y denote the loss variable. The logloss shown in the histograms is calculated by $\ln(1 + Y)$. The histograms show that the log-transformed loss is much less skewed. Since the log-transformed loss is positive, gamma distributions can be used to fit the data.

For these analyses, we were most interested in the text descriptions of each claim. Table 3 summarizes the textual characteristics of the training and test datasets. The training set contains 22,759 total words, roughly four times the 5,057 words in the test set, with 2,090 and 1,006 unique words, respectively. Both datasets have similar linguistic

Table 3. Summary statistics of text descriptions in the training set and the test set.

	Training set	Test set
Total number of words	22,759.00	5,057.00
Number of unique words	2,090.00	1,006.00
Average word length	6.31	6.29
Average number of words per description	4.56	4.87
Average ratio of stopwords	0.17	0.18
Yule's K	258.92	128.90
Herdan's C	0.76	0.81

patterns, with an average word length of about 6.3 characters and an average of approximately four to five words per description. The proportion of stopwords is also consistent across the two sets, around 17%–18%. Measures of lexical diversity show some variation: Yule's K is higher for the training set (258.9) than for the test set (128.9), indicating greater repetition of words, while Herdan's C is slightly lower in the training set (0.76 vs. 0.81), consistent with a more diverse vocabulary in the smaller test sample. Yule's K (Tweedie and Baayen 1998; Yule 2014) measures vocabulary repetitiveness, with larger values indicating less lexi-

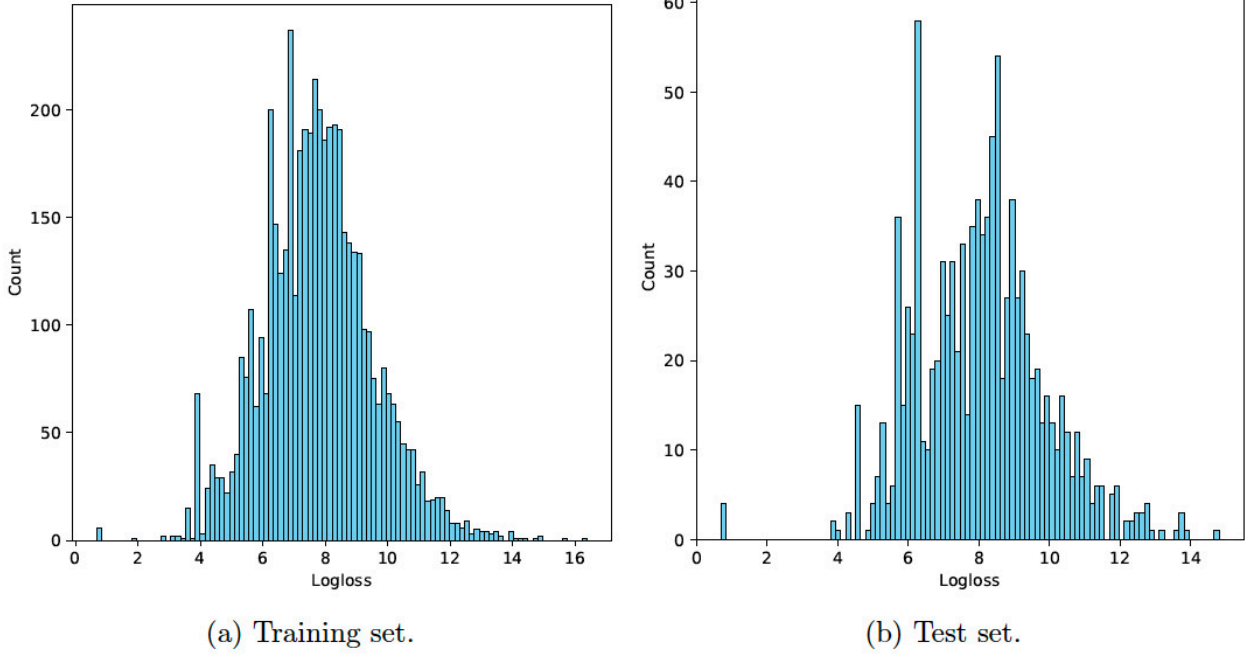


Figure 1. Histograms of the log transformed loss.

cal diversity, whereas Herdan’s C' (Herdan 1960) quantifies lexical richness on a scale between 0 and 1, where higher values represent a more varied vocabulary. Overall, the statistics suggest that the two datasets are comparable in linguistic structure, with minor differences in lexical richness and word repetition.

Figures 2(a) and 2(b) show the top keywords extracted from the claim descriptions in the training set and the test set, respectively. The top keywords are selected based on their frequency of occurrence in the incident descriptions. For example, the word “damage” appears about 2,000 times in the incident descriptions of the training set. As shown in the two figures, the sets of top keywords from the training set differ from those of the test set. However, the first keyword (i.e., damage) is the same for both the training set and the test set.

To dive a bit deeper into the semantic relationships captured within the claim descriptions, we first considered a conversion of claim descriptions to semantic embeddings via the all-mpnet-base-v2 model with the SentenceTransformer library. Next, we passed these embeddings into the 2d UMAP algorithm and constructed the scatter plot in Figure 3, with loss type represented by color and relative loss amount reflected by the size of the scatter plot point.

The UMAP process constructs a graph in high-dimensional space, where each node is a claim. Next, nearby claims are connected based on distances in embedding space. It estimates the data’s underlying structure and optimizes a low-dimensional layout that preserves the local relationships from the original graph. This process is similar to the t -SNE (t -distributed stochastic neighbor embedding) algorithm but is faster and more faithful to global structure (McInnes et al. 2018).

As a result, the proximity of points within the UMAP visual implies semantic similarity. A number of clusters emerged, demonstrating that the embedding model successfully captured differences in language associated with specific event types. Some overlap is visible, which may indicate shared language within descriptions. Larger circles represent relatively high-loss events. Large losses appear in multiple clusters, suggesting that loss isn’t driven by description alone, and multiple claim types (e.g., fire, wind, etc.) can cause significant loss. However, a lack of large losses is evident in some clusters (e.g., hail, lightning).

4. PROMPT ENGINEERING

This section introduces prompt engineering, which plays an essential role when using pretrained LLMs to create features from text data. A prompt is an input provided to an LLM to produce the desired output. In other words, a prompt is a set of instructions an LLM uses to predict the desired results. Prompt engineering refers to the process of designing and refining input prompts to optimize the performance of LLMs such as ChatGPT.

Figure 4 shows a schematic diagram of how to use a pretrained LLM to create features from text data.

4.1. PROMPT-BASED LEARNING

In traditional supervised learning systems for text data, we develop a model $P(y|x; \theta)$ to predict an output y (e.g., a label) for an input x . The model parameters θ are learned by using a dataset that contains pairs of inputs and outputs. The main issue with supervised learning is that labeled data are needed to train a model $P(y|x; \theta)$. For many tasks, however, it is difficult to find large quantities of labeled data.

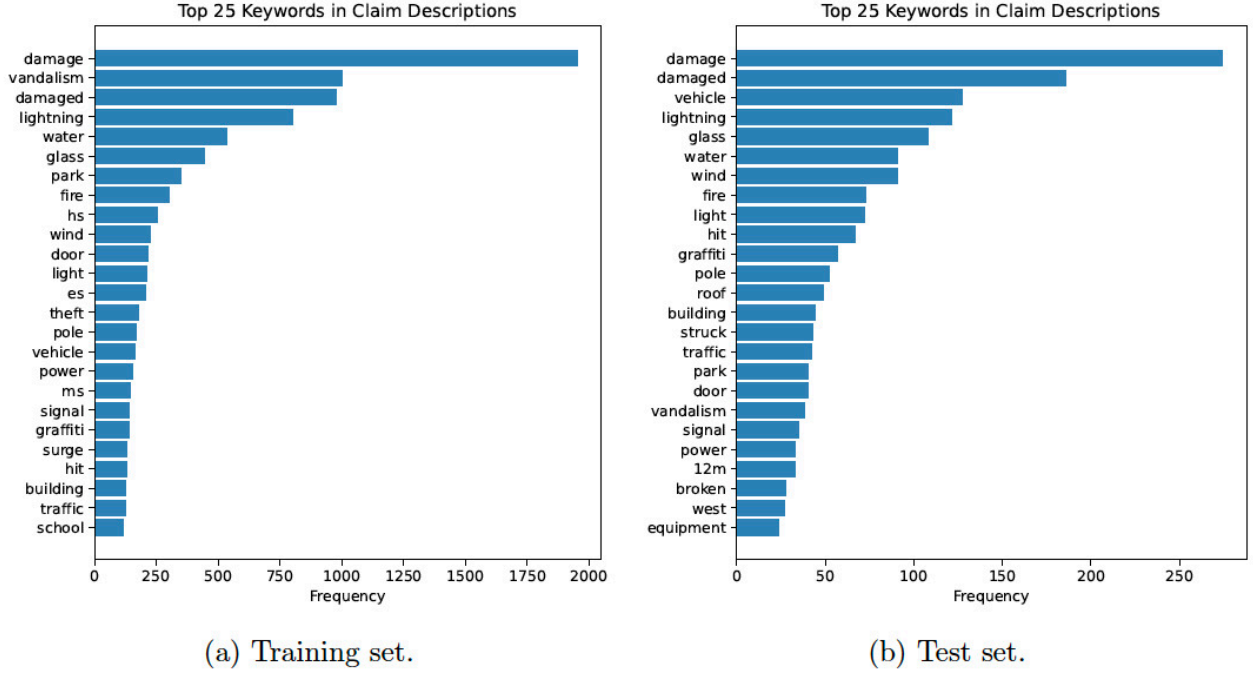


Figure 2. Distribution of common keywords in claim descriptions.

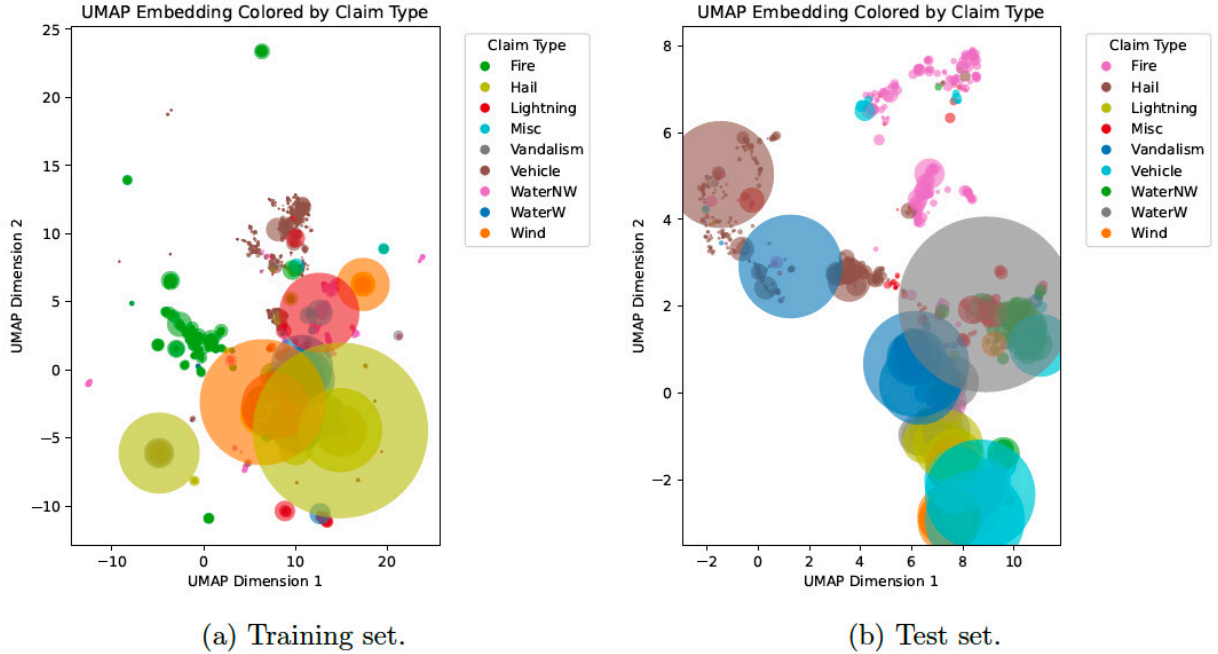


Figure 3. UMAP dimensions 1 and 2 of claim description embeddings.

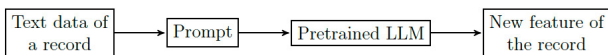


Figure 4. Schematic overview of using a pretrained LLM to generate new features.

Prompt-based learning circumvents this issue by using a pretrained LM that models the probability $P(\mathbf{x}; \boldsymbol{\theta})$ of text $\mathbf{x}$ itself and using this probability to predict $\mathbf{y}$.

Prompt-based learning consists of three major steps (Liu et al. 2023): prompt addition, answer search, and answer mapping. In the first step, a prompting function $f_p(\mathbf{x})$ is used to convert an input text $\mathbf{x}$ into a prompt $\mathbf{x}' = f_p(\mathbf{x})$. In the majority of existing work, the prompt function is a template that includes two slots, an input slot and an answer

slot. For example, the following template can be used as a prompt function for insurance text data:

$$f_p(\mathbf{x}) = \begin{array}{l} \text{Incident description: } \mathbf{x}. \\ \text{You are an assistant that classifies property} \\ \text{insurance incidents by risk severity into} \\ \text{one of three categories: 'low', 'med', or 'high'}. \\ \text{Only return the classification.} \end{array} \quad (1)$$

In the above prompt function, the first sentence is the input slot. The second and the third sentences form the answer slot.

In the second step, a pretrained LM is employed to search the highest-scoring text $\mathbf{z}$ based on filled prompts. For example, the filled prompt corresponding to the prompt given in Equation (1) can be:

$$f_{fill}(\mathbf{x}, \mathbf{z}) = \begin{array}{l} \text{Incident description: } \mathbf{x}. \\ \text{Risk severity classification: } \mathbf{z}. \end{array} \quad (2)$$

The pretrained LM will search over the set of potential answers by calculating the probability of the filled prompts:

$$\hat{\mathbf{z}} = \arg \max_{\mathbf{z} \in \mathcal{Z}} P(f_{fill}(\mathbf{x}, \mathbf{z}); \boldsymbol{\theta}),$$

where $\mathcal{Z} = \{\text{'low'}, \text{'med'}, \text{'high'}\}$.

In the third step, the highest-scoring answer $\hat{\mathbf{z}}$ is mapped to the highest-scoring output $\hat{\mathbf{y}}$. This step is usually trivial as the answer is the output required by the underlying task. In some situations, multiple answers will be mapped to the same output.

4.2. PRINCIPLES OF PROMPTING

Because LLMs generate outputs based on the prompt they receive, the quality and structure of the prompt significantly influences the accuracy, relevance, and usefulness of the response. Phoenix and Taylor (2024) introduced five principles of prompting, which we briefly introduce in this subsection.

The first principle is to give clear direction. The prompt should be clear, unambiguous, and specific. Vague prompts tend to produce vague answers. Role-playing is a good way to create prompts. For example, in the prompt given in Equation (1), we give the LLM the role of the assistant who is tasked to classify property insurance incidents.

You are an assistant that classifies property insurance incidents by risk severity into one of three categories: low, med, or high. Classify the following incident: lightning damage to tree, fire alarm panel.

You are an assistant that classifies property insurance incidents by risk severity into one of three categories: low, med, or high. Classify the following incident: lightning damage to tree, fire alarm panel. Only return the classification name.

The second principle is to specify the output format. LLMs such as ChatGPT can produce output in different formats. For classification tasks, if we only need the classification categories, we need to specify this in the prompt. For instance, if we give the prompt shown in the first example to ChatGPT, we see the following output:

Classification: med

Reasoning:

Lightning damage to a tree typically represents a relatively low risk unless it results in secondary

hazards (e.g., falling onto property or causing fire).

Damage to the fire alarm panel can have significant implications for fire safety and may require immediate repair or replacement to comply with safety codes.

The combination of electrical system impact and potential interruption to building safety mechanisms elevates the incident above low risk, but it does not involve structural damage, injury, or major fire, so it does not qualify as high risk.

If we use the prompt shown in the second example, the output produced by ChatGPT is “med.” By adding one additional instruction to the prompt, we got the classification category as the desired output.

The third principle is to provide examples. In the literature, a zero-shot prompt is where LLMs are asked to perform a task without giving any examples of how to do it. One-shot prompting provides one example, and few-shot prompting provides a few examples. Zero-shot prompting is useful when we want concise instructions or do not have labeled examples handy. It is especially effective with modern LLMs that have been trained on a wide variety of tasks.

The fourth principle is to evaluate the quality of the output generated by the language model. In many practical applications, this evaluation is performed informally through a process of trial and error—users iteratively modify prompts and assess the responses by manually inspecting their relevance, accuracy, or usefulness. While this approach can be effective, it is often inefficient and time-consuming, particularly when scaling up the use of LLMs.

A more systematic and scalable strategy is to incorporate a feedback or evaluation loop into the prompting workflow. This loop can include human-in-the-loop assessments, automated metrics, or comparison against ground truth data, depending on the task. Such a mechanism allows for more objective and consistent evaluation of outputs, enabling iterative refinement of prompts based on measured performance. However, it is important to consider the cost implications of this approach, especially when using commercial LLMs that charge per token. Each round of evaluation can incur significant expenses. Therefore, implementing an evaluation loop must strike a balance between improving prompt quality and managing computational and financial resources effectively.

The fifth principle is to divide the labor, which means breaking down a complex or large task into smaller, more manageable components. This modular approach not only simplifies the overall problem but also enables more focused and effective prompting for each subtask. By addressing one step at a time, users can design clearer prompts and achieve more reliable outputs from the language model.

4.3. BATCH PROMPTING

Batch prompting is the practice of sending multiple prompts at once to a language model in a single application programming interface (API) call or request. It is commonly used to process a set of similar tasks or inputs in parallel, rather than one at a time.

Batch prompting is beneficial in that it reduces overhead and latency by handling multiple prompts in one operation,

and it is cost-effective. Many LLM APIs (such as OpenAI’s ChatGPT) allow batching, which can reduce per-prompt costs.

Batch prompting has two approaches. The first is to combine multiple prompts in a list and send the list to an LLM in a single API call. The LLM then will return a list of responses. The second is to combine multiple prompts into a single prompt.

The following are descriptions of 5 insurance incidents:

1. lightning damage
2. lightning damage at Comm. Center
3. lightning damage at water tower
4. lightning damage to radio tower
5. vandalism damage at recycle center

You are an assistant assigned to classify the risk severity into one of three categories: low, med, or high. Return only the classification name for each of the incidents

For instance, if we give the prompt shown in the example above to ChatGPT, we see the following output:

1. med
2. med
3. high
4. high
5. med

This example shows that multiple incidents can be classified using a single prompt.

5. EMPIRICAL ANALYSIS

This section presents some empirical results of using LLMs to extract features from insurance text data. ChatGPT and Llama represent two leading LLMs. ChatGPT is a commercially available, highly polished model optimized for general-purpose use; it offers strong performance. In contrast, Llama is expected to continue the open-source tradition, enabling researchers and developers to download, fine-tune, and deploy models locally.

We used the ChatGPT-4o model (cutoff date October 2023) and the Llama-3.2-3B model (cutoff date December 2023) to classify incident severity into different categories based on incident descriptions.² ChatGPT-4o is hosted in the cloud, fine-tuned for safety and alignment, and designed for seamless use in commercial applications. Llama-3.2-3B is a smaller 3-billion-parameter open-weight model from Meta, designed for lightweight local deployment and research. The classification labels generated by the LLMs were subsequently incorporated into a gamma GLM to evaluate their effectiveness in predictive modeling. Figure 5 presents a flowchart illustrating the overall data analysis process.

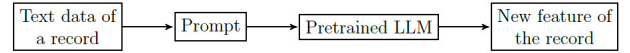


Figure 5. A flowchart of entire data analysis process.

5.1. CLASSIFICATION ACCURACY

We applied the two LLMs to classify the incident severity into two categories (low and high), three categories (low, medium, and high), and five categories (negligible, low, medium, high, and catastrophic). The cost of using ChatGPT-4o for these tasks was \$26, and the runtime was about 2 hours.

Tables 4, 5, and 6 show the contingency tables of the results produced by ChatGPT-4o and Llama-3.2-3B. For the two-category results, categories NA, Low, and Hig denote unknown, low, and high, respectively. For the three-category results, categories NA, Low, Med, and Hig correspond to unknown, low, medium, and high, respectively. For the five-category results, categories NA, Neg, Low, Med, Hig, and Cat correspond to unknown, negligible, low, medium, high, and catastrophic, respectively. We assigned the output of an LLM to the category “unknown” when the model did not produce a valid classification. This occurred in cases where the text description of an incident was missing or contained no meaningful information, or when the LLM failed to generate a classification and returned an error.

Table 4 shows the contingency tables between the two-category result and the three-category result. Table 4(a) shows that when ChatGPT-4o was prompted to classify incidents into two categories, 2,892 (1 + 1,042 + 1,849) incidents were classified into the low category. When ChatGPT-4o was prompted to classify incidents into three categories, 3,637 (1,849 + 1,788) incidents were classified into the medium category. The low-category incidents from the two-category result were split into the low-category and the medium-category in the three-category result.

Comparing Tables 4(a) and 4(b) shows that ChatGPT-4o produced more consistent results than Llama-3.2-3B. For example, low-category incidents from the two-category result were not classified into the high-category from the three-category result by ChatGPT-4o. However, the two-category result and the three-category result produced by Llama-3.2-3B were mixed.

Figures 6, 7, and 8 show the box plots of the log-loss by risk categories for the training set. The risk categories were produced by ChatGPT-4o and Llama-3.2-3B based on the incident text descriptions in the training set. The category NA includes the incidents not classified by the LLMs. Categories Neg, Low, Med, Hig, and Cat denote incidents with severity levels from low to high. Summary statistics corresponding to these figures are also provided in Appendix A. Given the increasing patterns of the median log-loss in the

² The relevant code is available at <https://github.com/casact>.

Table 4. Contingency tables of the two-category and three-category results produced by ChatGPT-4o and Llama-3.2-3B on the training set.

(a) ChatGPT-4o.						(b) Llama-3.2-3B.					
	NA	Low	Med	Hig	Total		NA	Low	Med	Hig	Total
NA	6	8	0	0	14	NA	1	7	9	2	19
Low	1	1042	1849	0	2892	Low	7	693	569	444	1713
Hig	1	1	1788	295	2085	Hig	16	721	1069	1453	3259
Total	8	1051	3637	295	4991	Total	24	1421	1647	1899	4991

Table 5. Contingency tables of the two-category and five-category results produced by ChatGPT-4o and Llama-3.2-3B on the training set.

(a) ChatGPT-4o.							
	NA	Neg	Low	Med	Hig	Cat	Total
NA	14	0	0	0	0	0	14
Low	16	506	2257	113	0	0	2892
Hig	2	1	397	1374	192	119	2085
Total	32	507	2654	1487	192	119	4991

(b) Llama-3.2-3B.							
	NA	Neg	Low	Med	Hig	Cat	Total
NA	0	7	10	0	1	1	19
Low	21	339	690	355	169	139	1713
Hig	66	419	967	517	474	816	3259
Total	87	765	1667	872	644	956	4991

Table 6. Contingency tables of the three-category and five-category results produced by ChatGPT-4o and Llama-3.2-3B on the training set.

(a) ChatGPT-4o.							
	NA	Neg	Low	Med	Hig	Cat	Total
NA	8	0	0	0	0	0	8
Low	22	490	539	0	0	0	1051
Med	2	17	2115	1452	51	0	3637
Hig	0	0	0	35	141	119	295
Total	32	507	2654	1487	192	119	4991

(b) Llama-3.2-3B.							
	NA	Neg	Low	Med	Hig	Cat	Total
NA	1	3	9	2	4	5	24
Low	36	338	650	210	88	99	1421
Med	20	240	587	345	205	250	1647
Hig	30	184	421	315	347	602	1899
Total	87	765	1667	872	644	956	4991

box plots, the risk categories obtained by both LLMs make sense.

Tables 7, 8, and 9 and Figures 9, 10, and 11 show the results for the test set. The resulting patterns are similar to those of the training set.

In summary, the contingency tables show that the results produced by ChatGPT-4o are more consistent than

those produced by Llama-3.2-3B. The box plots show that the results produced by both ChatGPT-4o and Llama-3.2-3B are meaningful.

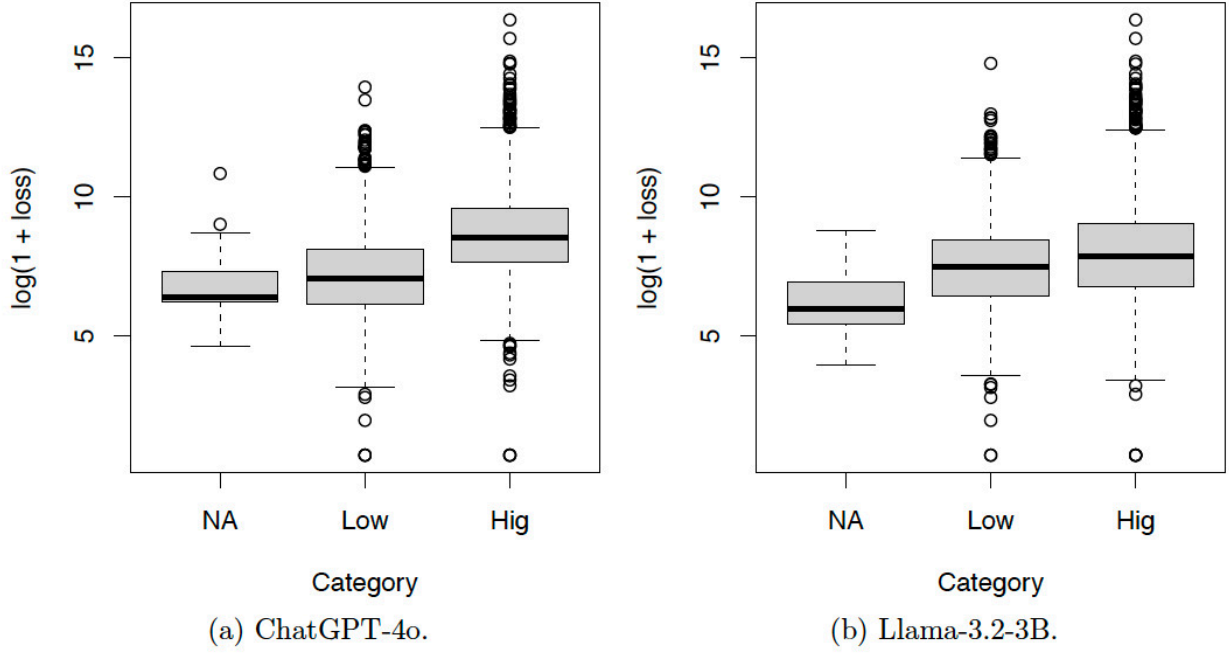


Figure 6. Box plots of log-loss by risk classes for the two-category approach produced by ChatGPT-4o and Llama-3.2-3B on the training set.

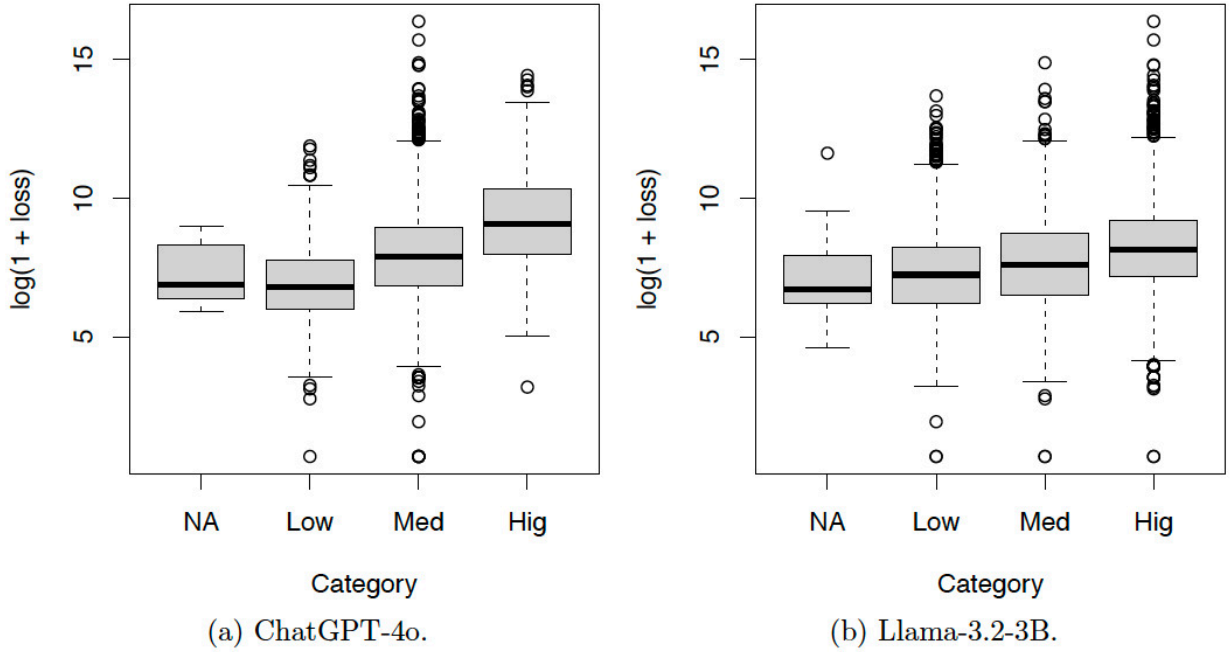


Figure 7. Box plots of log-loss by risk classes for the three-category approach produced by ChatGPT-4o and Llama-3.2-3B on the training set.

5.2. LOSS PREDICTION

This section presents the loss prediction results by incorporating the risk categories produced by LLMs. The summary statistics of the loss variable are shown in [Table 2](#). The risk categories produced by LLMs are ordinal categorical vari-

ables. It is common to treat ordinal categorical variables as continuous variables in regression analysis.

[Figure 12](#) shows the line plots of the average log-loss against the risk levels, which are integers used to denote the risk categories. For the five-category risk classification, 1 to 5 denote the risk categories from negligible to catastrophic. The line plots show that the mean log-loss and

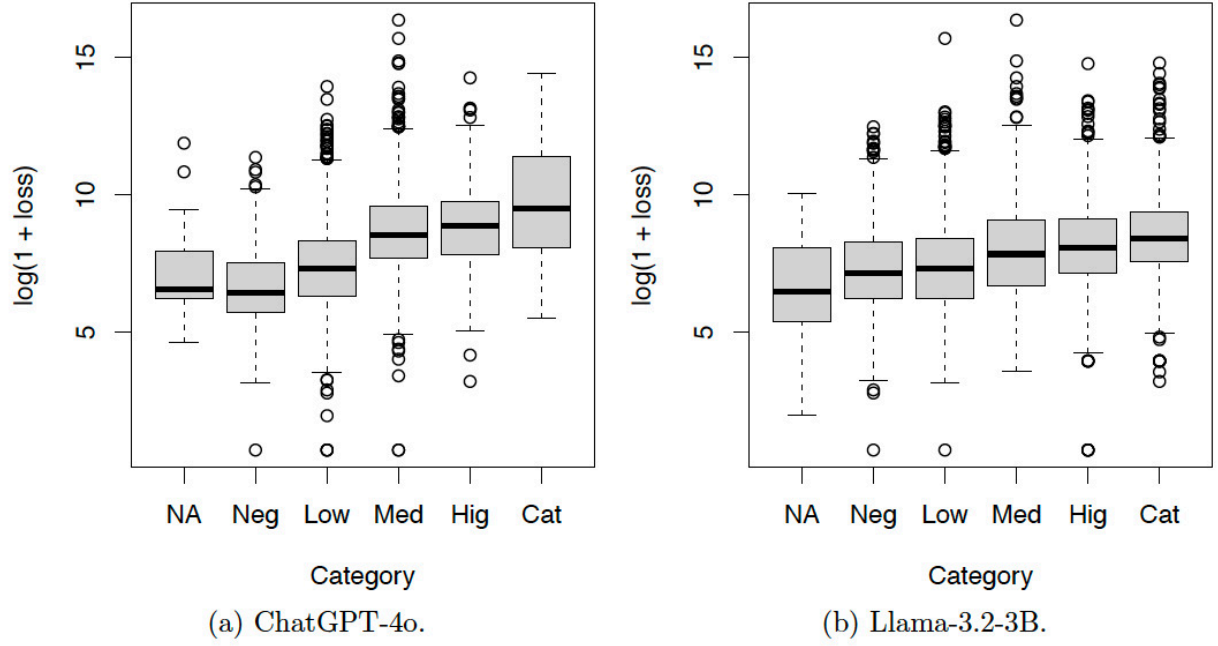


Figure 8. Box plots of log-loss by risk classes for the five-category approach produced by ChatGPT-4o and Llama-3.2-3B on the training set.

Table 7. Contingency tables of the two-category and three-category results produced by ChatGPT-4o and Llama-3.2-3B on the test set.

(a) ChatGPT-4o.						(b) Llama-3.2-3B.					
	NA	Low	Med	Hig	Total		NA	Low	Med	Hig	Total
NA	0	1	0	0	1	NA	1	0	1	0	2
Low	0	305	228	1	534	Low	1	121	147	86	355
Hig	0	0	419	85	504	Hig	3	120	222	337	682
Total	0	306	647	86	1039	Total	5	241	370	423	1039

Table 8. Contingency tables of the two-category and five-category results produced by ChatGPT-4o and Llama-3.2-3B on the test set.

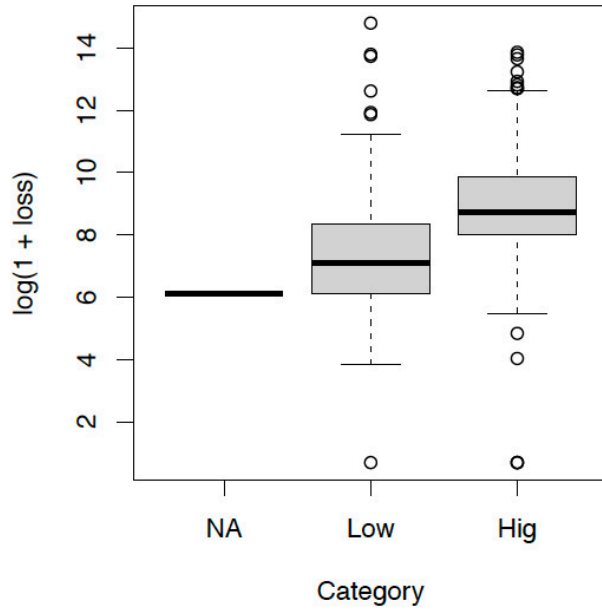
(a) ChatGPT-4o.							
	NA	Neg	Low	Med	Hig	Cat	Total
NA	1	0	0	0	0	0	1
Low	2	242	281	8	0	1	534
Hig	0	1	115	303	57	28	504
Total	3	243	396	311	57	29	1039

(b) Llama-3.2-3B.							
	NA	Neg	Low	Med	Hig	Cat	Total
NA	0	0	2	0	0	0	2
Low	2	86	142	58	24	43	355
Hig	6	92	184	116	111	173	682
Total	8	178	328	174	135	216	1039

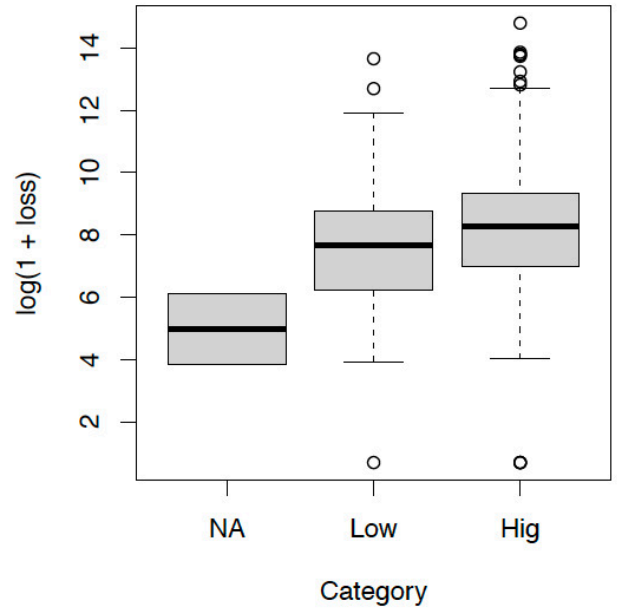
Table 9. Contingency tables of the three-category and five-category results produced by ChatGPT-4o and Llama-3.2-3B on the test set.

(a) ChatGPT-4o.							
	NA	Neg	Low	Med	Hig	Cat	Total
NA	0	0	0	0	0	0	0
Low	3	236	67	0	0	0	306
Med	0	7	329	301	10	0	647
Hig	0	0	0	10	47	29	86
Total	3	243	396	311	57	29	1039

(b) Llama-3.2-3B.							
	NA	Neg	Low	Med	Hig	Cat	Total
NA	1	0	2	1	0	1	5
Low	0	57	97	38	21	28	241
Med	3	65	137	68	42	55	370
Hig	4	56	92	67	72	132	423
Total	8	178	328	174	135	216	1039



(a) ChatGPT-4o.



(b) Llama-3.2-3B.

Figure 9. Box plots of log-loss by risk classes produced by ChatGPT-4o and Llama-3.2-3B on the test set.

the risk levels have approximately linear relationships. We can treat the ordinal risk categories obtained by LLMs as continuous variables in regression analysis.

We fitted gamma GLMs with the log link function to the training set. Then we used the fitted model to predict the log-loss for the test set. Table 10 shows the in-sample and out-of-sample validation measures of the fitted values from the gamma GLMs. The R^2 and R_a^2 were calculated as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad (3)$$

$$R_a^2 = 1 - \frac{(n-1) \sum_{i=1}^n (y_i - \hat{y}_i)^2}{(n-k-1) \sum_{i=1}^n (y_i - \bar{y})^2}, \quad (4)$$

where y_i and $\hat{y}_i$ denote the observed log-loss and predicted log-loss for the i th observation, $\bar{y}$ is the average observed log-loss, n is the number of observations of the underlying dataset, and k is the number of covariates, excluding the intercept. The mean absolute error (MAE) was calculated as follows:

$$MAE = \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{n}. \quad (5)$$

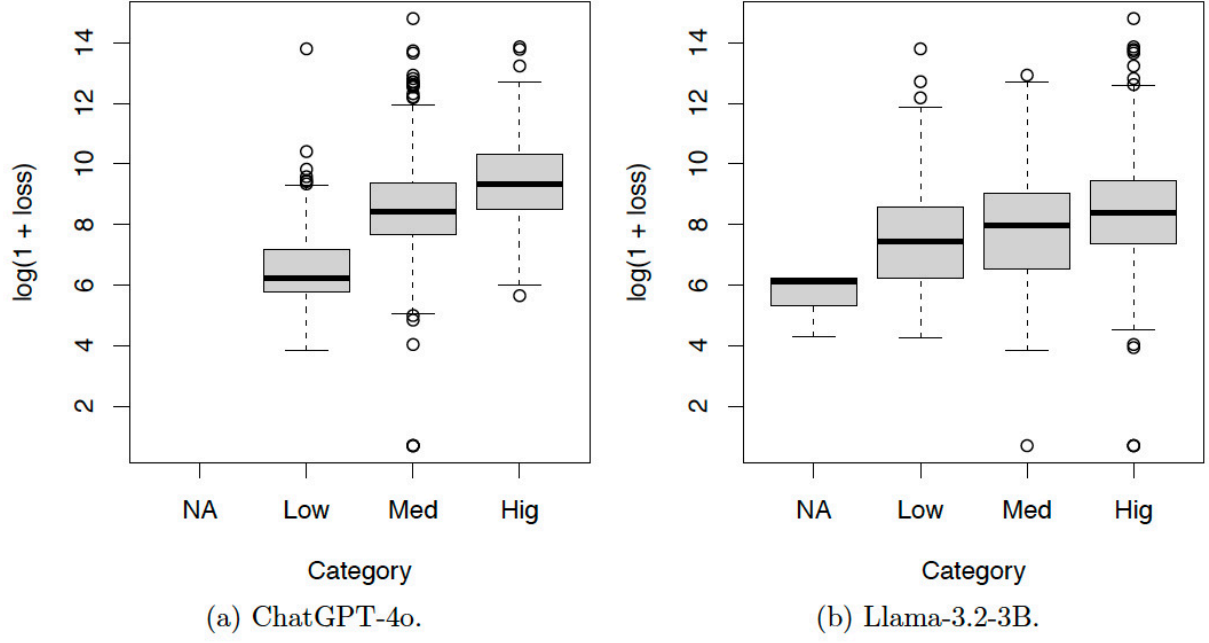


Figure 10. Box plots of log-loss by risk classes produced by ChatGPT-4o and Llama-3.2-3B on the test set.

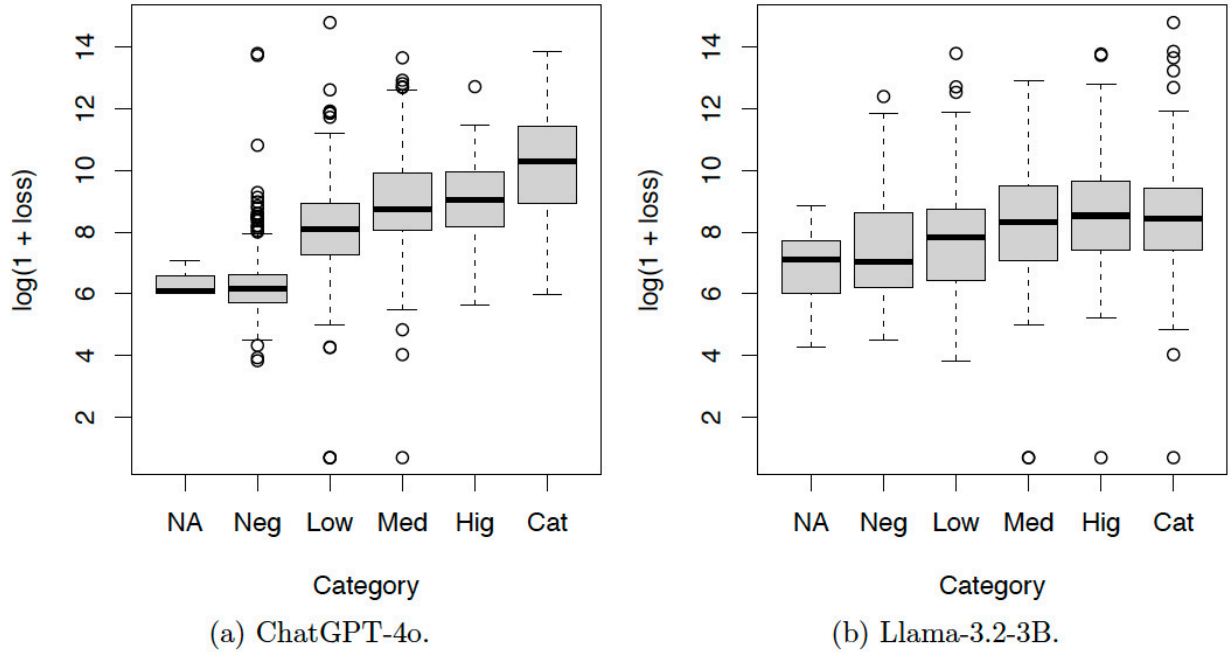


Figure 11. Box plots of log-loss by risk classes produced by ChatGPT-4o and Llama-3.2-3B on the test set.

The validation measures can be interpreted as follows. For R^2 and R_a^2 , a higher value indicates a better fit. For MAE, a lower value indicates better predictive accuracy.

The validation measures in Table 10 show that including the labels obtained by LLMs produced a better fit. For example, the out-of-sample R^2 with the five-category label from ChatGPT-4o is 0.3971, which is higher than 0.3398 for the out-of-sample R^2 without any LLM labels. The R_a^2 and the MAE are consistent with this observation.

However, the R^2 values for the Llama-3.2-3B model are lower than those for the ChatGPT-4o model, indicating that ChatGPT-4o achieved better classification performance. This result is expected, given that ChatGPT-4o is believed to have over 1.8 trillion parameters, whereas Llama-3.2-3B has 3 billion. The significantly larger model size of ChatGPT-4o enables it to capture more complex patterns and relationships in the data, leading to more accurate predictions.

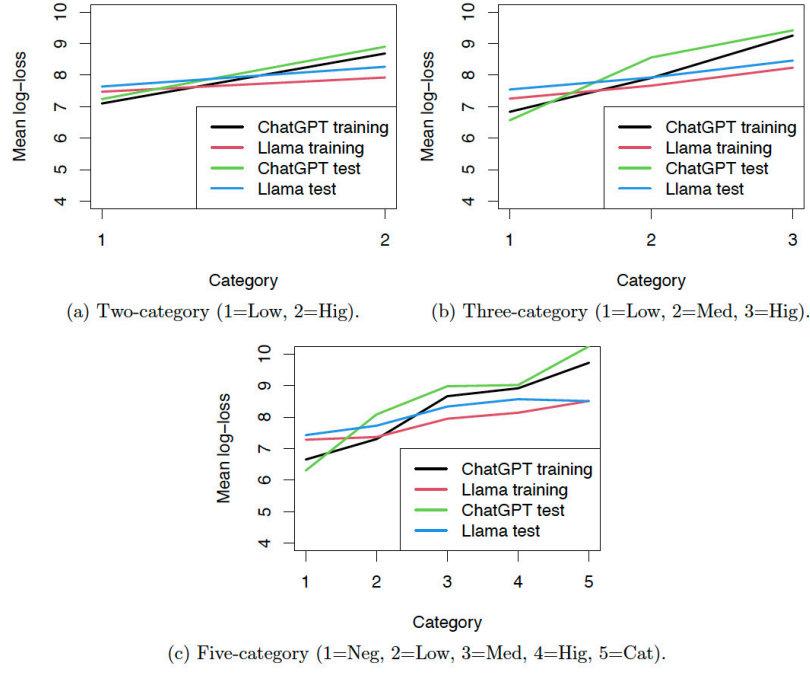


Figure 12. Line plots of the mean log-loss against the risk levels.

Table 10. Validation measures from the gamma regression models.

(a) ChatGPT-4o.						
	In-sample			Out-of-sample		
	R^2	R_a^2	MAE	R^2	R_a^2	MAE
No LLM-based label	0.3742	0.3732	1.0585	0.3398	0.3347	1.0694
Two-category label	0.3974	0.3963	1.0382	0.3820	0.3766	1.0176
Three-category label	0.3911	0.3900	1.0401	0.3750	0.3695	1.0133
Five-category label	0.4015	0.4004	1.0332	0.3971	0.3918	0.9889
(b) Llama-3.2-3B.						
	In-sample			Out-of-sample		
	R^2	R_a^2	MAE	R^2	R_a^2	MAE
No LLM-based label	0.3742	0.3732	1.0585	0.3398	0.3347	1.0694
Two-category label	0.3762	0.3751	1.0572	0.3430	0.3373	1.0671
Three-category label	0.3831	0.3820	1.0505	0.3510	0.3454	1.0646
Five-category label	0.3788	0.3777	1.0566	0.3498	0.3442	1.0589

Table 11 shows the regression coefficients and their standard errors from the gamma GLMs fitted to the training set when the risk classification labels produced by ChatGPT-4o were used. Tables 11(b), 11(c), and 11(d) show that the labels produced by ChatGPT-4o are significant.

Table 12 shows the regression coefficients and their standard errors from the gamma GLMs fitted to the training set when the risk classification labels produced by Llama-3.2-3B were used. Tables 12(a), 12(b), and 12(c) show that the labels produced by Llama-3.2-3B are also significant.

We also compared our results with those obtained by Manski et al. (2021), who used correlation coefficients to measure the predictive performance. Table 13 reports the

correlations between the observed log-loss and the log-loss predicted by different models for both the training and test sets. The baseline model without LLM-based labels achieved correlations of 0.6117 and 0.5956 on the training and test sets, respectively. Incorporating LLM-generated categorical labels improves these correlations, particularly for ChatGPT-4o, whose five-category labeling yields the highest out-of-sample correlation (0.6426). Results from Llama-3.2-3B show modest improvements over the baseline but remain slightly below ChatGPT-4o's performance. Compared with the gamma double GLM from (Manski et al. 2021), which performs well on the training data (0.6638) but poorly on the test set (0.4868), the LLM-augmented

Table 11. Regression coefficients and their standard errors from the gamma regression models fitted to the log-loss data. The risk classification labels were produced by ChatGPT-4o.

(a) Without risk levels from LLMs.				(b) With two-category risk levels.			
	Est.	Std. Err.	P-Value		Est.	Std. Err.	P-value
(Intercept)	2.1275	0.0095	0.0000	(Intercept)	1.9920	0.0133	0.0000
Vandalism	-0.2637	0.0105	0.0000	Vandalism	-0.2298	0.0105	0.0000
Fire	0.1108	0.0168	0.0000	Fire	0.0708	0.0168	0.0000
Lightning	0.0106	0.0114	0.3537	Lightning	-0.0274	0.0115	0.0177
Wind	0.0326	0.0142	0.0220	Wind	0.0409	0.0139	0.0033
Hail	0.1737	0.0229	0.0000	Hail	0.1673	0.0224	0.0000
Vehicle	-0.0811	0.0114	0.0000	Vehicle	-0.0618	0.0112	0.0000
WaterNW	0.0392	0.0159	0.0139	WaterNW	0.0284	0.0156	0.0693
WaterW	0.0927	0.0130	0.0000	WaterW	0.0741	0.0128	0.0000
				label	0.0910	0.0064	0.0000

(c) With three-category risk levels.				(d) With five-category risk levels.			
	Est.	Std. Err.	P-value		Est.	Std. Err.	P-value
(Intercept)	2.0018	0.0141	0.0000	(Intercept)	1.9986	0.0126	0.0000
Vandalism	-0.2511	0.0103	0.0000	Vandalism	-0.2445	0.0103	0.0000
Fire	0.0549	0.0173	0.0015	Fire	0.0023	0.0180	0.8975
Lightning	-0.0043	0.0113	0.7028	Lightning	-0.0268	0.0115	0.0197
Wind	0.0327	0.0140	0.0196	Wind	0.0293	0.0139	0.0351
Hail	0.1645	0.0226	0.0000	Hail	0.1606	0.0224	0.0000
Vehicle	-0.0661	0.0113	0.0000	Vehicle	-0.0718	0.0111	0.0000
WaterNW	0.0337	0.0157	0.0317	WaterNW	0.0158	0.0157	0.3126
WaterW	0.0845	0.0128	0.0000	WaterW	0.0665	0.0128	0.0000
label	0.0670	0.0057	0.0000	label	0.0571	0.0039	0.0000

models demonstrate better generalization and more consistent predictive power across datasets.

5.3. REPRODUCIBILITY

Despite their strong performance, LLMs pose significant challenges for reproducibility. Unlike traditional statistical or machine-learning models, LLMs are typically proprietary and hosted through cloud-based APIs, meaning users cannot control or access the model weights or full training data. Consequently, model updates by providers can alter outputs even when identical prompts and parameters are used. Moreover, stochastic generation mechanisms, such as temperature sampling, introduce additional variability unless explicitly fixed. Differences in hardware, API versions, and tokenization can further affect results.

[Table 14](#) shows the default values of the temperature parameter and the top-p parameter, which control the randomness of the LLMs' outputs. Higher values for either parameter increase variability and creativity in responses. Lower values make the model's outputs more deterministic and reproducible across runs. In all experiments described in this paper, the default values of the parameters were used as we did not specify any parameters for the LLMs.

To examine the variability of labels generated by LLMs, we ran the Llama-3.2-3B model four additional times. [Table 15](#) presents the Chi-squared test statistics comparing pairs

of label sets across the five runs, including the run discussed earlier. Since the critical value of the Chi-squared distribution with 25 degrees of freedom at the 1% significance level is 44.31, the results suggest that the labels generated across different runs of the LLM exhibit strong dependence.

[Table 16](#) shows the validation measures of the gamma regression model using labels from five runs of the Llama-3.2-3B model. The table shows that the validation measures are consistent across runs, suggesting stable model performance.

5.4. DATA EXPOSURE

Pretrained LLMs are trained by using massive amounts of data from the internet. Because the dataset used in our study was obtained from the internet, it is likely that the data were used to train the LLMs. To examine whether the dataset was used to train the Llama-3.2-3B model, which is used in our study, we prompted Llama-3.2-3B to generate loss amount by providing the incident descriptions.

[Figure 13](#) presents scatter plots of the observed loss amounts versus the loss amounts generated by Llama-3.2-3B, both in log scale. The plots show that the points are dispersed around the center, suggesting that the generated losses do not exhibit a systematic relationship with the observed losses and appear largely random.

Table 12. Regression coefficients and their standard errors from the gamma regression models fitted to the log-loss data. The risk classification labels were produced by Llama-3.2-3B.

(a) With two-category risk levels.				(b) With three-category risk levels.			
	Est.	Std. Err.	P-value		Est.	Std. Err.	P-value
(Intercept)	2.0941	0.0130	0.0000	(Intercept)	2.0666	0.0118	0.0000
Vandalism	-0.2628	0.0105	0.0000	Vandalism	-0.2571	0.0104	0.0000
Fire	0.1057	0.0169	0.0000	Fire	0.0926	0.0169	0.0000
Lightning	0.0052	0.0115	0.6542	Lightning	-0.0064	0.0115	0.5766
Wind	0.0316	0.0142	0.0261	Wind	0.0321	0.0141	0.0230
Hail	0.1692	0.0229	0.0000	Hail	0.1662	0.0227	0.0000
Vehicle	-0.0781	0.0114	0.0000	Vehicle	-0.0757	0.0113	0.0000
WaterNW	0.0398	0.0159	0.0124	WaterNW	0.0460	0.0158	0.0037
WaterW	0.0922	0.0130	0.0000	WaterW	0.0951	0.0129	0.0000
label	0.0205	0.0055	0.0002	label	0.0290	0.0034	0.0000

(c) With five-category risk levels.			
	Est.	Std. Err.	P-value
(Intercept)	2.0871	0.0115	0.0000
Vandalism	-0.2524	0.0106	0.0000
Fire	0.0959	0.0170	0.0000
Lightning	-0.0052	0.0117	0.6563
Wind	0.0361	0.0142	0.0110
Hail	0.1763	0.0228	0.0000
Vehicle	-0.0756	0.0114	0.0000
WaterNW	0.0452	0.0159	0.0045
WaterW	0.0967	0.0129	0.0000
label	0.0133	0.0021	0.0000

Table 13. Correlations of the observed log-loss and the log-loss predicted by different models.

	In-sample	Out-of-sample
No LLM-based label	0.6117	0.5956
ChatGPT-4o		
Two-category label	0.6304	0.6238
Three-category label	0.6254	0.6250
Five-category label	0.6337	0.6426
Llama-3.2-3B		
Two-category label	0.6134	0.5979
Three-category label	0.6189	0.6026
Five-category label	0.6155	0.6037
Manski et al. (2021)		
Gamma double GLM	0.6638	0.4868

6. CONCLUSIONS

This paper examined the use of LLMs to generate structured features from unstructured text data in the insurance setting, particularly with the purpose of improving loss prediction. We demonstrate that pretrained LLMs, specifically GPT-4o and Llama-3.2-3B, can effectively classify incident

Table 14. Default parameter values used by ChatGPT-4o and Llama-3.2-3B.

	ChatGPT-4o	Llama-3.2-3B
Temperature	1.0	0.8
Top-p	1.0	0.9

descriptions into ordinal severity categories via prompt-engineering techniques. These model-generated labels exhibit ordinal consistency, align with average loss magnitudes, and serve as informative predictors when incorporated into GLMs for severity estimation.

Empirical results indicate that the incorporation of LLM-derived features improves both in-sample and out-of-sample performance relative to models using only traditional variates. In particular, the five-category prediction yields the best accuracy, with higher R^2 and lower MAE than baseline models.

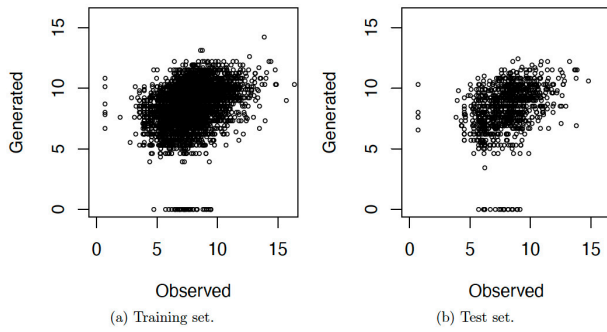
These findings have important implications for actuarial science. First, they show that LLMs can serve as effective, low-cost tools for augmenting traditional actuarial models with context from unstructured text data. Second, the success of prompt-based approaches supports their applicability in settings where fine-tuning is infeasible. Finally, the

Table 15. Chi-squared test statistics between pairs of the five-category labels from the five runs of the Llama-3.2-3B model.

	Training set				Test set			
	Run2	Run3	Run4	Run5	Run2	Run3	Run4	Run5
Run1	1,038.97	1,295.57	1,181.05	1,288.82	160.43	164.25	108.88	154.94
Run2	-	1,057.99	1,019.97	1,153.72	-	175.10	145.16	153.89
Run3	-	-	1,065.89	1,257.39	-	-	135.32	173.70
Run4	-	-	-	1,244.54	-	-	-	130.36

Table 16. Validation measures of the gamma regression model with five-category labels from five runs of the Llama-3.2-3B model.

	In-sample			Out-of-sample		
	R^2	R_a^2	MAE	R^2	R_a^2	MAE
Run1	0.3788	0.3777	1.0566	0.3498	0.3442	1.0589
Run2	0.3795	0.3783	1.0540	0.3535	0.3479	1.0599
Run3	0.3793	0.3781	1.0531	0.3569	0.3513	1.0546
Run4	0.3787	0.3776	1.0554	0.3457	0.3400	1.0685
Run5	0.3804	0.3792	1.0543	0.3534	0.3478	1.0597

**Figure 13. Scatter plots of the observed loss and the loss generated by Llama-3.2-3B in the log scale.**

interpretability of ordinal labels offers practical value for downstream modeling and decision makers.

Future work may investigate ensemble approaches, the integration of multiple LLMs, or hybrid models that combine LLM-generated features with traditional NLP tech-

niques. Additionally, as open-weight models continue to evolve, evaluating their performance parity with proprietary models like GPT-4o remains an active area of interest. Overall, this research underscores the promise of LLMs in augmenting actuarial models and opens new directions for the use of text data in insurance analytics.

ACKNOWLEDGMENTS

Guojun Gan and Christopher Shultz would like to acknowledge the financial support provided by the Casualty Actuarial Society (CAS). This research was supported in part through computational resources provided by the University of Connecticut, Storrs, CT.

Submitted: July 07, 2025 EDT. Accepted: April 14, 2026 EDT.

Published: June 15, 2026 EDT.

REFERENCES

- Baillargeon, Jean-Thomas, Luc Lamontagne, and Etienne Marceau. 2020. "Mining Actuarial Risk Predictors in Accident Descriptions Using Recurrent Neural Networks." *Risks* 9 (1): 7. <https://doi.org/10.3390/risks9010007>.
- Balona, Caesar. 2024. "ActuaryGPT: Applications of Large Language Models to Insurance and Actuarial Work." *British Actuarial Journal* 29. <https://doi.org/10.1017/s1357321724000102>.
- Chang, Yupeng, Xu Wang, Jindong 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>.
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding." *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)* (Minneapolis, Minnesota), June, 4171–86. <https://doi.org/10.18653/v1/N19-1423>.
- Dong, Panyi, and Zhiyu Quan. 2025. "Automated Machine Learning in Insurance." *Insurance: Mathematics and Economics* 120 (January): 17–41. <https://doi.org/10.1016/j.insmatheco.2024.10.002>.
- Frees, Edward, Gee Lee, and Lu Yang. 2016. "Multivariate Frequency–Severity Regression Models in Insurance." *Risks* 4 (1): 4. <https://doi.org/10.3390/risks4010004>.
- Herdan, Gustav. 1960. *Type-Token Mathematics: A Textbook of Mathematical Linguistics*. Mouton.
- Hollmann, Noah, Samuel Müller, and Frank Hutter. 2023. "Large Language Models for Automated Data Science: Introducing CAAFE for Context-Aware Automated Feature Engineering." *Advances in Neural Information Processing Systems* 36: 44753–75.
- Lee, Gee Y. 2023. "Multivariate Insurance Portfolio Risk Retention Using the Method of Multipliers." *North American Actuarial Journal* 27 (4): 787–805. <https://doi.org/10.1080/10920277.2022.2161578>.
- Lee, Gee Y, Scott Manski, and Tapabrata Maiti. 2019. "Actuarial Applications of Word Embedding Models." *ASTIN Bulletin* 50 (1): 1–24. <https://doi.org/10.1017/asb.2019.28>.
- Lee, Gee Y., and Peng Shi. 2019. "A Dependent Frequency–Severity Approach to Modeling Longitudinal Insurance Claims." *Insurance: Mathematics and Economics* 87 (July): 115–29. <https://doi.org/10.1016/j.insmatheco.2019.04.004>.
- 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, Yinhuan, Tsz Chai Fung, Liang Peng, and Linyi Qian. 2023. "Diagnostic Tests before Modeling Longitudinal Actuarial Data." *Insurance: Mathematics and Economics* 113 (November): 310–25. <https://doi.org/10.1016/j.insmatheco.2023.09.002>.
- Liao, Xiyue, Guoqiang Chen, Ben Ku, Rahul Narula, and Janet Duncan. 2019. "Text Mining Methods Applied to Insurance Company Customer Calls: A Case Study." *North American Actuarial Journal* 24 (1): 153–63. <https://doi.org/10.1080/10920277.2019.1649155>.
- Liu, Pengfei, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. "Pre-Train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing." *ACM Computing Surveys* 55 (9): 1–35. <https://doi.org/10.1145/3560815>.
- Malberg, Simon, Edoardo Mosca, and Georg Groh. 2024. "FELIX: Automatic and Interpretable Feature Engineering Using LLMs." In *Machine Learning and Knowledge Discovery in Databases. Research Track*. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-70359-1_14.
- Manski, Scott, Kaixu Yang, Gee Y. Lee, and Tapabrata Maiti. 2021. "Loss Amount Prediction from Textual Data Using a Double GLM with Shrinkage and Selection." *European Actuarial Journal* 12 (2): 503–28. <https://doi.org/10.1007/s13385-021-00294-x>.
- McInnes, Leland, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. "UMAP: Uniform Manifold Approximation and Projection." *Journal of Open Source Software* 3 (29): 861. <https://doi.org/10.21105/joss.00861>.
- Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. "Distributed Representations of Words and Phrases and Their Compositionality." *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2* (Red Hook, NY, USA), NIPS'13, 3111–19.
- Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. "GloVe: Global Vectors for Word Representation." *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. <https://doi.org/10.3115/v1/d14-1162>.
- Phoenix, James, and Mike Taylor. 2024. *Prompt Engineering for Generative AI: Future-Proof Inputs for Reliable AI Outputs at Scale*. O'Reilly.
- Raiaan, Mohaimenul Azam Khan, Md. Saddam Hossain Mukta, Kaniz Fatema, et al. 2024. "A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges." *IEEE Access* 12: 26839–74. <https://doi.org/10.1109/access.2024.3365742>.
- Salton, Gerard, and Christopher Buckley. 1988. "Term-Weighting Approaches in Automatic Text Retrieval." *Information Processing and Management* 24 (5): 513–23. [https://doi.org/10.1016/0306-4573\(88\)90021-0](https://doi.org/10.1016/0306-4573(88)90021-0).

- Smyth, Gordon K., and Arūnas P. Verbyla. 1999. "Adjusted Likelihood Methods for Modelling Dispersion in Generalized Linear Models." *Environmetrics* 10 (6): 695–709. [https://doi.org/10.1002/\(sici\)1099-095x\(199911/12\)10:6<695::aid-env385>3.0.co;2-m](https://doi.org/10.1002/(sici)1099-095x(199911/12)10:6<695::aid-env385>3.0.co;2-m).
- Tweedie, Fiona J., and R. Harald Baayen. 1998. "How Variable May a Constant Be? Measures of Lexical Richness in Perspective." *Computers and the Humanities* 32 (5): 323–52. <https://doi.org/10.1023/a:1001749303137>.
- Wang, Zichong, Zhibo Chu, Thang Viet Doan, Shiwen Ni, Min Yang, and Wenbin Zhang. 2024. "History, Development, and Principles of Large Language Models: An Introductory Survey." *AI and Ethics*, ahead of print. <https://doi.org/10.1007/s43681-024-00583-7>.
- Wüthrich, Mario V., and Michael Merz. 2023. *Statistical Foundations of Actuarial Learning and Its Applications*. 1st ed. Springer Actuarial. Springer.
- Yang, Lu, and Peng Shi. 2018. "Multiperil Rate Making for Property Insurance Using Longitudinal Data." *Journal of the Royal Statistical Society Series A: Statistics in Society* 182 (2): 647–68. <https://doi.org/10.1111/rssa.12419>.
- Yule, George Udny. 2014. *The Statistical Study of Literary Vocabulary*. 1st paperback. Cambridge University Press.
- Zappa, Diego, Mattia Borrelli, Gian Paolo Clemente, and Nino Savelli. 2021. "Text Mining in Insurance: From Unstructured Data to Meaning." *Variance* 14 (1): 1–15. <https://doi.org/10.66573/001c.28150>.

APPENDIX

A. SUMMARY STATISTICS OF DIFFERENT CATEGORIES
OBTAINED BY LLMS

This appendix presents summary statistics for the risk categories shown in Figures 6, 7, and 8, allowing readers to clearly compare the differences among them.

Table A.1. Summary statistics of log-loss by risk classes for the two-category approach produced by ChatGPT-4o and Llama-3.2-3B on the training set.

	Min	1st Q	Mean	Median	3rd Q	Max
ChatGPT-4o						
NA	4.6151	6.2166	6.9375	6.3863	7.1351	10.8378
Low	0.6931	6.1115	7.1000	7.0604	8.1020	13.9511
Hig	0.6931	7.6468	8.6877	8.5180	9.5894	16.3745
Llama-3.2-3B						
NA	3.9441	5.3954	6.1244	5.9454	6.9218	8.7834
Low	0.6931	6.4126	7.4736	7.4664	8.4454	14.8081
Hig	0.6931	6.7592	7.9244	7.8596	9.0273	16.3745

Table A.2. Summary statistics of log-loss by risk classes for the three-category approach produced by ChatGPT-4o and Llama-3.2-3B on the training set.

	Min	1st Q	Mean	Median	3rd Q	Max
ChatGPT-4o						
NA	5.9181	6.3847	7.2621	6.8839	8.1578	9.0094
Low	0.6931	5.9940	6.8351	6.8143	7.7814	11.8846
Med	0.6931	6.8599	7.9108	7.8836	8.9542	16.3745
Hig	3.1950	7.9915	9.2569	9.0712	10.3379	14.4265
Llama-3.2-3B						
NA	4.6151	6.2166	7.0705	6.7262	7.8496	11.6239
Low	0.6931	6.2166	7.2542	7.2449	8.2430	13.6911
Med	0.6931	6.5096	7.6667	7.6014	8.7495	14.8847
Hig	0.6931	7.1708	8.2355	8.1446	9.1919	16.3745

Table A.3. Summary statistics of log-loss by risk classes for the five-category approach produced by ChatGPT-4o and Llama-3.2-3B on the training set.

	Min	1st Q	Mean	Median	3rd Q	Max
ChatGPT-4o						
NA	4.6151	6.2166	7.1506	6.5539	7.9478	11.8846
Neg	0.6931	5.7189	6.6546	6.4189	7.5178	11.3659
Low	0.6931	6.3107	7.3046	7.2994	8.3091	13.9511
Med	0.6931	7.6831	8.6656	8.5251	9.5798	16.3745
Hig	3.1950	7.8325	8.9163	8.8660	9.7717	14.2670
Cat	5.5114	8.0783	9.7262	9.5100	11.3999	14.4265
Llama-3.2-3B						
NA	1.9459	5.3817	6.5939	6.4701	8.0648	10.0516
Neg	0.6931	6.2166	7.2844	7.1460	8.2631	12.4828
Low	0.6931	6.2166	7.3694	7.3139	8.3960	15.7049
Med	3.5467	6.6932	7.9505	7.8382	9.0859	16.3745
Hig	0.6931	7.1722	8.1403	8.0661	9.1248	14.7840
Cat	3.1950	7.5809	8.5124	8.3990	9.3855	14.8081