APPLICATIONS OF MODERN NLP TECHNIQUES

FOR PREDICTIVE MODELING IN ACTUARIAL SCIENCE

by

Shuzhe Xu

A Dissertation Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Philosophy in Computational Sciences

Middle Tennessee State University

October 2021

Dissertation Committee:

Dr. Don Hong, Co-Chair

Dr. Salvador E. Barbosa, Co-Chair

Dr. Vajira Manathunga

Dr. Zachariah Sinkala

Dr. Qiang Wu

ABSTRACT

In this dissertation, the research focuses on Natural Language Processing (NLP) applications in actuarial science. NLP techniques, as powerful text analytic tools, can help actuaries to exploit the information in textual data. Recently, many NLP techniques have been applied in different research fields, but only a few NLP applications can be found in actuarial science. This dissertation researches NLP techniques in actuarial science and proposes some NLP solutions for actuarial applications.

This dissertation consists of five chapters. The first chapter is an introduction of NLP and some opportunities for its use in actuarial science. The possibilities of traditional actuarial applications incorporating NLP are also discussed. A few NLP applications proposed by actuaries are also introduced as references.

The second chapter is the literature review of relevant NLP techniques. Some basic technologies are introduced such as word embeddings and tokenizations. Also, advanced NLP tools such as Bidirectional Encoder Representation for Transformers (BERT) and related techniques are discussed.

The third chapter is an NLP application based on extended truck warranty data. This chapter develops a BERT-based aggregate loss model with a rescaled 10-value scale severity to predict future losses based on the frequency distribution of claim counts with contracts and severity distribution of claim records. The NLP tool helps to extract information from the textual description in the data, and the extracted values are exploited to predict loss severity.

The fourth chapter is another NLP application for basic truck warranty data.

A data-based portfolio allocation model is proposed to predict losses using the modern portfolio theory (MPT) developed by Nobel Laureate Harry Markowitz in 1952. In this chapter, BERT is applied to improve the accuracy of multi-class classification in the BERT enhanced data-based portfolio allocation model. Also, a technique similar to the one used in chapter 3 is applied to derive a BERT-based severity model for multi-class aggregate loss prediction through a different approach with the BERT enhanced data-based portfolio allocation model.

The last chapter summarizes the described applications. The applications of modern NLP techniques for predictive analytics are practical and promising. However, applications in actuarial science are almost nonexistent. This dissertation demonstrates the possibilities of NLP applications to improve predictive modeling in actuarial science. The NLP techniques can help to gather information from textual descriptions discarded by traditional models. The possible improvements that can be made in future research are also described in this chapter.

DEDICATION

This dissertation is dedicated to my parents, who taught me, encouraged me and supported me throughout my life. Thanks for all your patience, love and unconditional support.

ACKNOWLEDGMENTS

I would like to appreciate all people who taught and supported me during my academic life at MTSU. It is hard to imagine completing this work without those peoples' help. First of all, I would like to thank my supervisor, Dr. Don Hong. He taught me a lot of lessons, not only in the classroom but also in life, through his knowledge and experience. Dr. Hong shared his professional knowledge to help me do my research. He also supported me when I had encountered difficulties. I may not have finished this dissertation without his support and suggestions. I would like to express my special thanks of gratitude to my dissertation co-supervisor Dr. Barbosa who first introduced and inspired me to my current research topic. I am grateful for his valuable suggestions and really enjoyed the time working with him. I would also like to thank my committee members Dr. Vajira Manathunga, Dr. Qiang Wu, and Dr. Zachariah Sinkala for not only teaching me a lot of mathematical and statistical knowledge but also providing me many constructive comments and suggestions for my research. Last but not least, I also am grateful to Dr. John Wallin, the COMS Program director, and all faculty members in the Math Department and the COMS program for their support.

Contents

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

With the development of computer technologies, such as machine learning and artificial intelligence, many researchers and scientists have devoted themselves to finding a way for machines to read and understand human language. To achieve this goal, a subfield of computer science, artificial intelligence, and linguistics, known as Natural Language Processing (NLP), was introduced to process human language. Generally, tasks and challenges in NLP involve text mining, natural language understanding, natural language generation, and speech recognition.

Converting information to text is one of the main methods used by humans to store information that can be found in books, documents, reports, emails, websites, etc. Analyzing this textual data may render helpful information for researchers as well as to the general public. However, textual data is one of the less exploited data types in the machine learning area, especially in actuarial science related applications.

Traditional methods in actuarial science were built based on statistical and mathematical concepts. It is difficult to process textual data directly by traditional methods. Therefore, textual data was usually neglected in actuarial science. However, textual data contains hidden information which is useful when analyzing limited data. Some NLP techniques, such as word embeddings that convert texts into numerical vectors, could help to process textual data. Although these techniques provide a solution to incorporating texts into traditional actuarial models,

the models may not be able to handle the converted values like high dimensional vectors. Unlike general high dimensional data, common dimension reducing techniques are not appropriate for high dimensional vectors extracted from texts. To process this high-dimensional data, machine learning models enhanced with NLP techniques can be a good strategy.

The Bidirectional Encoder Representations from Transformers (BERT) is a machine learning based NLP tool released by Google in late 2018 [12]. BERT obtains new state-of-the-art results on many NLP tasks such as General Language Understanding Evaluation (GLUE), as a pre-trained language representation model that requires fine-tuning to process different NLP downstream tasks. In recent years, many BERT-Based Tools and applications have been developed in different fields [58, 45, 16, 3]. For example, predicting a review helpfulness score as an NLP task is quite challenging. There are many factors that determine the helpfulness score of a review. Some of these are extractive information, such as the overall star rating for each product obtained directly from the data, and others are abstractive information like linguistic features that are more difficult to extract from the review text. In [59], a neural network (NN) based model was developed with BERT features, instead of explanatory variables, and was used to rank the helpfulness of product review data collected by Amazon.com, using the ratio of helpful votes to total votes for each review. This NN-based tool was used to analyze the product review data by incorporating BERT features. The proposed model predicts the helpfulness of customer reviews with a ranking score by analyzing the review text, its star rating, and the product type. The prediction should

help consumers to make a better purchase decision.

There are some examples and opportunities for NLP applications that can be found in the insurance industry [36]:

1. NLP has been used for applications of enhancing marketing strategies by insurance companies. NLP models can help insurance companies to extract the topic from texts or covert comments and feedback into structured data for sentiment analysis.
2. During the policy renewal time period, text mining applications can help underwriters to process a large number of policies to check for compliance and report any changes. These applications provide better tracking for the underwriting process, and may also be helpful for the insured to have more transparency during the renewal period.
3. Analyzing and classifying claim textual data are two common NLP tasks in actuarial analytics. It can reduce errors introduced by humans, and save time. Also, NLP can help to detect insurance fraud through classifiers and predict losses more accurately by incorporating information hidden in textual descriptions to the model.

The textual analysis provides some new options, such as text mining, to process the insurance data through different approaches for better results. Some researchers indicate that text mining has potential in actuarial data analysis [32, 36]. It is important to choose an appropriate strategy to apply NLP to actuarial models. Gee Y. Lee et al. introduced the applications of word embeddings incorporated

with the generalized additive model which are extended from the generalized linear model [32]. In the applications, the authors extracted features from short textual descriptions of insurance claims. In 2020, Antoine Ly et al. mentioned in their paper that some NLP tools could be useful in insurance analytics and introduced several popular NLP techniques such as Word2Vec, Doc2Vec, and BERT [36]. The paper also emphasized that BERT could be a potential tool in insurance data analysis based on its state-of-the-art performance in many NLP applications [12].

This dissertation aims at applying text mining techniques in NLP for predictive modeling in actuarial applications. BERT, as a newly developed NLP technique, is applied to different actuarial models for model enhancement and improvement, to enhance traditional statistical or mathematical models that only use numeric inputs. The dissertation is organized as follows. Chapter 2 is the introduction of relevant NLP techniques. The next chapter describes an application of the BERT-enhanced aggregate loss model on extended truck warranty data. The BERT model in Chapter 3 increases the robustness via the BERT-modified severity using textual data. In Chapter 4, several new approaches to data-based portfolio modeling are discussed. The BERT-enhanced portfolio model improves the multi-class classification accuracy to get a better prediction. Chapter 5 is a summary and conclusion; it also suggests future avenues for work.

CHAPTER 2

PROGRESS ON NLP, BERT-TOOL, AND APPLICATIONS

2.1 Introduction

Text is one the most widely used communication methods in human history. It can be used to transmit and record information as textual data. Text mining, also known as text data mining, is a field of data analytics that derives hidden information from textual data. Textual data can be a document, a paragraph, a sentence, or a single word. Generally, text mining is the process of converting unstructured textual data which is difficult to be used in traditional mathematical or statistical models, to structured data such as numerical values that can be easily used in those traditional models. Traditionally, unstructured textual data is read and transformed into structured data manually by humans. However, these kinds of processes can be expensive when analyzing a large-scale document with thousands of words and sentences. With the evolution of computer technologies, textual data is processed and incorporated into numerical data using machines. The first extraction of information from text through the machine was in 1979 by DeJong [11]. A probabilistic model to calculate the likelihood of each word given its context was introduced by Bahl et al. in 1983 [5]. Semantic analysis, a common task of NLP, was developed later in the 1990s by Grishman and Sundheim [19]. In their approach, the linguistic information was extracted using templates created by humans. However, the manual templates were limited in many cases. With the

growth of textual information due to the Internet, subsequently, researchers introduced algorithms using HTML formatted text [2, 13]. According to [26, 36], two of basic NLP tasks are:

1. Named entity recognition, which is a technique to extract values of common names from text and can extend to the recognition of metric values.
2. Extraction of related information from words and their contexts.

Applying rules is a simple way to do named entity recognition. A rule is applied for a detected pattern. It can be used to extract information from textual data by regular expressions [52]. For example, a regular expression can help to identify a name that follows the word "Mr". Rabiner developed one of the earliest name entity recognition algorithms using probabilities in 1989 [47]. In 1997, Bikel et al. introduced another statistical model to identify the role of each word in a given document [8]. This statistical model is similar to some text mining techniques of modern NLP.

When processing textual data, some common words are not useful for modeling due to their high frequency in documents. Therefore, some techniques are essential to pre-process the textual data before modeling. One of the techniques is to remove some connective words such as "and", some common words like "the", or suffixes like "-ful". Such words or suffixes may not be helpful for understanding the meaning of sentences. Another technique is to split a document into several pieces of sentences or words, a process called tokenization. Many modern

NLP techniques involve tokenization as the essential pre-processing step to analyze the textual data piece by piece. In 2016, the WordPiece tokenization method [56] was developed using machine learning techniques to split words into smaller pieces to retain suffixes. The WordPiece tokenization is trained on a large-scale corpus to tokenize words and sentences efficiently.

After the tokenization, textual data needs to be transformed into numerical values in order to be used by mathematical or statistical models. One technique, word embeddings, converts textual data into vectors with numerical values that can be incorporated with other numerical data [32, 60].

2.2 Word Embeddings

It is difficult to analyze textual information directly by machine learning. In order to understand the language, the converted vectors are required to contain not only the information of each word but also the relations with the corresponding context. In 2003, Bengio et al. developed a large-scale language model based on neural networks. The model processed the text as an unsupervised learning task. The main idea of the model was the transformation of raw words into words vectors, known as word embeddings. Word embeddings are vector representations of words as numerical values through a mapping [54]. The input of the mapping is a set of non-duplicated words from the original text, called a dictionary. The output of the mapping is the vector representations of the corresponding words from the input. After the mapping is implemented, the textual data can be processed

by machine learning models. The fundamental method of word embeddings is bag-of-words (BOW) introduced in 1954 by Harris [21]. The BOW method represents the frequency of each word in each sentence of the given document. This representation is simple, but there is no semantic difference between each word. For example, the word ”bank” has multiple meanings in reality but has not in its embeddings. Generally, there are two types of word embeddings [6]:

1. Models that use word counts or frequencies information are called count-based models.
2. Models based on the context information and usually incorporated with neural networks are named prediction-based models.

2.2.1 Count-based Models

Count-based models process words by collecting word counts or word-context co-occurrence counts in a corpus (a large and structured set of texts produced in a natural communicative setting that can be read by machines). The basic idea of early count-based models is to build count vectors that represent the frequencies of words from the dictionary of a given corpus. This idea is simple, but some common words like “a” or “is” may have a very high frequency in those count vectors. To solve this issue, term weighting-based schemes were introduced by researchers, including term frequency [34], inverse document frequency [27], and term frequency-inverse document frequency [50]. The term frequency, $tf(w, d)$,

of word w in document d can be defined as:

$$tf(w,d) = \frac{\text{numeber of times that } w \text{ appears in } d}{\text{total number of words in } d} \tag{1}$$

The inverse document frequency $idf(w,d)$ can be defined as:

$$idf(w,d) = \log(\frac{\text{number of sentences in } d}{\text{number of sentences in } d \text{ containing } w}) \tag{2}$$

And the term frequency-inverse document frequency $Tf\text{-}Idf$ is obtained by multiplying $tf(w,d)$ and $idf(w,d)$. These term weighting-based approaches can decrease the weights of common words in articles. Count-based models that leverage word-context were introduced by Deerwester et al. in 1990 [10]. These count-based models incorporated word-context by implementing co-occurrence matrices used widely in NLP research. One of the more recent count-based word embedding models is Global Vectors for Word Representation (GloVe), which was released by Pennington et al. in 2014 [43]. GloVe obtains vector representations of words by global word-word co-occurrence statistics, which tabulates how frequently a word co-occurs with another word in a given corpus via an unsupervised learning algorithm. The main idea of the GloVe model is that ratios of word-word co-occurrence probabilities can help to encode some form of meaning.

2.2.2 Prediction-based Models

With the development of machine learning algorithms, prediction-based embedding models are more popular. The first prediction-based model was introduced with artificial neural networks in 2003 [7]. The model applied word embeddings in its first layer of artificial neural networks. An artificial neural network is an information gathering and processing model which has a similar structure to a biological neural system. Therefore, artificial neural networks can learn through examples. Generally, artificial neural networks have multiple layers to transmit and process the information of inputs, including an input layer, several hidden layers, and an output layer. In 2010, Mikolov et al. used a Recurrent Neural Network (RNN) as an optimized way to train a language model. RNNs are a standard type of neural network that can be extended over time and is designed to process sequences such as texts, through cycles in its structure which can pass the historical information from the sequences. The general idea of using RNNs is to store and pass information of earlier words in the text in the hidden layers, which can help the model to analyze texts among long sentences. Due to their structures, RNNs have limitations when dealing with long sequences of words. One of the main tasks of prediction-based models is to speed up and improve the accuracy of training processes. Mikolov et al. proposed two models for training embeddings called the continuous skip-gram and bag-of-words (CBOW) models in 2013 known as Word2Vec [38, 40]. The two models defined a context window C of size k on anywhere of a sentence of size n, where $k \leq n$. The skip-gram model predicts the surrounding context from the central word in C. The CBOW

model predicts the central word based on its context in C. The two models used neural networks in their training step and built relations between words and corresponding contexts [39]. In 2014, Mikolov and Le extended the Word2Vec model to the Doc2Vec model by adding a new embedding that can map a paragraph to a vector [30]. The vector is called a paragraph vector and represents the information, such as the topic, of the paragraph from its context.

2.3 Transformers

There are some problems when using Word2Vec models. One of the main problems is that the relations of words within the model are purely statistical and contain only spatial proximity. To obtain more complex relations, many traditional language models are based on neural networks, such as RNNs and convolutional neural networks (CNNs), as their encoder-decoder mechanism. A CNN is one type of neural network which can extend across space through shared weights [1]. However, RNNs have difficulties in handling long sequences and CNNs can be time-consuming when processing large scale of textual data. In order to improve efficiency, Google researchers proposed the transformer model based on the attention mechanism [55] in 2017. The transformer model can function like a human brain by giving attention only to the most important information. The model is designed to reduce the cost of the training stage and increase the accuracy of predictions without either traditional RNN or CNN structures. Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based NLP tool re-

leased by Google in late 2018 [12]. Many researchers have shown that BERT can process product review data with high accuracy, working alongside reading comprehension and aspect-based sentiment analysis [45, 16]. Recently, many NLP task-related types of research are running with BERT-based models [44, 23, 3].

2.3.1 BERT

BERT is a language model trained on a large corpus that allows fine-tuning for specific tasks. It was developed by Jacob et al. in 2018. It enables outstanding results on many NLP tasks such as General Language Understanding Evaluation (GLUE), Natural Language Inference (NLI), Corpus of Language Acceptability (CoLA), etc [12]. The general idea of BERT is to pre-train the language model by using large-scale corpora in a transformer model [55]. Pre-trained representations can be either contextual or context-free. The contextual representation can be bidirectional and unidirectional. Context-free models such as GloVe and word2vec create a single word embedding representation for each word from the document, regardless of its context. Contextual models, on the other hand, create a representation of each word based on the context. The bidirectional representation uses both the left and the right context for each word, and the unidirectional representation uses only the left or the right context. The BERT model is trained bidirectionally through multiple layers using transformer neural networks. It learns the words and their contexts during pre-training. Then, the BERT model can be used to learn some specific details for the given textual task during fine-tuning. The model tokenizes its input using WordPiece [56] as its word embedding.

The WordPiece tokenization is applied before pre-training and fine-tuning for the model:

1. A special token $[CLS]$ is added before each sequence.
2. Another special token $[SEP]$ is used for sentence separation.
3. All inputs are tokenized based on a large vocabulary using WordPiece tokenization [56] as token embeddings. The WordPiece tokenization can split a sentence or a word into small pieces. The sentences or the words with similar pieces have a similar meaning or a strong relation.
4. In order to represent that a word belongs to a specific sentence in the text, segment embeddings are applied to every token.
5. Positional embeddings are also applied to show the location of the word in the given text.

During pre-training, BERT is trained using a large plain text corpus such as Wikipedia and is structured by combining several encoders extracted from transformers. The bidirectional encoder architecture is pre-trained with two main tasks [12]:

1. Masked Language Modeling, is a process that replaces about 15% of the original words by $[MASK]$ tokens. The model can then be trained to predict the masked words.
2. Next Sentence Prediction, as a classification problem, is used to train a model

to predict whether one sentence follows another, given two sentences.

The goal of pre-training is to learn the language by minimizing the loss functions for these two tasks. The pre-trained BERT can go through fine-tuning to solve downstream tasks using its learned language. The input and output of fine-tuning are specific to the downstream task. In this dissertation, the inputs are textual descriptions from the data sets and the outputs are the extracted vector representations that can be exploited using neural network models.

BERT has showed its adaptivity for multiple end tasks by optimizing different fine-tuning processes [44, 23]. BERT is useful for multiple downstream tasks without changing its pre-trained language model. Compared to other language methods, BERT has better prediction accuracy in many different downstream tasks, especially those that feature extraction-related tasks. BERT has two primary model sizes with different parameters:

BERT$_{Base}$: 12 layers, 768 hidden dimensions, 12 self-attention heads and 110 million total parameters.
BERT$_{Large}$: 24 layers, 1024 hidden dimensions, 16 self-attention heads and 340 million total parameters.

In 2020, 24 smaller BERT models that referenced were released [53]. The releases of smaller BERT models were intended for some environments with limited computational resources.

BERT is a powerful and popular NLP tool that is already used in different tasks by many NLP researchers. For example, Ly et al. introduced the NLP techniques and emphasized that BERT is a potential tool for insurance data analysis in 2020 [36]. In this dissertation, the $\mathbf{BERT}_{Base}$ model will be used in extended truck warranty and basic truck warranty actuarial science studies.

CHAPTER 3

BERT-BASED LOSS SEVERITY PREDICTION MODEL FOR EXTENDED WARRANTY DATA ANALYSIS

3.1 Introduction

Traditional probability and statistical based models on aggregate claim data are widely used in risk assessment, loss reserving, and rate-making [48, 15, 51, 18]. However, these aggregated models may neglect information such as individual claim details [57]. Nowadays, many actuaries are seeking new methods through machine learning techniques to process claim data [57, 29]. By applying machine learning based methods, the information not considered in traditional models can be used in processing [57, 29, 46, 42]. Machine learning based methods such as Decision Trees and Neural Networks [57, 46], are used to process large-scale structured information. In the last two decades, many machine learning models have been developed by actuaries to process data in insurance analytics [57, 42].

Machine learning tools can help actuaries achieve many goals, including fraud detection and predictive modeling. By applying these machine learning based methods, complicated models that incorporate more information such as dependencies can be considered. For example, traditional aggregated models for determining premium rates, also called rate-making, are based on the independence of loss frequency and severity. These models are not reliable without the indepen-

dence assumption. Actuaries developed a generalized linear mixed model which incorporates the dependency of frequency and severity [15, 20, 17, 25]. Machine learning techniques are implemented not only for loss reserving and rate-making but also for multiple actuarial tasks such as survival analysis [42, 35, 28]. Thus, machine learning based methods is becoming more popular in actuarial analytics [29, 33].

In insurance analytics, text information was formerly utilized only for descriptive purposes. It was difficult to exploit text directly by traditional methods without pre-processing. However, these textual data may contain hidden information which can contribute to better actuarial analysis for loss reserving and rate-making. By using word embeddings, actuaries can apply hidden textual information in modeling. There are very few applications in insurance analytics that use word embeddings. G.Y Lee et al. developed a generalized additive regression model for the Wisconsin Local Government Property Insurance Fund data, to show how to improve insurance claims management and risk mitigation procedures in 2019 [32]. Just like the application by G.Y Lee et al., NLP techniques have demonstrated reliability and accuracy in other research areas [41, 13, 56]. In 2019, we applied a Bidirectional Encoder Representations from Transformers (BERT) model for a predictive task dealing with product review helpfulness. Thus, we believe these NLP techniques can also be implemented in insurance analytics.

BERT is a language representation model that can be fine-tuned for various tasks. As of 2020, it obtains new state-of-the-art results on many NLP tasks such

as General Language Understanding Evaluation (GLUE), Natural Language Inference (NLI), Corpus of Language Acceptability (CoLA), etc [12]. In this chapter, we describe a BERT-based model for loss severity prediction by incorporating textual information found in claims and repair records of truck warranties.

3.2 Description of Data and Data Pre-processing

In this chapter, a general strategy is described to process raw textual data. The strategy provides a solution for developing a traditional actuarial model incorporating textual information processed by BERT.

The dataset in this chapter was collected from extended warranty policies with coverage of 4 years. The dataset was extracted from a raw data set and was then expanded via simulation for actuarial modeling. The dataset contains each truck's warranty contract policy information (contract table), and claim details created over a 5-year period for warranted trucks (claim table). The data in the claim table includes textual descriptions that explain the causes of truck failures. Attributes in the contract table and the claim table are listed in Table 1 and Table 2.

There are 7,557 claim records contained in the dataset. In this study, the data was split into training and testing sets using Scikit Learn, a machine learning tool in Python for data splitting, with 60% for training, 40% for testing, and no overlapping data among the two datasets.

Furthermore, some addtional pre-processing was added to the claim table for machine learning. Claim records with missing values were removed, and extreme

cases with abnormal values, for example, multiple repairs in one claim, were also removed.

Some attributes of the tables are explained as below:

Attributes	Description
SERIAL NO.	Each truck has a unique serial number as its identification
MODEL	There are several different models for trucks
BUILD_DT	Truck-built date
PLCY_NM	Policy name
WARR_START	Warranty started date
WARR_END	Warranty ended date
CVRG_TYPE	Coverage type

Table 1: Attributes in Contract Table

Attributes	Description
SERIAL No., MODEL, BUILD_DT, and CVRG_TYPE	as same as Contract table
CLAIM	Each claim record has a unique number
FAILDAT	Truck failed date
REPADAT	Truck repaired date
DEFECTDESC	Short description of cause of failures (No more than 5 words)
GROUPDESC	Group of troubles and failures
COMPLAINT	Complaint from customers
CAUSE	Detailed description of cause of troubles and failures
CORRECTION	Detailed description of repair
LABORPD	Paid amount for labor
PARTSPD	Paid amount for broken parts
OTHERPD	Paid amount for other expenses
TOTALPD	Total paid amount

Table 2: Attributes in Claim Table

The COMPLAINT, CAUSE, and CORRECTION attributes are textual descriptions with long sentences or paragraphs.

In the raw data, limited useful information is available for building traditional prediction models. Many of the non-textual information seem to have no direct relation to the total loss amount. Therefore, NLP techniques are used to extract features to build the prediction model.

3.3 BERT-based Aggregate Loss Model

The aggregate loss model, which was built to predict the future loss based on the frequency distribution of claim counts with contracts and severity distribution of claim records, is described in this section.

3.3.1 Aggregate Loss Model

In insurance industry data, N usually represents the number of losses to the insured who is a person or entity buys insurance. X represents the claim payments of the insurer (an insurer is an entity that provides insurance). X can also represent the claim payment of the reinsurer, an entity issuing the reinsurance policy (Reinsurance is insurance that an insurance company buys from another insurance company to reduce the risk of claims), or the deductibles paid by the insured. In this chapter, N refers to the claim count random variable. The distribution of N is the claim count distribution, known as the frequency distribution. X refers to the individual loss size random variable and its distribution, known as the severity

distribution.

The aggregate loss model is used to predict total loss payments by insurance companies. The model is based on the number of historical claims and the amounts of each claim as its random variables. Generally, there are two ways to model the number of total losses on all claims that occurred in a specific time period with a set of insurance contracts.

The first is the collective risk model that is usually applied to independent and identically distributed (i.i.d.) observations. The collective risk model represents the aggregate losses by a sum S of individual payment amounts X_1, X_2, X_3,...,X_N with a claim count random variable N:

$$S = X_1 + X_2 + X_3 + \cdots + X_N, N = 0, 1, 2, ..., \tag{3}$$

In this equation, $S = 0$ when $N = 0$. The independence assumptions of this model are:

1. X are independent and identically distributed random variables conditional on $N = n$.
2. The common distribution of X does not depend on n, conditional on $N = n$.
3. The distribution of N is independent of the values X_1, X_2, X_3,...,X_N.

The second model is the individual risk model. This model represents the aggregate loss amount as a sum S of n insurance contracts. The loss amounts of

contracts are random variables X, which are assumed to be independent but not necessarily identically distributed. The distribution of these random variables has a probability mass at zero in general, which presents the probability of no loss occurring on that contract. The individual risk model is used to calculate the total losses of a set of insurance contracts. The model becomes a special case of the collective risk model when X are identically distributed with the distribution of N being the degenerate distribution of $Pr(N = n) = 1$.

The distribution of total losses S is determined by the distribution of random variable N and the distribution of random variable X. Usually, the frequency and the severity of claims are modeled separately by using this approach. An alternative approach of total loss S is to obtain the distribution of S simply by information that directly comes from S. Modeling S separately by using the distribution of N and the distribution of X has its applications, such as changing individual deductibles. Policy limits can be implemented easily by modifying the details of the severity distribution. The impact on claims frequency of changing deductibles can be better observed using this approach.

Let S denote aggregate losses associated with a set of observed claims X_1, X_2, X_3,...,X_N with the number of claims N. The model is designed in three steps:

1. Build a model for the distribution of N based on the data in the contract table as the frequency model.
2. Develop a model for the distribution of X based on the data from the claim table as the severity model.

3. Estimate the expected value of S by using the frequency model and severity model.

The total expected loss amount $\mathbb{E}(\mathrm{S})$ is measured by the multiplication of the expected frequency $\mathbb{E}(\mathrm{N})$ and the expected severity $\mathbb{E}(\mathrm{X})$.

$$\mathbb{E}(\mathrm{S}) = \mathbb{E}(\mathrm{N})\,\mathbb{E}(\mathrm{X}) \tag{4}$$

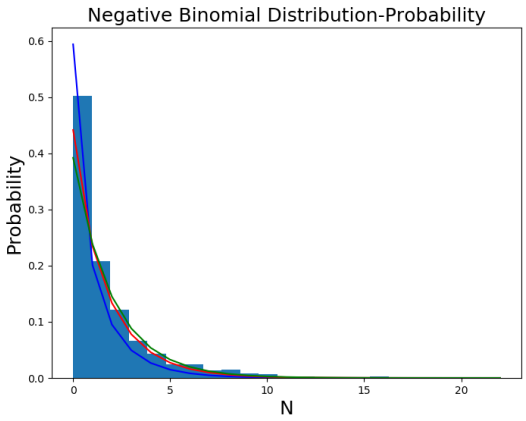
The frequency distribution of the contract data is determined by fitting a negative binomial distribution based on the shape of the histogram (Figure 1). The frequency is listed in Table 3. The sample mean is 1.55166 and the sample variance is 6.19509. A Poisson distribution is not considered, since there is a large difference between mean and variance. The probability mass function of the negative binomial distribution is:

$$\Pr(\mathrm{N} = \mathrm{n}) = (\binom{\mathrm{n} + \mathrm{r} - 1}{\mathrm{r} - 1})\mathrm{p}^{\mathrm{r}}(1 - \mathrm{p})^{\mathrm{n}}, \qquad \text{for} \quad \mathrm{n} \geq 0, 0 < \mathrm{p} \leq 1 \tag{5}$$

Where r is the number of successes, n is the number of failures, and p is the probability of success. The fitted distribution is showed in Figure 1. The red curve is the optimization with smallest errors when $r = 0.872801$ and $p = 0.391902$ (Green: $r = 1, p = 0.391902$;Blue: $r = 0.556025, p = 0.391902$). The parameters r and p are estimated using Maximum Likelihood Estimation (MLE).

Number of claims for each contract	Frequency
0	1233
1	509
2	299
3	162
4	105
5	59
6	59
7	32
8	38
9	21
10	16
11	7
12	8
13	3
14	4
15	0
16	6
17	1
18	1
19	0
20	1
21	0
22	1

Table 3: Frequency Distribution



(a) Frequency

Figure 1: Fitted Negative Binomial Distribution for Frequency by Three Set of Parameters

In this study, BERT is applied to process textual data via a fine-tuning step. In order to incorporate BERT appropriately, one approach is to predict the loss amount for each test data directly through a regression model. We previously developed a model to automatically assign the review helpfulness score using the information extracted by BERT from review comments [59]. Similar to the scoring model, BERT can help adjust losses by predictions using regression. Alternatively, another different approach provided in this chapter is to rescale severity levels, based on the loss amount, to fit a severity distribution by its histogram for the aggregate loss model. In Table 4, the severity is mapped to a 10-value scale with a likelihood of gamma distribution on its histogram. By this pre-processing, the BERT model can be utilized to map the rescaled severity levels to fit a new gamma distribution. The definition of the severity level follows several rules:

1. An appropriate number of divided subsets is required for the severity distribution. A smaller number may improve the accuracy of the BERT model to predict the severity but will also increase errors from the fitted distribution since a distribution generated by a few points may be too rough for fitting. On the other hand, a larger number can result in a better fitting but may increase the predictive errors. Hence, a 10 to 15 value scale was deemed reasonable. In this chapter, a 10 value scale is selected.

2. Different widths for each severity level may result in different shapes of histograms. The widths are modified intentionally to generate a gamma-like histogram for better fitting in this chapter. Other values for different distributions

can also be considered.

3. For each divided subset of severity, the losses are assumed to be distributed uniformly for a simple calculation of the average. The average severity for each subset can be used to measure the expected severity for the 10-value scale severity system.

Loss amount(X)	Severity level(L)
$0 \leq X < 100$	1
$100 \leq X < 200$	2
$200 \leq X < 300$	3
$300 \leq X < 500$	4
$500 \leq X < 700$	5
$700 \leq X < 1200$	6
$1200 \leq X < 1700$	7
$1700 \leq X < 2600$	8
$2600 \leq X < 3600$	9
$3600 \leq X$	10

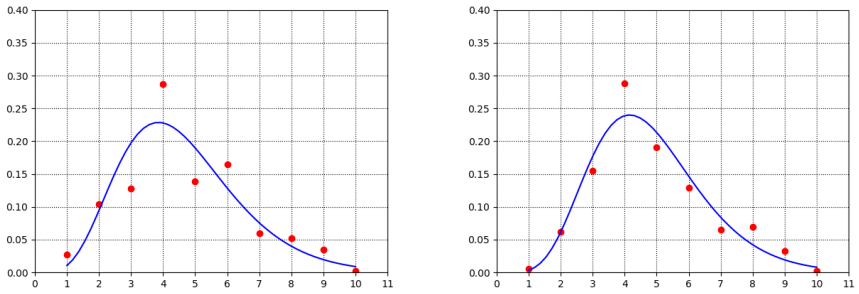
Table 4: 10-Value Scale Severity Level

The gamma distribution with parameters alpha and beta is given in Equation (6):

$$f(x;\alpha,\beta) = \frac{\beta^{\alpha} x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)} \tag{6}$$

Where α is the shape parameter and β is the scale parameter.

In the severity model, the BERT-modified severity levels are generated from the textual description data from the claim table, using the BERT NLP tool. The



(a) Severity (b) BERT-modified

Figure 2: Fitted Gamma Distribution for Severity

BERT model helps to extract hidden information from text via its context, and converts it by modifying the original severity level to the BERT-modified severity level by feeding it through a neural network. The process of generating BERT-modified severity levels is the focus of this chapter.

3.3.2 BERT-based Severity Level Model

In the previous section, the BERT-modified severity level was introduced. The proposed model uses BERT to extract features from the raw textual data. The dimension (length of the converted vector) of extracted features is high and may not be useful without BERT since the extracted features are difficult to read in raw form. Thus, BERT-based neural networks are built to apply the extracted features as the strategy in this study.

During the fine-tuning process, a neural network is formed by incorporating the BERT pre-trained model with one additional output layer to generate the modified severity by regression. The input to the neural network model is a vector that

contains the extracted BERT features from the CAUSE column, the DEFECTDESC column, and the GROUPDESC column. The output of the neural networks model is the modified 10-value scale severity. The model is built as:

$$\hat{L} = f(w_1^T x_1 + w_2^T x_2 + w_3^T x_3 + b), \tag{7}$$

Where $\hat{L}$ is the modified severity, f is the activation function, w_1, w_2 and w_3 are vectors of weight of specific terms x_1, x_2 and x_3. The length of w_1, w_2 and w_3 are determined by the length of the corresponding input x_1, x_2 and x_3. In this test, x_1 are the extracted features by BERT from the CAUSE column, x_2 are the BERT features from the DEFECTDESC column, and x_3 are BERT features from the GROUPDESC column. b is a bias vector. The BERT-modified severity is generated through the neural network trained on the 10-value scale severity level. The input of the NN are elements in x. For each element in x, it multiples the corresponding weight in the hidden layer. The bias b is added to this. The output of the NN is the summation of outputs that come from the activation function.

The loss function of the model is measured by mean squared error (MSE) between the 10-value scale severity level L_i with the BERT-modified 10-value scale severity level $\hat{L}_i$:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(L_i - \hat{L}_i)^2, \tag{8}$$

The goal of fine-tuning is to minimize the loss function. The procedure of the

BERT based severity level model is listed as follows:

Model 1 BERT-Based Severity Level Model

Require: Textual data x_{i1}, x_{i2}, x_{i3} and x_{j1}, x_{j2}, x_{j3}, original severity X_i and X_j
Ensure: Parameters of gamma distribution $\hat{\alpha}, \hat{\beta}$ and α, β

Split data into training set with size of m, and testing set with size of n
for $j = 1$ to m **do**
    $L_j \leftarrow 10\text{VALUESCALE}(X_j)$
end for
for $j = 1$ to m **do**
    Train BERTSEVERITY model using x_{j1}, x_{j2}, x_{j3} as its input and L_j as its output
end for
for $i = 1$ to n **do**
    $L_i \leftarrow 10\text{VALUESCALE}(X_i)$
end for
for $i = 1$ to n **do**
    $\hat{L}_i \leftarrow \text{BERTSEVERITY}(x_{i1}, x_{i2}, x_{i3})$
end for
$\alpha, \beta \leftarrow \text{MOMENTMATCH}(L_i)$
$\hat{\alpha}, \hat{\beta} \leftarrow \text{MOMENTMATCH}(\hat{L}_i)$

3.3.3 Fine-tuning

Several different sets of hyper-parameters were tested to archive the best result. The tested values of hyper-parameters are listed in Table 5.

The parameter max_seq_length specifics the lengths of input tokens are used to train the model. The $train\_batch\_size$ is the number of samples processed before the model is updated. The adjusted value of max_seq_length was 192 and $train\_batch\_size$ was 16.

	Hyper parameter
max_seq_length	96, 128, 144, 192, 256
$train\_batch\_size$	8, 16, 32
$Modeltype$	$BERT_{Base}$
$Optimizer$	Adam

Table 5: Fine-tuning hyper parameters

3.4 Results

In the aggregate loss model, the BERT-based model is applied in order to improve the performance of predicting results of severity. In this study, an examination is designed to test the robustness by comparing the standard deviation of means and variances from multiple tests.

3.4.1 Validation and Test

To examine the robustness of the BERT-modified severity level model, a group of tests was designed. We created 10 subsets by randomly selecting 90% of the original data. Each subset contains 90% records with both original severity levels and BERT-modified severity levels.

For each subset, we fitted two gamma distributions with parameters and calculated the mean α/β and variance α/β^2 of the distributions for both the original severities and the BERT-modified severities. The average of the means, variances, and standard deviations of the 10 tests were then calculated. A example of fitted distribution is showed in Figure 2 ($\alpha = 6.0691, \beta = 1.3117$ for defined sever-

ity level; $\alpha = 7.4716, \beta = 1.5536$ for BERT-modified severity). More detailed severities for 10 tests is listed in Table A.1 and Table A.2. The graphs of fitted distribution are shown in Figure 3-5.

	Mean (Pre-defined)	Mean (BERT-modified)	Variance (Pre-defined)	Variance (BERT-modified)
test1	4.619321	4.800765	3.529984	2.400382
test2	4.625349	4.810426	3.518216	2.405213
test3	4.627702	4.807969	3.527620	2.403985
test4	4.627555	4.803411	3.512953	2.401706
test5	4.623438	4.807381	3.520162	2.403691
test6	4.620791	4.801206	3.528455	2.400603
test7	4.615792	4.798559	3.520227	2.399280
test8	4.638583	4.818409	3.534133	2.409205
test9	4.631084	4.812233	3.522627	2.406117
test10	4.621085	4.804293	3.519267	2.402147
average	4.625070	4.806465	3.523364	2.403233
std.dev.	0.006565	0.006068	0.006465	0.003034

Table 6: Average and Standard Deviation of 10 Tests

The results in Table 6 show that the standard deviation of the mean value of BERT-modified severity is close to the value of the original severity level, and the standard deviation of BERT-modified severity is 7.57% lower. The standard deviation of variances is slightly different, the standard deviation of BERT-modified severity is 53.07% smaller. The difference in the standard deviation of the variances shows the BERT-modified severity is more stable when data changes.

In Table 7, the value of $\mathbb{E}(\mathrm{S})$ is the predicted loss of a single contract. The expected severity $\mathbb{E}(\mathrm{X})$ is calculated using the 10-value scale under uniformly

$\mathbb{E}(\mathrm{N})$	1.354287	Var(N)	3.4556747
$\mathbb{E}(\mathrm{X})$	631.1294	Var(X)	144498.6
$\mathbb{E}(\mathrm{X})$(BERT)	678.2765	Var(X)(BERT)	97673.2
$\mathbb{E}(\mathrm{S})$	854.7303	Var(S)	1572172
$\mathbb{E}(\mathrm{S})$(BERT)	918.581	Var(S)(BERT)	1722091

Table 7: Predictive Results

distributed assumption and the fitted gamma distribution. For example, if the mean of the fitted gamma distribution is 4.655647, then the expected severity $\mathbb{E}(\mathrm{X})$ can be determined using the equation: $(4.655647 - 4) * (700 - 500) + 500 = 631.1294$. The total predicted loss amount for 2,565 contracts in the test set is 2,192,383 by original severity, and 2,356,160 by BERT-modified severity. Compared to the real loss amount 2,437,637 from the data, the predicted losses using BERT enhanced model are closer to the true losses.

Overall, the test results from the data show that the aggregate loss model enhanced by BERT results in a better prediction. From the comparison of their standard deviations, the BERT enhanced aggregate loss model has better stability (standard deviation) than the original aggregate loss model on the different data sets (Table 6). The result supports the strategy proposed in this study applying the BERT NLP tool to a traditional actuarial model.

CHAPTER 4

BERT-Based Model Enhancement on Portfolio Allocation Method in Basic Warranty Data Study

4.1 Introduction

In a recent MS Thesis research project on basic warranty data study, completed by C.L. Zhang [61], the loss payment data was divided into groups and multi-class logistic regression classification algorithms were applied to determine the probabilities to achieve a more precise prediction on the total loss for the policy. Along this research direction, we first formulate the model as the data-based portfolio allocation model, then propose a BERT-based model enhancement, using textual information in the claims and repair records, to achieve better prediction accuracy and stability. The model enhancement involves a neural network classifier to classify the claim payment portfolios. The BERT-based severity model discussed in the previous Chapter is also applied to portfolio distribution modeling. Portfolio variances are calculated for further analysis of the model.

Traditionally, portfolio allocation is an investing strategy that helps determine what percentage of assets should be in diversified investments to optimize the risk-return trade-off. Modern portfolio theory (MPT) was developed by Nobel Laureate Harry Markowitz [37] aiming to minimize market risk and maximize investors' returns by creating efficient weighted percentages for portfolios, based on the calculation of their variances and correlations to result in lower total variabil-

ity. In [22], advanced data science methods were applied to improve MPT and to efficiently compute a solution of portfolio criterion. Leveraging this emulator, we first introduce a data-based portfolio model for the product's manufacturer basic warranty data study, then apply a recently developed natural language processing (NLP) tool called Bidirectional Encoder Representations from Transformer (BERT) to incorporate textural information for the model enhancement.

Many traditional models for loss prediction and reserve evaluation may neglect information such as individual claims' details [57]. In products' manufacturer basic warranty data, the payment records can usually be divided into different groups, based on payment types. For example, under a basic warranty for moving trucks, the payment dataset consists of information including the truck model, number of claims, each claim's loss amount (usually the insurance payment), loss descriptions, payment types, and payment amounts, among others. During the insured period, a basic truck warranty covers most losses of a new truck, within a set time period from the truck's sell date. Each payment to truck warranty claim can be divided into three groups: labor payments, parts payments, and other payments (such as coverage of towing services). Analogous to the risk probability and returns in investments, loss payments can be classified according to the payment nature and corresponding severity level. A data-based portfolio predictive model that efficiently computes weighted percentages associated with loss severity values of corresponding classified loss payments can be important and very practical when determining basic warranty policies.

The first task of this chapter is to formulate a data-based portfolio predictive

model through the basic warranty data study. First, the multiple-class logistic regression model was used to predict the probabilities that a given observation belongs to in its corresponding portfolio, where the parameters of the model were fitted by the training data. Then, for each portfolio, the mean payment amount can be calculated or estimated using training data. The predicted total payment is calculated by using the total expectation formula. This method is easy to implement and yields satisfactory prediction accuracies.

With the goal of incorporating textual information to enhance the data-based portfolio model, the second task in this chapter is to develop a BERT-based classification algorithm to achieve efficient computing of the probabilities associated with portfolio allocations of loss data.

The remainder of this chapter is organized as follows. Section 4.2 gives a description of the data and data-based portfolio modeling. Detailed procedures of the model with the logistic classification algorithm for the model are presented in the following section. In Section 4.4, the BERT-based classification for model enhancement is discussed. Also, BERT-based severity modeling method is applied to estimate the portfolio loss distributions for a possible even more precise prediction of aggregate losses. Portfolio variances are calculated for further analysis and reference. Conclusions and final remarks are included in the final section.

4.2 Description of Data and Data Pre-processing

4.2.1 Data Explanation

The dataset in this chapter is based on trucks under a 2-year basic warranty policy that was extracted from the same raw data set as that used in the previous chapter and was then expanded via simulation for actuarial modeling. The dataset, collected over a 5-year period, contains each truck's warranty contract policy information and claims details for warranted trucks. In the claim table, the total loss amount for each record contains 3 categories: parts-payments, labor-payments, and other-payments. In many cases, the amount of labor-payments in a claim record are relatively small compared to the parts-payments. The category of other-payments consists the payments that do not belong to parts or labor, e.g. towing services. The data include textual descriptions that explain the causes of truck failures. Attributes in the claim table are listed in Table 8.

The dataset is composed of 11,742 claim records. Similarly, the data was split as what we did it chapter 3 by Scikit Learn package in Python for data splitting with 60% for training, 40% for testing and no overlapping data among the two datasets.

There are several features of a basic truck warranty policy which make it different from a general auto insurance policy, with challenges requiring the applications of different techniques for rate-making. Due to the newness of trucks, the maintenance cost is the most significant portion under a basic warranty. Also, for the basic warranty loss severity consideration, identifying the risk stemming from

Explanatory Variable	Description
Model	Alphanumeric values of truck models that categorized in different models.
Defect Code	Numeric value as the cause of the failure, categorized in 73 values.
Cause Description	Textual data as the description of the failure.
Hour Meter	Numeric value of the truck driven time.
Deal to Fail	Numeric value that shows the time difference between the shipped date and the failure date.
Labor Payment	Numeric value of labor payment.
Parts Payment	Numeric value of parts payment.
Other Payment	Numeric value of other payment.
Total Payment	Numeric value of total payment.

Table 8: Explanatory Variables in Simulated Data

the driver is not as important as it is under general auto insurance policies.

Compared to the truck extended warranty data, the truck basic warranty data have some characteristics, such as shorter time periods, larger sample sizes, and more similarity on trucks' conditions. A truck basic warranty usually has up to 2 years of coverage, and all of the trucks in the basic warranty data were brand new when purchased. The data size of the basic warranty is obviously larger than the extended warranty for the same model of trucks.

In the truck basic warranty data, the records of payments can be divided into four different groups based on the different types of payments [61]: labor-dominant (LD), parts-dominant (PD), other-dominant (OD), and none-dominant (ND) payments. There are several interesting phenomena in the data observa-

tions of PD-payments. First, the total payment amount will always be relatively large. Second, the main source of total payments is from the cost of parts. Third, the parts-payment not only has the largest mean value among other types of payments, but it also has the largest standard deviation. The initial thresholds to classify the payments are 75%, i.e., the ratio of parts-payment to total payment is 75% or above defines the PD-payment group. The classes of LD-, OD-, and ND-payments are selected similarly. The class of PD-payments always has a high mean value and a large variance. This feature is similar to the investment scenarios with a high average return rate and a high volatility in modern portfolio theory (MPT). Therefore, we would like to classify the loss claims of different portfolios and apply the portfolio allocation strategy to predict aggregate loss with small variance. For this purpose, the multi-nomial classification will play an important role in determining corresponding probabilities, especially with portfolios having both a high mean value and a large variance.

4.2.2 Data Exploratory Analysis

Table 9 shows the summary (from the original work), for the 5 data classes with mean value information regarding the TOTAL PAYMENT from the entire training dataset and its corresponding four subsets, namely labor-dominant (LD), parts-dominant (PD), other-dominant (OD), and none-dominant (ND) groups [61].

In the next section, the other predictors in the multi-nomial logistic classification algorithm include explanatory variables from DEAL TO FAIL, DEFECT CODE and SALES MODEL.

Dataset	Min.	Q1	Median	Mean	Q3	Max.
ND	18.23	228.79	293.76	353.87	416.18	3153.95
LD	8.86	72.00	84.00	105.69	99.00	1238.08
PD	2.02	706.10	955.31	1085.73	1311.56	6849.83
OD	32.87	147.81	173.07	216.04	205.81	1261.42
Total	2.02	171.79	293.69	495.72	646.57	6849.83

Table 9: Quantiles and Mean Values of TOTAL PAYMENT in 5 Data Classes From the MS Student's Thesis [C.L. Zhang, 2021]

4.3 Data-Based Portfolio Model

As we have seen from the previous discussion, it is essential to compute the probabilities corresponding to each portfolio of the payment data in order to predict the aggregate loss payment under a basic warranty policy. In this section, we outline the portfolio allocation model with a multi-nomial classification algorithm.

4.3.1 Multiple-Class Classification

A classification problem identifies which group one or more observation belong to, given that there are two or more groups and each observation is from exactly one of these groups. In statistics, the terminology of a so-called group is class [24]. Multi-class classification has been widely used in handwriting and speech recognition [31, 41, 14], text classification, and information retrieval [4, 9]. A typical k-classes multi-class classification problem can be defined as follows. For a given data D in the form of $\{x_1, x_2, ..., x_n, y_1, y_2, ..., y_n\}$, and class labels

$L = \{l_1, l_2, ..., l_k\}$ where $l_i \neq l_j$ for any $i \neq j$. $x_1, x_2, ..., x_n$ are the inputs, and $y_1, y_2, ..., y_n \in L$ are the output of labels.

There are several algorithms to perform classification. The outputs of a logistic regression are probabilities that each observation belongs to corresponding classes. The outputs of some other algorithms, such as linear discriminant analysis (LDA), provide only the classes each observation belongs to. In this study for portfolio probability calculation, multiple-class logistic regression is implemented.

4.3.2 Data-Based Portfolio Allocation Model

In C.L. Zhang's thesis, a multiple-class logistic regression model is applied to predict the probabilities that a given observation belongs to one of the corresponding groups (LD, PD, OD, and ND) [61]. The parameters are estimated by the training dataset. To predict the mean payment amount for each group, some advanced algorithms can be applied based on data distributions. We can simply use the sample mean payment for each group and find the predicted total losses using the weighted average formula:

$$\mathbb{E}(T) = \sum_{i=0}^{n} \mathbb{E}(X|Y_i) Pr(Y_i) \tag{9}$$

Where $\mathbb{E}(T)$ is the expected total losses. $\mathbb{E}(X|Y_i), \text{for } i = 0, 1, ..., n$ is the expected loss under Y_i class with corresponding probability $Pr(Y_i)$. The probability $Pr(Y_i), \text{fori} = 0, 1, ..., \text{n}$ can be determined by the classifier, and the conditional

expected loss for each ith class $\mathbb{E}(X|Y_i)$ can be calculated in various ways. In this chapter, the term $\mathbb{E}(X|Y_i)$ is simply estimated using sample mean.

4.3.3 Analysis Results

By using the multiple-class logistic regression/classification algorithm, we can classify and estimate the probability of each payment in its corresponding group. The classification results are shown in Table 12 and Table 16.

The prediction accuracy, the sum of true positive ratios, is 0.5284. Since the losses from the PD group always have a higher mean value and a larger variance, we pay more attention to this group. From the Table 12, we have $\frac{825}{1267} = 0.6511$ and see that more than 65% of claim payments in the records are correctly predicted.

Without using classification algorithm, for this particular test dataset, the total payment calculated by using initial group percentages for weighted average yields the real aggregate payment amount: 2,301,937. In comparison, the predicted aggregate payment amount is 2,323,561 and prediction is 0.9394% more than the real aggregate payment amount. The root mean squared error (RMSE) is the measurement used to calculate the error of prediction:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(s_i - \hat{s}_i)^2}, \tag{10}$$

The RMSE calculated based on the predicted total payment amount and the real total payment amount is 483.4353.

Experiment results also show that this model outperforms other general predictive models such as generalized linear model based on this dataset.

4.4 NLP-BERT Enhanced Portfolio Allocation Model

Traditional methods in actuarial science were built based on statistical and mathematical concepts. It is difficult to process textual data directly by traditional methods in general; therefore, textual data was usually neglected in actuarial science. However, textual data can contain hidden information which may be useful when analyzing limited data. Some NLP techniques, such as word embeddings that convert text into numerical vectors, could help process textual data. Although these techniques provide a solution for incorporating texts into traditional actuarial models, the models may not be able to handle the converted values as high dimensional vectors. Unlike general high dimensional data, common dimension reducing techniques are not appropriate for the vectors extracted from texts. To process this high dimensional data, models enhanced with machine learning based on NLP techniques should be considered, and are a good strategy for their use.

4.4.1 BERT Enhanced Data-Based Portfolio Allocation Model

The next goal of this work is to enhance the data-based portfolio model by using BERT to extract features from the textual data. BERT-based neural networks are applied to process the extracted features for the model enhancement. BERT converts textual descriptions into vectors as BERT features through the fine-tuning

step. The input of the neural network classifier is the vector converted contains the BERT-extracted features. The neural networks are formed by incorporating BERT pre-trained model with one additional output layer to perform a multi-class classification so that the output of the neural networks model is a matrix of probabilities.

In general, a multi-layered feed forward neural network uses the a back-propagation algorithm for the multi-class classification problem. The input and output of the hidden nodes of neural networks can be defined as:

$$a_j = \sum_{i=1}^{n} w_{ji} x_i,\ z_j = \boldsymbol{f}(a_j) \text{ for } j = 1, 2, ..., H, \tag{11}$$

where x_i is in the input D, w_{ji} is the corresponding weight associated with the jth node, H is the number of hidden nodes, and $\boldsymbol{f}$ is a logistic function as the active function.

For the enhanced data-based portfolio model, let

$$\hat{Y} = g(w_1^T x_1 + w_2^T x_2 + ... + w_m^T x_m + b), \tag{12}$$

where g is the activation function with softmax, w_1, w_2, ..., w_m are vectors of weights corresponding to the terms $x_1, x_2, \cdots, x_m$. The length of $w_1, w_2, \cdots, w_m$ are determined by the length of the corresponding input $x_1, x_2, \cdots, x_m$. In this study, $x_1, x_2, \cdots, x_m$ are the features extracted from the BERT hidden layers and b is the bias vector. $\hat{Y}$ is the output of the classifier consisting of the elements

$y_1, y_2, \cdots$ as the probabilities of corresponding classes for each single prediction. Let z_i be the outputs in the hidden layers of the neural network as the input of the softmax layer and notice the sum of y_i's is equal to 1. We have

$$y_i = \frac{e^{z_i}}{\sum_k e^{z_k}},$$

The neural network model was trained using historical data in the claims table. The loss function of the model is measured by cross-entropy:

$$Loss = \sum_i t_i \ln(y_i), \tag{13}$$

where t_i is the true value and y_i is the prediction by softmax. The procedure followed for the BERT enhanced data-based portfolio allocation model is listed:

Several different sets of hyper-parameters were tested to achieve the best result. The tested values of hyper-parameters are listed in Table 10. The adjusted value of max_seq_length was 64 and $train\_batch\_size$ was 32.

	Hyper parameter
max_seq_length	16, 32, 64, 96, 128, 144
$train\_batch\_size$	8, 16, 32, 48
$Modeltype$	$BERT_{Base}$
$Optimizer$	Adam

Table 10: Fine-tuning Hyper Parameters

With the probabilities associated with portfolios and the corresponding loss

Model 2 BERT Enhanced Data-Based Portfolio Allocation Model

Require: Textual data x_i and x_j, Class labels Y_j
Ensure: Total predicted loss $\mathbb{E}(T)$
1: Split data into training set with size of m, and testing set with size of n
2: **for** $j = 1$ to m **do**
3: Train BERTMULTICLASSCLASSIFICATION model using x_j, and Y_j
4: **end for**
5: Calculate sample mean $\mathbb{E}(X|Y_1), \mathbb{E}(X|Y_2), \mathbb{E}(X|Y_3), \mathbb{E}(X|Y_4)$ using training set.
6: **for** $i = 1$ to n **do**
7: $y_{i1}, y_{i2}, y_{i3}, y_{i4} \leftarrow$ BERTMULTICLASSCLASSIFICATION(x_i)
8: **end for**
9: $\mathbb{E}(T) \leftarrow 0$
10: **for** $i = 1$ to n **do**
11: $\mathbb{E}(T) \leftarrow \mathbb{E}(T) + \mathbb{E}(X|Y_1) \times y_{i1} + \mathbb{E}(X|Y_2) \times y_{i2} + \mathbb{E}(X|Y_3) \times y_{i3} + \mathbb{E}(X|Y_4) \times y_{i4}$
12: **end for**

severity values, the aggregate loss can be calculated as a weighted average.

4.4.2 Validation and Classification Accuracy Results

A hold-out validation test was designed to achieve better parameters and prevent over-fitting for the model based on the split data. Ten percent of the training data was used as the development set in order to adjust the parameters. To show whether the information from the textual data was helpful, two classification models were tested: The BERT-based neural network multi-class classification model that incorporated information extracted from the textual description of the CAUSE attribute, and the logistic regression multi-class classification model that only used explanatory variable with numerical values discussed in last section.

The classification predictions and errors of the two models are shown in Tables 11-16 in percentages, respectively. The measurements of Precision, Recall, and F1 Score are usually applied to examine classifiers. Precision, also called Positive Predictive Value, is the ratio of True Positives to True Positives and False Positives. A low value of precision may indicate a large number of False Positives. Recall, also called Sensitivity, is the ratio of True Positives to True Positives and the number of False Negatives. A low value of Recall may indicate in a large number of False Negatives. F1-Score is defined by $2 \times \frac{Precision \times Recall}{Precision + Recall}$. The F1-Score is the measurement of balance between Precision and Recall.

Prediction \ Reality	ND	LD	PD	OD	Total
ND	1236	97	252	256	1841
LD	41	486	12	35	574
PD	247	20	955	69	1291
OD	283	79	48	581	991
Total	1807	682	1267	941	4697

Table 11: BERT based Neural Network Multi-class Classification Contingency Table

Prediction \ Reality	ND	LD	PD	OD	Total
ND	1028	178	338	324	1868
LD	60	332	6	30	428
PD	516	98	825	290	1729
OD	203	74	98	297	672
Total	1807	682	1267	941	4697

Table 12: Logistic Multi-class Classification Contingency Table

Prediction \ Reality	ND	LD	PD	OD	Total
ND	26.3147%	2.0651%	5.3651%	5.4503%	39.1952%
LD	0.8729%	10.3470%	0.2555%	0.7452%	12.2206%
PD	5.2587%	0.4258%	20.3321%	1.4690%	27.4856%
OD	6.0251%	1.6819%	1.0219%	12.3696%	21.0986%
Total	38.4714%	14.5199%	26.9747%	20.0341%	100%

Table 13: BERT based Neural Network Multi-class Classification Contingency Table in Percentage

Prediction \ Reality	ND	LD	PD	OD	Total
ND	21.8863%	3.7897%	7.1961%	6.8980%	39.7701%
LD	1.2774%	7.0683%	0.1277%	0.6387%	9.1122%
PD	10.9857%	2.0864%	17.5644%	6.1742%	36.8107%
OD	4.3219%	1.5755%	2.0864%	6.3232%	14.3070%
Total	38.4714%	14.5199%	26.9747%	20.0341%	100%

Table 14: Logistic Regression Multi-class Classification Contingency Table in Percentage

The classification accuracy, the number of correct predictions divided by the total number of predictions, is 0.69. In Table 15 and Table 16, it shows that the BERT-based enhancement model clearly improves the logistic classification based model. Using the BERT-enhanced model, we obtain a predicted aggregate payment amount of 2,371,387 by sample means, which is 3.017% more than the real aggregate payment amount.

Class	Precision	Recall	F1-Score
ND	0.68	0.66	0.67
LD	0.80	0.72	0.75
PD	0.72	0.77	0.74
OD	0.60	0.61	0.60

Table 15: BERT based Neural Network Multi-class Classification Accuracy

Class	Precision	Recall	F1-Score
ND	0.55	0.57	0.56
LD	0.78	0.49	0.60
PD	0.48	0.65	0.55
OD	0.44	0.32	0.37

Table 16: Logistic Regression Multi-class Classification Accuracy

Modern Portfolio Theory analyzes variances and correlations of portfolio data. We applied the data-split algorithm 10 times to form 10 sets of training and testing data based on 11,742 claim records, respectively, then calculate the portfolio variances as well as the total loss variances based on that and obtain the following results for possible further analysis and reference. The variances are listed in Tables 17-21.

	Mean	Varuance	Std.Dev
Test 1	353.64	52943.05	230.09
Test 2	360.45	46626.26	215.93
Test 3	358.08	47529.97	218.01
Test 4	356.41	49381.31	222.22
Test 5	355.88	48836.88	220.99
Test 6	357.86	48539.95	220.32
Test 7	361.06	52235.62	228.55
Test 8	357.85	50983.85	225.8
Test 9	361.36	54067.25	232.52
Test 10	353.18	45373.33	213.01
Average	357.58	49651.75	222.74

Table 17: Mean, Variance, and Standard Deviation of ND

	Mean	Varuance	Std.Dev
Test 1	106.06	9367.17	96.78
Test 2	101.34	7596.26	87.16
Test 3	104.30	7766.21	88.13
Test 4	102.52	8422.72	91.78
Test 5	101.27	6485.54	80.53
Test 6	101.13	8721.01	93.39
Test 7	105.12	9066.64	95.22
Test 8	99.61	5910.01	76.88
Test 9	102.57	8321.76	91.22
Test 10	102.17	8107.54	90.04
Average	102.61	7976.49	89.11

Table 18: Mean, Variance, and Standard Deviation of LD

	Mean	Varuance	Std.Dev
Test 1	1078.71	400787.9	633.08
Test 2	1084.86	425849.4	652.57
Test 3	1070.66	390180.2	624.64
Test 4	1077.58	395893.6	629.2
Test 5	1085.11	415975.6	644.96
Test 6	1081.15	417208.7	645.92
Test 7	1081.95	389373.3	624
Test 8	1070.86	393895	627.61
Test 9	1080.27	417659.3	646.27
Test 10	1067.29	382674.2	618.61
Average	1077.84	402949.7	634.69

Table 19: Mean, Variance, and Standard Deviation of PD

	Mean	Varuance	Std.Dev
Test 1	216.51	17447.37	132.09
Test 2	217.53	20196.02	142.11
Test 3	215.06	19115.55	138.26
Test 4	212.02	16627.4	128.95
Test 5	215.54	17717.74	133.11
Test 6	216.99	17714.44	133.1
Test 7	213.24	17625.14	132.76
Test 8	221.42	20937.27	144.7
Test 9	214.96	16377.73	127.98
Test 5	218.81	19807.49	140.74
Average	216.21	18356.61	135.38

Table 20: Mean, Variance, and Standard Deviation of OD

	Mean	Varuance	Std.Dev
Test 1	494.39	274652.4	524.07
Test 2	490.37	274804.3	524.22
Test 3	486.37	262727.1	512.57
Test 4	487.90	268345.6	518.02
Test 5	489.17	274915.2	524.32
Test 6	495.71	279517.2	528.69
Test 7	493.73	269866.4	519.49
Test 8	497.13	272704.5	522.21
Test 9	494.17	278842.1	528.15
Test 10	489.1	263875.3.2	513.69
Average	491.8	272035	521.54

Table 21: Mean, Variance, and Standard Deviation of Original Total Loss

4.5 BERT-Based Severity Model for Multi-Class Aggregate Loss Prediction

In the previous calculation of aggregate loss using data-based portfolio allocation model, the severity means were determined by the sample means. In this section, we apply the BERT-based loss severity model discussed in Chapter 3 to portfolio data distributions to predict corresponding aggregate losses. The conditional expected severities $\mathbb{E}(S_i|X_i)$ in Equation 9 are the expected aggregate loss amounts for each class, and the distribution of the total loss would be a weighted average of the mean severities of subclasses.

The severity level is defined based on the same scale as that in previous chapter. For each severity level, the losses are assumed to be uniformly distributed. By observing the histogram of the severity level for each class, gamma distributions are fitted for all classes as the same was done in Chapter 3. The fitted distributions are listed as follows from Table 22 to Table 25.

4.5.1 Predicted Aggregate Loss and Comparison

The average losses are calculated using the definition of the rescaled severity by the assumption that the losses at each severity level are distributed uniformly. The corresponding weights in percentages were obtained by the ratio of class counts to total count in the training data set. The expected severity is measured using the Equation 7. The aggregate loss of the proposed model is 2,509,164 for 4,697 claims. Comparing to reality of 2,301,937, the prediction is 9% greater than

Parameter	Value
α	8.362012
β	2.312543
mean	3.615938
variance	1.56362

Table 22: Gamma Distribution of ND

Parameter	Value
α	2.383158
β	1.296036
mean	1.838806
variance	1.418793

Table 23: Gamma Distribution of LD

Parameter	Value
α	14.80725
β	2.631891
mean	5.626087
variance	2.137659

Table 24: Gamma Distribution of PD

Parameter	Value
α	6.11038
β	2.075307
mean	2.944325
variance	1.418742

Table 25: Gamma Distribution of OD

the real losses. The results of three models are compared by RMSE in Table 26. The the two BERT models incorporated textual data have improved RMSE against the data-based portfolio model without textual information.

Model	RMSE
Data-Based Portfolio Model (sample mean)	483.4353
NLP-BERT Enhanced Portfolio Allocation Model (sample mean)	430.6993
BERT-Based Severity Level Portfolio Model	431.013

Table 26: Comparison of RMSE for Predictive Models

CHAPTER 5

Conclusions and Final Remarks

In this dissertation, the two studies of NLP applications demonstrated the potential possibilities to improve predictive modeling in actuarial science. In Chapter 3, the BERT enhanced severity level model is developed. The model improves the stability of severity estimation in traditional aggregate loss modeling among different datasets. The study showed a way to exploit textual data which is not used in many traditional applications. There are three models described in Chapter 4, the data-based portfolio allocation model for predictive analytics is very practical in many applications and is particularly promising when enhanced by NLP tools, such as BERT. The BERT enhanced portfolio model improves the prediction accuracy through the improvement of classification rates using textual data.

There are many improvements that can be made in future research for the models:

1. For the severity modeling, different re-scaling of severity can be considered and different distributions can be fitted based on the re-scaling to achieve better prediction. The possibilities of using NLP for frequency modeling can also be considered.

2. If the loss severity distributions can be estimated, then the data-based portfolio model can be used in the portfolio tail value at risk calculation for further risk evaluation [49].

3. Modern Portfolio Theory analyzes variances and correlations of portfolio data. More advanced methods and tools in MPT can be applied for model development along with certain criteria emphasized on variances and correlations.

4. Some NLP tools, such as BERT, require a lot of computational power and large amounts of computer memory. The BERT models in this dissertation used the $\mathbf{BERT}_{Base}$ model released in later 2018 and trained by a Geforce RTX 2080ti graphic card. The more recent BERT model such as the $\mathbf{BERT}_{small}$ model released in 2020 [53], can be considered for some environments with limited computational resources.

References

[1] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938, 2018.

[2] Charu C Aggarwal and ChengXiang Zhai. An introduction to text mining. In *Mining text data*, pages 1–10. Springer, 2012.

[3] Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*, 2019.

[4] Chidanand Apté, Fred Damerau, and Sholom M Weiss. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems (TOIS)*, 12(3):233–251, 1994.

[5] Lalit R Bahl, Frederick Jelinek, and Robert L Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE transactions on pattern analysis and machine intelligence*, (2):179–190, 1983.

[6] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, 2014.

[7] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The journal of machine learning research*, 3:1137–1155, 2003.

[8] Daniel M Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. Nymble: a high-performance learning name-finder. *arXiv preprint cmp-lg/9803003*, 1998.

[9] Mita K Dalal and Mukesh A Zaveri. Automatic text classification: a technical review. *International Journal of Computer Applications*, 28(2):37–40, 2011.

[10] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[11] Gerald DeJong. Prediction and substantiation: A new approach to natural language processing. *Cognitive Science*, 3(3):251–273, 1979.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[13] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74, 2008.

[14] Yair Even-Zohar and Dan Roth. A sequential model for multi-class classification. *arXiv preprint cs/0106044*, 2001.

[15] Edward W Frees and Ping Wang. Copula credibility for aggregate loss models. *Insurance: Mathematics and Economics*, 38(2):360–373, 2006.

[16] Zhengjie Gao, Ao Feng, Xinyu Song, and Xi Wu. Target-dependent sentiment classification with bert. *IEEE Access*, 7:154290–154299, 2019.

[17] José Garrido, Christian Genest, and Juliana Schulz. Generalized linear models for dependent frequency and severity of insurance claims. *Insurance: Mathematics and Economics*, 70:205–215, 2016.

[18] Riccardo Gatto. Values and tail values at risk of doubly compound inhomogeneous and contagious aggregate loss processes. *Mathematical and computer modelling*, 54(5-6):1523–1535, 2011.

[19] R Grisham and B Sundheim. Message understanding: a brief history. In *Proceedings of the Sixth Message Understanding Conference*. San Francisco, 1996.

[20] Susanne Gschlößl and Claudia Czado. Spatial modelling of claim frequency and claim size in non-life insurance. *Scandinavian Actuarial Journal*, 2007(3):202–225, 2007.

[21] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[22] Michael Ho, Zheng Sun, and Jack Xin. Weighted elastic net penalized mean-variance portfolio design and computation. *SIAM Journal on Financial Mathematics*, 6(1):1220–1244, 2015.

[23] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[24] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[25] Himchan Jeong, Emiliano A Valdez, Jae Youn Ahn, and Sojung Park. Generalized linear mixed models for dependent compound risk models. *Available at SSRN 3045360*, 2017.

[26] Jing Jiang. Information extraction from text. In *Mining text data*, pages 11–41. Springer, 2012.

[27] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 1972.

[28] Sameem Abdul Kareem, S Raviraja, Namir A Awadh, Adeeba Kamaruzaman, and Annapurni Kajindran. Classification and regression tree in prediction of survival of aids patients. *Malaysian Journal of Computer Science*, 23(3):153–165, 2010.

[29] Kevin Kuo. Deeptriangle: A deep learning approach to loss reserving. *Risks*, 7(3):97, 2019.

[30] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR, 2014.

[31] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[32] Gee Y Lee, Scott Manski, and Tapabrata Maiti. Actuarial applications of word embedding models. *ASTIN Bulletin: The Journal of the IAA*, 50(1):1–24, 2020.

[33] Susanna Levantesi and Virginia Pizzorusso. Application of machine learning to mortality modeling and forecasting. *Risks*, 7(1):26, 2019.

[34] Hans Peter Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development*, 1(4):309–317, 1957.

[35] Mikael Lundin, Johan Lundin, HB Burke, Sakari Toikkanen, Liisa Pylkkänen, and Heikki Joensuu. Artificial neural networks applied to survival prediction in breast cancer. *Oncology*, 57(4):281–286, 1999.

[36] Antoine Ly, Benno Uthayasooriyar, and Tingting Wang. A survey on natural language processing (nlp) and applications in insurance. *arXiv preprint arXiv:2010.00462*, 2020.

[37] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[38] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[39] Tomas Mikolov, Jiri Kopecky, Lukas Burget, Ondrej Glembek, et al. Neural network based language models for highly inflective languages. In *2009 IEEE international conference on acoustics, speech and signal processing*, pages 4725–4728. IEEE, 2009.

[40] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[41] Yi Lu Murphey and Yun Luo. Feature extraction for a multiple pattern classification neural network system. In *Object recognition supported by user interaction for service robots*, volume 2, pages 220–223. IEEE, 2002.

[42] L Ohno-Machado, MG Walker, and MA Musen. Hierarchical neural networks for survival analysis. *Medinfo*, 8(Pt 1):828–832, 1995.

[43] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 confer-*

ence on empirical methods in natural language processing (EMNLP), pages 1532–1543, 2014.

[44] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[45] Marco Polignano, Pierpaolo Basile, Marco De Gemmis, Giovanni Semeraro, and Valerio Basile. Alberto: Italian bert language understanding model for nlp challenging tasks based on tweets. In *6th Italian Conference on Computational Linguistics, CLiC-it 2019*, volume 2481, pages 1–6. CEUR, 2019.

[46] Zhiyu Quan and Emiliano A Valdez. Predictive analytics of insurance claims using multivariate decision trees. *Dependence Modeling*, 6(1):377–407, 2018.

[47] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[48] Jiandong Ren. A multivariate aggregate loss model. *Insurance: Mathematics and Economics*, 51(2):402–408, 2012.

[49] Jimmy Risk and Michael Ludkovski. Sequentialdesign and spatial modeling for portfolio tail risk measurement. *SIAM Journal on Financial Mathematics*, 9(4):1137–1174, 2018.

[50] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[51] Fengqin Tang and Jianming Bai. Precise large deviations for aggregate loss process in a multi-risk model. *Journal of the Korean Mathematical Society*, 52(3):447–467, 2015.

[52] Cynthia A Thompson, Mary Elaine Califf, and Raymond J Mooney. Active learning for natural language parsing and information extraction. In *ICML*, pages 406–414. Citeseer, 1999.

[53] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*, 2019.

[54] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394, 2010.

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[56] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[57] Mario V Wüthrich. Machine learning in individual claims reserving. *Scandinavian Actuarial Journal*, 2018(6):465–480, 2018.

[58] Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. Bert post-training for review reading comprehension and aspect-based sentiment analysis. *arXiv preprint arXiv:1904.02232*, 2019.

[59] Shuzhe Xu, Salvador E Barbosa, and Don Hong. Bert feature based model for predicting the helpfulness scores of online customers reviews. In *Future of Information and Communication Conference*, pages 270–281. Springer, 2020.

[60] Diego Zappa, Mattia Borrelli, Gian Paolo Clemente, and Nino Savelli. Text mining in insurance: From unstructured data to meaning. *Variance, In press. Available online: https://www. variancejournal. org/articlespress/(accessed on 1 June 2019)*, 2019.

[61] Chuanlong Zhang. Aggregate loss prediction using multiple-class classification techniques. *MS Thesis, MTSU*, June 2021.

APPENDICES

APPENDIX A

TABLES

Severity	Count	Modified
1	191	40
2	715	424
3	864	1051
4	1965	1963
5	938	1294
6	1112	877
7	413	442
8	351	472
9	238	225
10	14	13

Severity	Count	Modified
1	188	37
2	694	417
3	883	1034
4	1971	1991
5	934	1282
6	1115	890
7	406	446
8	361	470
9	232	222
10	17	12

Severity	Count	Modified
1	186	37
2	709	414
3	873	1056
4	1948	1948
5	949	1300
6	1115	901
7	410	441
8	360	467
9	237	225
10	14	12

Severity	Count	Modified
1	184	39
2	704	414
3	879	1055
4	1949	1973
5	945	1282
6	1130	893
7	402	437
8	360	467
9	232	229
10	16	12

Severity	Count	Modified
1	194	40
2	700	405
3	877	1059
4	1944	1962
5	952	1285
6	1117	905
7	417	437
8	353	477
9	232	220
10	15	11

Severity	Count	Modified
1	195	37
2	711	417
3	857	1049
4	1951	1957
5	957	1303
6	1120	898
7	409	444
8	355	461
9	232	221
10	14	14

Table A.1: Severity Level and BERT-modified Severity Level of 10 Tests

Severity	Count	Modified
1	189	39
2	711	413
3	879	1053
4	1951	1972
5	936	1296
6	1121	888
7	407	431
8	360	478
9	231	218
10	16	13

Severity	Count	Modified
1	191	40
2	707	413
3	877	1055
4	1967	1975
5	945	1291
6	1106	892
7	399	437
8	364	457
9	228	228
10	17	13

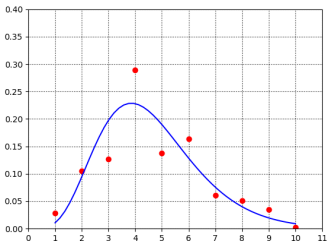
Severity	Count	Modified
1	182	40
2	706	405
3	869	1035
4	1939	1971
5	953	1298
6	1127	896
7	409	439
8	360	474
9	240	230
10	16	13

Severity	Count	Modified
1	189	33
2	707	418
3	858	1039
4	1948	1970
5	957	1298
6	1127	893
7	403	440
8	363	469
9	235	228
10	14	13

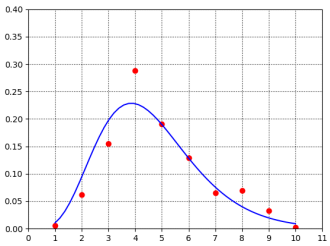
Table A.2: Severity Level and BERT-modified Severity Level of 10 Tests

APPENDIX B

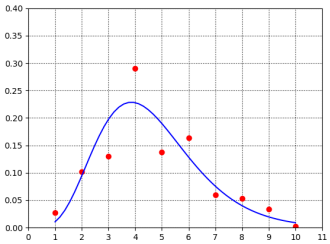
FIGURES



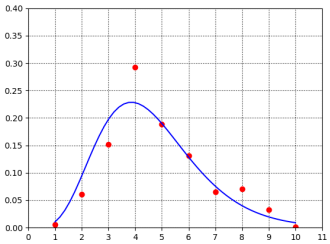
(a) test1



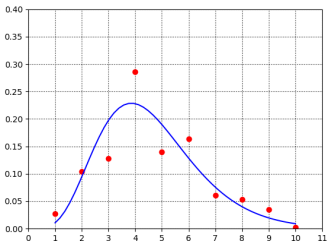
(b) test1 BERT



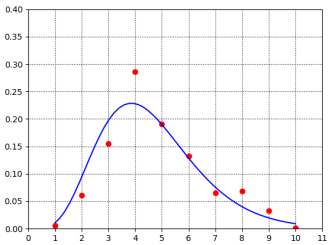
(c) test2



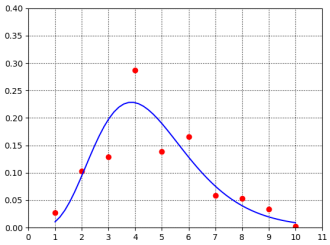
(d) test2 BERT



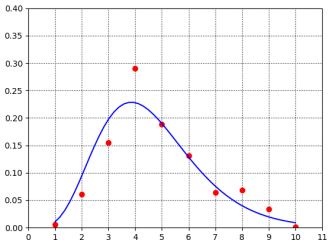
(e) test3



(f) test3 BERT

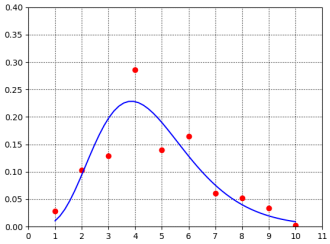


(g) test4

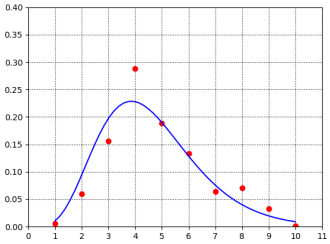


(h) test4 BERT

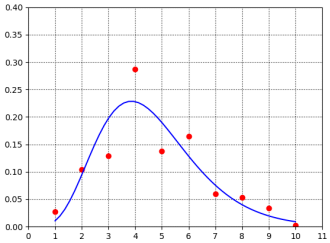
Figure 3: Fitted Gamma Distributions of 10 Tests



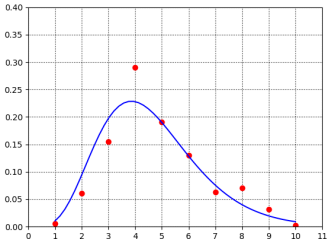
(a) test5



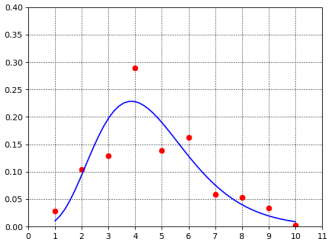
(b) test5 BERT



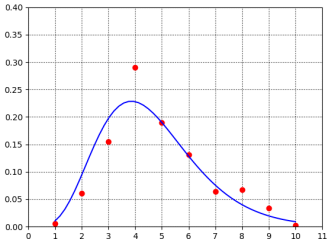
(c) test6



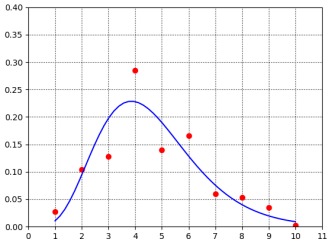
(d) test6 BERT



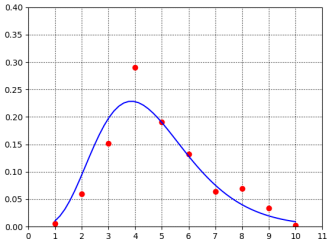
(e) test7



(f) test7 BERT

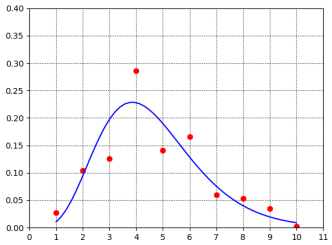


(g) test8

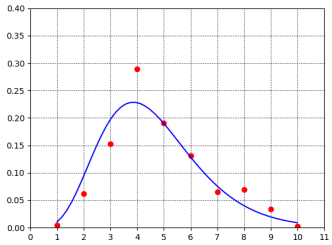


(h) test8 BERT

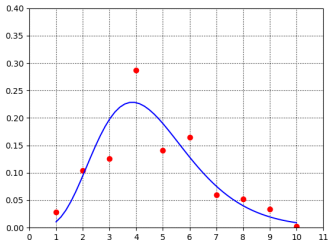
Figure 4: Fitted Gamma Distributions of 10 Tests



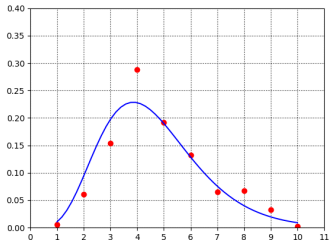
(a) test9



(b) test9 BERT

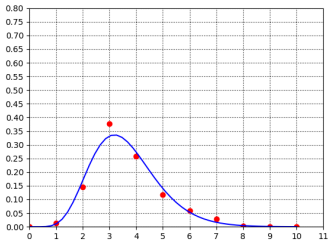


(c) test10

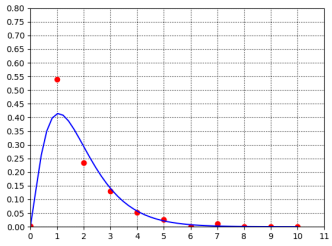


(d) test10 BERT

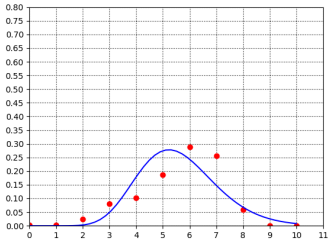
Figure 5: Fitted Gamma Distributions of 10 Tests



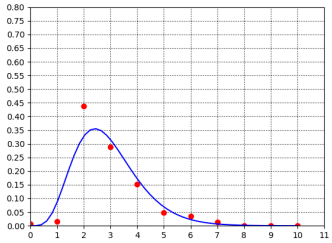
(a) ND



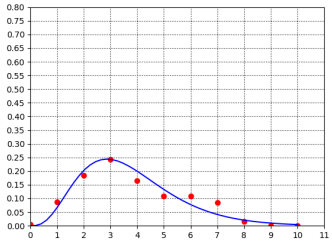
(b) LD



(c) PD



(d) OD



(e) Original

Figure 6: Fitted Gamma Distribution for Each Class