ACTUARIAL APPLICATIONS OF WORD EMBEDDING MODELS

BY

GEE Y LEE, SCOTT MANSKI AND TAPABRATA MAITI

ABSTRACT

In insurance analytics, textual descriptions of claims are often discarded, because traditional empirical analyses require numeric descriptor variables. This paper demonstrates how textual data can be easily used in insurance analytics. Using the concept of word similarities, we illustrate how to extract variables from text and incorporate them into claims analyses using standard generalized linear model or generalized additive regression model. This procedure is applied to the Wisconsin Local Government Property Insurance Fund (LGPIF) data, in order to demonstrate how insurance claims management and risk mitigation procedures can be improved. We illustrate two applications. First, we show how the claims classification problem can be solved using textual information. Second, we analyze the relationship between risk metrics and the probability of large losses. We obtain good results for both applications, where short textual descriptions of insurance claims are used for the extraction of features.

KEYWORDS

Risk mitigation, insurance claims adjustment, word embedding, text mining, word2vec, GloVe, word similarity, generalized additive models, actuarial modeling, classification, logistic regression, loss modeling.

1. INTRODUCTION

When an insurance claim arises, one of the first things that is reported to a claims manager of an insurance company is a short textual description of the insurance claim, along with demographic information regarding the policyholder. Losses may be reported via a notice form, where one of the common forms used is the Association for Cooperative Operations Research and Development (ACORD) form; see page 7.16 of Kearney (2010). Upon this initial report, the claims department is responsible for a number of important

Astin Bulletin 50(1), 1-24. doi:10.1017/asb.2019.28

tasks. The claims department would identify the policy and set adequate reserves, contact the insured, investigate the claim, document the claim, and determine the precise cause of loss, liability, and the loss amount. In this process, the claims manager would coordinate with the actuarial department in order to predict the amount of insurance claim and set the adequate case reserve for each claim. In practice, parametric loss models are fit to data, and the resulting distribution is used for the prediction of the ultimate claim amount.

In this process, regression analysis helps us understand the relationship among variables. The goal of a standard regression model is to understand the relationship between the response variable $\mathbf{y}$ and explanatory variables $\mathbf{X}$, and predict future responses under a set of assumptions. The relationship $g\{\mathbb{E}[\mathbf{y}]\} = \mathbf{X}\boldsymbol{\beta}$, where typically a distributional assumption is imposed on the error, allows us to interpret the relationship between $\mathbf{y}$ and $\mathbf{X}$ in a systematic way. In case $\mathbf{y}$ is a binary response, logistic regression can be used. For a general overview of regression, see Frees (2009). When forming the $\mathbf{X}$ matrix for empirical research using traditional approaches, useful information is discarded from the analysis, because traditional regression analysis requires numeric descriptor variables. Textual information has been one example.

In this paper, we demonstrate how textual information can be transformed into a format where traditional analysis can be performed. Then, the extracted information is used to build a regression model for the response variable of interest. Through this demonstration, we show how information can be extracted from textual data. We demonstrate two illustrative case studies in insurance claims classification and insurance risk mitigation. We believe these approaches demonstrate how text-processing methods can improve insurance analytics in the actuarial practice. In addition, understanding the factors that relate with large insurance losses may help us mitigate future insurance losses.

Text mining has been utilized in a variety of contexts, including but not limited to spam filtering, sentiment analysis, customer churn, stock returns, and politics. Recently, Mikolov *et al.* (2013) introduced the vector representation for words, and Pennington *et al.* (2014) demonstrated how the Global Vectors for word representation (GloVe) algorithm can be used to obtain analogous representations for words. Neural network-based text analysis methods essentially treat the vector representation of words as the hidden layer of a neural network, which is trained over a huge corpus of text. The resulting word embeddings can be used for classic machine learning tasks, such as text classification. Recurrent neural networks (RNNs) can be utilized to classify large textual data, essentially treating textual data as a sequence of words; see Goodfellow *et al.* (2016) and Goldberg (2017) for a treatment of neural networks, and how they can be utilized for prediction tasks.

Neural network-based text models are powerful in terms of their prediction. Meanwhile, we are interested in models that allow us to interpret the relationship between explanatory variables and response variables. One way this can

https://doi.org/10.1017/asb.2019.28 Published online by Cambridge University Press

be achieved is to consider projecting the vector representation of words onto a set of axes, which we are able to understand. Given a vector representation of phrases, instead of using the vectors directly inside a prediction algorithm, one may consider projecting the vectors onto axes defined by known keywords. This can be achieved by taking the cosine similarity between the phrase and the predefined keywords. This is the approach we will take in this paper.

Generalized additive models (GAMs) extend linear models to contain smooth functions for each of the covariates, while retaining inference about the functions. Applications of GAMs have been discussed in Hastie and Tibshirani (1990). These applications include studying kyphosis in laminectomy patients, atmospheric ozone concentration, and the intensity of ischemic heart disease risk factors, among others.

Moreover, Hastie *et al.* (2009) describes an example of utilizing GAMs to classify e-mails as spam. While this example does analyze text, the method has several significant differences from that of our analysis. The spam example observes the number of occurrences of certain words, and fits a GAM with each word as a covariate. While the spam example in Hastie *et al.* (2009) also discusses the interpretability of the model, the primary goal is to predict the probability of an e-mail being spam. In our analysis, we start by quantifying the similarity between a description and a series of selected words. The main interest in this analysis is the interpretation of the smoothing functions. This provides a much more complete explanation of the various factors that lead to high losses, and therefore, may provide for a more improved strategy of risk mitigation. A recent work, Wood (2017), provides a comprehensive overview of GAMs.

There is a vast literature in insurance claims modeling, where parametric models are employed to understand insurance claims distributions. To the best of our knowledge, combining text mining approaches with loss modeling is a new approach, which has not been attempted in the past. In particular, there seems to be no prior work utilizing text mining approaches to empirically understand and model insurance claims data. The rest of the paper proceeds in the following order: In Section 2, the Wisconsin Local Government Property Insurance Fund (LGPIF) data are introduced. In Section 3, the word similarity model is introduced. In Section 4, two applications of word similarity models are presented: insurance claims categorization and insurance risk mitigation. In Section 5, concluding remarks are provided.

2. DATA

2.1. Overview

In this section, we provide some summary statistics for the data set. For our case study, we utilize a unique data set of claim descriptions and loss amounts from the Wisconsin LGPIF. The data set is obtained from the Office of the

https://doi.org/10.1017/asb.2019.28 Published online by Cambridge University Press

Commissioner of Insurance (OCI) of Wisconsin. The Wisconsin LGPIF has been established to make property insurance available for local government units. The property fund essentially acts as an insurance company in the area, providing property coverage for thousands of government entities.

In Table 1, it is interesting to observe that claim categories with high frequency tend to have low severity, while claim categories with low frequency tend to have high severity. In order to illustrate this effect, the table has been sorted in decreasing order of the number of observations N. Modelers have studied insurance claim frequency and severity models, and empirically, it has been discovered that claim frequencies and severities are often correlated; see Frees *et al.* (2016).

2.2. Data generation

The number of claims observations is 4991 in the training sample and 1039 in the validation sample, which totals to 6030 observations. The data used for this paper are already in tabular form corresponding to the data generating processes in Section 4, and we consider the data cleaning process, including the metadata analysis, as a black-box process that has already been performed by the provider of the data. We assume all claims are closed, and the claim amounts are fixed.

Descriptions for the observed insurance claims are recorded in the data set. These claim descriptions are human generated, and there are 2797 unique words found in the training sample and validation sample all together. Figure 1 shows a projection of the word vectors in a two-dimensional space, for common words found in the data set. Word vectors are explained in Section 3.1. For now, imagine that there exists a framework, where every word corresponds to some two-dimensional vector, with related words having similar vector representations. A plot of common words found in the claim descriptions file may look like Figure 1.

In Figure 1, notice that *library* and *museum* appear close to each other, since they have similar functions, and hence may appear in similar contexts. Also, *graffiti*, *vandalism*, *theft*, and *stolen* all appear at a similar location on the plot. Imagine drawing an arrow from the point $(0, 0)$ to the word, and the reader may see that the vector corresponding to each of these words are very similar to one another. The angle between the words is small, and hence the cosine of the angle between the words would be large (close to 1). Another way to say this is that the dot product between the words is large. Now consider the word *hail* and its corresponding vector, and compare it with the vector corresponding to *graffiti*. The two words are somewhat unrelated, and hence the angle between these two words is large. Another way to say this is that the cosine between the unit vectors corresponding to the two vectors is negative, or in other words the dot product is negative.

https://doi.org/10.1017/asb.2019.28 Published online by Cambridge University Press

TABLE 1

SUMMARY STATISTICS OF LOSSES BY CLAIM CATEGORY.

	Validation sample					*Training sample*				
Peril	min	median	mean	max	N	min	median	mean	max	N
Vandalism	1	500	6190	981,599	310	6	587	2084	207,565	1774
Vehicle	1	3000	5662	135,268	227	37	2500	3905	111,740	852
Lightning	500	5000	11,623	88,603	123	1	4431	11,087	655,092	832
Water (weather)	55	19,337	51,608	411,641	38	1	8898	80,432	12,922,218	426
Miscellaneous	70	3025	9723	242,918	103	1	3929	29,150	2,633,822	362
Wind	325	9010	46,304	1,048,683	107	1	4960	18,125	492,478	296
Water (non-weather)	544	6739	60,538	2,672,184	67	1	6306	23,974	1,114,595	202
Fire	125	11,355	83,767	964,150	46	100	8964	81,762	1,570,619	171
Hail	7886	49,184	103,674	332,412	18	124	17,819	145,488	6,615,117	76

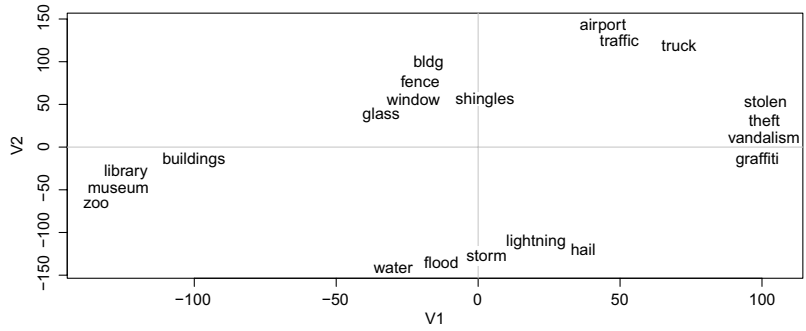

FIGURE 1: Two-dimensional projection of the word embeddings for common words.

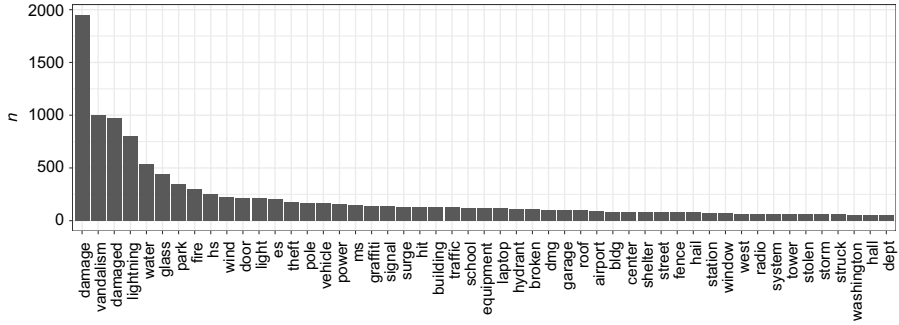

FIGURE 2: Distribution of common words.

2.3. Textual data preprocessing

Figure 2 shows the most common words in the data set. Stop-words such as *a, the, and*, etc., have been removed. No other preprocessing of the words has been performed. Notice that the word *damage* is most frequent in the descriptions. The word *vandalism* is also frequent, as vandalism turns out to be one of the most frequent claim causes in the data set. Abbreviations such as *hs* (high school), *ms* (middle school), *es* (elementary school), *dmg* (damage), and *bldg* (building) also appear in the data set.

Note that both *bldg* and *building* are valid words. Imagine we search for the word *building* in a search engine. In this case, phrases with the word *bldg* and *building* should both appear in the search. We also note that *building* and *buildings* both appear in the vocabulary. The reason why both *bldg* and *buildings* appear when *building* is already in the vocabulary list, is because they are distinct words. How much these words relate to claim occurrence is determined by the cosine similarities of the words with selected key words. The advantage of our approach is that minimal data preprocessing is needed, so that such similar words can all be kept in the data.

Note that the claim descriptions are short phrases, such as *lightning damage to building*, or *vandalism damage at recycle center*. A total of 4991 such

https://doi.org/10.1017/asb.2019.28 Published online by Cambridge University Press

claim descriptions are in the training data set. Each claim description is associated with the loss amount corresponding to the description. Each claim is categorized into one of the following claim categories: vandalism, fire, lightning, wind, hail, vehicle, water (weather related), water (non-weather related), and miscellaneous losses.

Table 1 shows a summary of the loss amounts for each of the nine claim categories. According to the table, vandalism has the lowest average loss amount, while the frequency of vandalism is the highest. Hail damages are the largest in terms of median and mean value, while the frequency is the lowest. The largest hail damage with a loss of 6.6 million is a hail damage to multiple buildings insured by the property fund. It is interesting to observe that the maximum loss amount happened within the weather-related water damages category, which is a loss of 12.9 million. This largest loss corresponds to a water damage to a school. The second most frequent claim category is the vehicles category. This category of claim happens when a vehicle (car, plow, truck, etc.) runs into a government property building or structure, such as a light pole.

To conclude the data description, we provide some ground-truth information regarding the quality of the data. The data have no missing values, and the categories for each claim are all observed. We inspected the claim categorizations and believe that the data quality is mostly dependable, although some errors may exist because the categorization has been performed by human experts.

3. Word Similarity Model

In order to incorporate textual data into a regression analysis, we need a framework for representing words as numeric vectors. In this framework, each word in the vocabulary is represented as a vector of a certain fixed dimension, allowing words to be added and subtracted. For an overview of natural language processing, Manning and Schutze (1999) is recommended. Mikolov *et al.* (2013) describes the neural network approach to natural language processing, and illustrates how the word2vec algorithm can be implemented. Pennington *et al.* (2014) introduced the GloVe algorithm for finding word embeddings.

3.1. Word embeddings

As described briefly in Section 2, we utilize a framework, where every word in the vocabulary is represented by a unique vector representation. The components of the vector each represent a feature of the word. Words with closer meaning will show up closer in the graph when plotted as shown in Figure 1. The fact that words can be represented as vectors allows a lot of flexibility in modeling. Standard vector operations can be performed on words, once this realization has been achieved. For example, the word *queen* may be obtained

https://doi.org/10.1017/asb.2019.28 Published online by Cambridge University Press

algebraically by adding and subtracting vectors: *queen* = *woman* + *king* − *man*. With vector representation of words, the relevance of the search string and the content can be determined by the cosine between the search string and the content. Suppose we are interested in the relevance between the word *queen*, and the vector: *woman* + *king* − *man*. In this case, the cosine between the two vectors will be close to 1. This will indicate that the two vectors are nearly identical. Now consider another word: *lightning*. In this case, the cosine between *queen* and *lightning* will have a low value. Recall that the cosine function ranges from −1 to 1, depending on the angle between two vectors. Hence, a cosine of 1 indicates high relationship, while a cosine of −1 indicates a low relationship.

3.1.1. *Word2vec*

One approach to obtaining the word embeddings is to use the word2vec algorithm. The word2vec algorithm can be illustrated using a simplified example. Suppose we use the following five words as our simplified vocabulary list:

$$V = \{\text{lightning, vandalism, vehicle, building, struck}\} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_5\}.$$

Consider the sentence:

lightning struck building

with center word $\boldsymbol{w}_5$, and context words $C = \{\tilde{\boldsymbol{w}}_1, \tilde{\boldsymbol{w}}_4\}$. What we want is some vector representation $\boldsymbol{W}$ of center words and $\tilde{\boldsymbol{W}}$ of context words. These word embedding matrices are obtained by letting an algorithm read through billions of sentences, maximizing a log likelihood, treating $\boldsymbol{W}$ and $\tilde{\boldsymbol{W}}$ as parameters to be estimated. For this, we can specify the probability of observing a context word $\tilde{\boldsymbol{w}}_j$ given a center word $\boldsymbol{w}_5$ by

$$Pr\left(\tilde{\boldsymbol{w}}_j|\boldsymbol{w}_5\right) = \frac{\exp\left(\tilde{\boldsymbol{w}}_j \cdot \boldsymbol{w}_5\right)}{\sum_{k=1}^{|V|} \exp\left(\tilde{\boldsymbol{w}}_k \cdot \boldsymbol{w}_5\right)}.$$

Going back to the sentence, *lightning struck building*, using a naive Bayes assumption (where we assume the conditional independence of the events of observing context words given a center word), the negative log likelihood becomes

$$L = -\log Pr(\tilde{\boldsymbol{w}}_1|\boldsymbol{w}_5) - \log Pr(\tilde{\boldsymbol{w}}_4|\boldsymbol{w}_5) = -\sum_{\tilde{\boldsymbol{w}}_j \in C} \tilde{\boldsymbol{w}}_j \cdot \boldsymbol{w}_5 + |C| \log \sum_{\tilde{\boldsymbol{w}}_k \in V} \exp\left(\tilde{\boldsymbol{w}}_k \cdot \boldsymbol{w}_5\right).$$

The negative log likelihood is minimized by gradient descent. Note that analytical formulas for the gradient can be obtained based on the likelihood. In practice, using the analytical forms of the gradient speeds up the convergence and makes the algorithm more stable. Extensions of the gradient descent algorithm such as Adam optimizers may be used as well. The $\boldsymbol{W}$ and $\tilde{\boldsymbol{W}}$ matrices start from a random initial matrix and is updated each step of the gradient descent iteration. Repeating this process for millions and billions of center

https://doi.org/10.1017/asb.2019.28 Published online by Cambridge University Press

words and context words, an input word matrix $\boldsymbol{W}$ and a context word matrix $\tilde{\boldsymbol{W}}$ are learned.

Word2vec may use either the continuous bag-of-words (CBOW) model or the continuous skip-gram model, depending on how the log likelihood is defined. In the CBOW model, the log likelihood represents the probability of observing the center word within a window of context words. In the continuous skip-gram model, the probability of observing the context word given a center word is used. Hence, the example shown above would correspond to a continuous skip-gram model.

Nowadays, word embedding matrices trained by word2vec can be downloaded from Google's website (https://code.google.com/archive/p/word2vec/). An extension of word2vec, which uses character *n*-grams, is fastText, which can be downloaded from (https://fasttext.cc). FastText is a library for learning word embeddings, created by Facebook's AI Research (FAIR) lab.

3.1.2. *Global vectors for word representation*

In this paper, we use the pretrained word embeddings obtained via an algorithm called GloVe, developed by Pennington *et al.* (2014). We note that other methods to create word embedding matrices exist. From our perspective, the end result of word2vec and GloVe is similar. We chose GloVe because it is a straightforward algorithm based on word counts, and the approach is well documented with an emphasis on reproducibility. In practice, the GloVe algorithm has additional benefits over word2vec in that the algorithm is more easily parallelized. GloVe word embedding matrices can be downloaded from https://nlp.stanford.edu/projects/glove/. From this website, the 300 dimension word vectors containing 400 thousand vocabularies, trained over 6 billion tokens appearing in the Wikipedia corpus, have been downloaded. The algorithm used for this particular word embedding is to minimize a cost function, which has the form

$$J = \sum_{s=1}^{|V|} \sum_{t=1}^{|V|} \Psi(M_{s,t}) \left(\boldsymbol{w}_s \cdot \tilde{\boldsymbol{w}}_t + b_s + \tilde{b}_t - \log\left(M_{s,t} + 1\right) \right)^2,$$

where $|V|$ is the size of the vocabulary, b_s, $\tilde{b}_t$ are bias terms, $M_{s,t}$ are entries of the co-occurrence matrix for all the words found in the corpus over which the algorithm is being applied, $\boldsymbol{w}_s$ and $\tilde{\boldsymbol{w}}_t$ are the word embeddings corresponding to the position in the co-occurrence matrix $M_{s,t}$, and Ψ is a weighting function:

$$\Psi(x) = \begin{cases} (x/x_{max})^{\xi}, & \text{if } x < x_{max}, \\ 1, & \text{otherwise.} \end{cases}$$

with $x_{max} = 100$ and $\xi = 3/4$. The motivation for $\xi = 3/4$ is empirical, and with this choice of the parameter, the performance of the model happens to improve when compared to $\xi = 1$. In practice, we may have any $0 < \xi < 1$. See Figure 3. The reader may understand this is a way to give more weight to rare word

https://doi.org/10.1017/asb.2019.28 Published online by Cambridge University Press

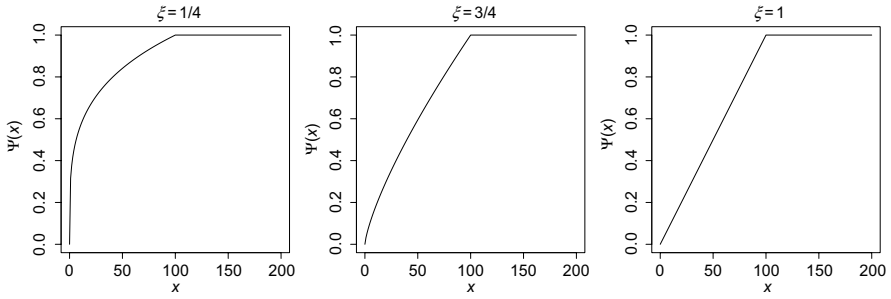

FIGURE 3: Plot of $\Psi(x)$ for different values of ξ, with $x_{max} = 100$.

combinations found in the corpus. Intuitively, the co-occurrence matrix $M_{s,t}$ records how often two words occur together in the training corpus.

The algorithm attempts to find a word embedding for each word that gives a dot product that is close to the transformed co-occurrence matrix entry of the word combination. The resulting word embedding gives a large dot product for those combinations of words that have a high co-occurrence matrix entry, and a low dot product for those combinations of words that correspond to zero matrix entries. Note that most of the entries of the matrix would be zeros. For additional details on the GloVe algorithm, we refer to the original paper by Pennington *et al.* (2014).

Word embedding matrices obtained this way have numerous applications. In the natural language processing literature, word embeddings are used to construct neural networks that can predict missing words in sentences, or translate sentences in one language to another. Applications in the actuarial science literature is new, best to our knowledge. In this paper, we focus on the application of word embeddings in an insurance claims analysis context, assuming that a good word embedding is given. For this, we utilize the concept of word similarities, as explained in Section 3.2.

3.2. Word similarities

Once the word vectors are obtained using approaches outlined in Section 3.1, explanatory variables can be formed using cosine similarity of words. The cosine similarity between two words $\boldsymbol{a}$ and $\boldsymbol{b}$ with nonzero vector representations is given by

$$\text{sim}_{\cos}(\boldsymbol{a}, \boldsymbol{b}) = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{||\boldsymbol{a}||_2 \cdot ||\boldsymbol{b}||_2}.$$

One may think geometrically, and interpret the cosine between two unit vectors as the dot product of the two vectors. The dot product ranges between -1 and 1, with 1 indicating identical vectors and -1 indicating two vectors that point in the opposite direction. Now consider two phrases $D_1 = (\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_S)$ and

https://doi.org/10.1017/asb.2019.28 Published online by Cambridge University Press

$D_2 = (\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_T)$. Hence, a phrase is simply a sequence of words. Goldberg (2017) suggests using the similarity metric

$$\text{sim}^*_{\cos}(D_1, D_2) = \sum_{s=1}^{S} \sum_{t=1}^{T} \text{sim}_{\cos}(\boldsymbol{a}_s, \boldsymbol{b}_t).$$

Essentially this metric assumes the vectors within a phrase can be added to form a vector representing the entire phrase. We tried using this metric, and discovered that the results could be improved using a different metric. Hence, in this paper, we use the similarity metric

$$\text{sim}_{\cos}(D_1, D_2) = \max_{s=1,\ldots,S} \left(\max_{t=1,\ldots,T} \left(\text{sim}_{\cos}(\boldsymbol{a}_s, \boldsymbol{b}_t) \right) \right).$$

Thus, the cosine similarity between two phrases corresponds to the maximum cosine similarity between any two words found within the phrases. In particular, the similarity between a single word $\boldsymbol{a}$ and $D = (\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_S)$ is given by

$$\text{sim}_{\cos}(\boldsymbol{a}, D) = \max_{s=1,\ldots,S} \left(\text{sim}_{\cos}(\boldsymbol{a}, \boldsymbol{b}_s) \right).$$

Defining the features this way is equivalent to max-pooling the features of a one-dimensional convolutional neural network (CNN). Max-pooling tends to work better than alternatives such as average pooling, as explained in page 120 of Chollet and Allaire (2018). When used with a single word $\boldsymbol{a}$, the similarity will be 1 if the particular word appears in the claim description. Hence, in some sense, the similarity can be thought of as a detector of whether a particular word or concept appears in the claim description. This provides more interpretability, as it is a generalization of indicators for word appearance. In Section 4, we demonstrate that using cosine similarities result in an improvement in the predictive accuracy in the claims classification task, compared to when word appearance indicators are used.

Note that we are not assuming that every word appearing in the LGPIF would appear in the word embedding matrix. Those words not appearing in the word embedding matrix could not be used as a word element in a description nor a search key under the current framework of our paper. Hence those words not found in the word embedding matrix are dropped from the model.

4. Applications

In this section, we demonstrate how the explanatory variables extracted from textual data can be used in specific models. We provide two examples: an example in claims classification and an example in risk mitigation. In both applications, the GAM framework is used as an underlying theme. We selected the GAM framework for its flexibility in capturing potential nonlinear effects of the cosine similarities. Word vectors are used in order to improve the

https://doi.org/10.1017/asb.2019.28 Published online by Cambridge University Press

classification results in all examples. These applications specifically use short textual descriptions of insurance claims, which may be found in initial reports of the claims to an insurance claims department. For long textual descriptions or claim adjuster notes, other methods such as RNNs using LSTM cells, or CNNs with multiple layers, may be preferred. For more details regarding neural network approaches to textual data analyses, see Goodfellow *et al.* (2016) and Goldberg (2017). Yet, the simplicity of our approach may make it attractive for actuaries working with simple textual descriptions of claims.

4.1. Claims classification

4.1.1. *Model*

In practice, an insurance claims manager would have to classify given claims based on their properties. This task may be supported with a claims classification engine, which would take the claim description as its input, and output the correct claim category. This motivates our first application of word embedding models, which is the classification of insurance claims into discrete categories.

The engine would be trained over a training data set, and validated over a test data set. Within the training data set, we assume category J_i and description D_i are observed for each claim i. Thus, the sample consists of observations $\{(J_1, D_1), \ldots, (J_n, D_n)\}$. Note that $D_i = (w_{i1}, w_{i2}, \ldots, w_{iq(i)})$, a description consisting of $q(i)$ words, where $q(i)$ is the number of words consisting the ith description. We assume that $n = n_0 + n_1 + \cdots + n_{j_{max}}$; thus, we imagine that nine samples each of size $n_0, n_1, \ldots, n_{j_{max}}$ are stacked to form the given sample of size n. In this paper, $j_{max} = 8$. The claim categories for the training sample and test sample are shown in Table 2. Thus, the response variable J_i takes on the values $0, \ldots, j_{max}$.

A GAM framework can be thought of as a generalized linear model (GLM) with predictors involving smooth functions. See Hastie and Tibshirani (1990) and Wood (2017). The explanatory variables, $\boldsymbol{u}_i$, are defined by

$$u_{i,k} = \text{sim}_{\cos}(w_k, D_i) \cdot I\big(\text{sim}_{\cos}(w_k, D_i) \geq \epsilon\big), \quad k = 1, \ldots, K,$$

where $\epsilon = 0.2$ is used, and w_k is the kth word used in the model and D_i is the description of the ith claim. For the model, we use a multinomial specification. We made this choice (as opposed to other classification methods), based on the fact that the multinomial model provides a framework that is easily generalizable to a GAM model. In this case, the probability of observing a specific peril type j is given by

$$\text{f}_i(j) = \begin{cases} 1 \Big/ \left(1 + \sum_{j'=0}^{j_{max}} \exp\left(\psi_{j',i}\right) \right), & \text{for base peril type } j = 0, \\ \exp\left(\psi_{j,i}\right) \Big/ \left(1 + \sum_{j'=0}^{j_{max}} \exp\left(\psi_{j',i}\right) \right), & \text{for peril type } 1 \leq j \leq j_{max}, \end{cases}$$

https://doi.org/10.1017/asb.2019.28 Published online by Cambridge University Press

TABLE 2

CLAIM CATEGORIES FOR TRAINING AND VALIDATION DATA SETS.

	Misc. ($J_i = 0$)	Vandalism ($J_i = 1$)	Fire ($J_i = 2$)	Lightning ($J_i = 3$)	Wind ($J_i = 4$)	Hail ($J_i = 5$)	Vehicle ($J_i = 6$)	Water (NW) ($J_i = 7$)	Water (W) ($J_i = 8$)	Total
Training	362	1774	171	832	296	76	852	202	426	4991
Test	103	310	46	123	107	18	227	67	38	1039

TABLE 3

WORDS USED FOR CLASSIFICATION.

vandalism	*fire*	*lightning*	*wind*	*hail*	*vehicle*	*water*

with

$$\psi_{j,i} = \alpha_j + \sum_{k=1}^{K} \phi_{j,k}(u_{i,k}),$$

where α_j is an intercept, and $\phi_{j,k}(k = 1, \ldots, K)$ may be smooth functions of the covariate, and K is the number of words used in the model. We denote the base peril type as miscellaneous claims, and call it $j = 0$. If we assume the functions are linear so that estimation time could be saved, then we have

$$\phi_{j,k}(u_{i,k}) = u_{i,k}\beta_{j,k}, \quad k = 1, \ldots, K$$

and hence

$$\mathrm{f}_i(j) = \begin{cases} 1 \Big/ \left(1 + \sum_{j'=0}^{j_{max}} \exp\left(\alpha_{j'} + \boldsymbol{u}_i'\boldsymbol{\beta}_{j'}\right)\right), & \text{for base peril type } j = 0, \\ \exp\left(\alpha_j + \boldsymbol{u}_i'\boldsymbol{\beta}_j\right) \Big/ \left(1 + \sum_{j'=0}^{j_{max}} \exp\left(\alpha_{j'} + \boldsymbol{u}_i'\boldsymbol{\beta}_{j'}\right)\right), & \text{for peril type } 1 \le j \le j_{max}. \end{cases}$$

Here, $\boldsymbol{\beta}_j$ are K-dimensional coefficients. Note that with this choice of $\phi_{j,k}$, the GAM model becomes the GLM model. This simplification is useful for reducing the computation time, when a large number of explanatory variables are included in the model.

Table 3 shows the $K = 7$ words used in the model. The reader may imagine projecting each claim description D_i onto a space represented by $K = 7$ axes. In this paper, the feature words have been selected by a human expert with a good understanding of the data set. In practice, the most frequent words found in the claim descriptions may be used as the key words. Abbreviated words are valid choices. In the claims classification problem, our goal is to construct an engine that gives the best possible classification result, and the focus is less on the interpretability of the coefficients resulting from the estimation. For applications such as that found in Section 4.2, the interpretability is more important, and hence the feature words should be selected more carefully by a human expert using the engine.

The reader may be curious why a censoring of the cosine similarities is needed. The reason is that cosine similarities smaller than the threshold is basically noise. One way to understand this phenomenon is to imagine a search engine returning results on the Internet. The first few results are highly related

https://doi.org/10.1017/asb.2019.28 Published online by Cambridge University Press

TABLE 4

CONFUSION MATRIX FOR MULTINOMIAL MODEL.

Actual category	Predicted category									
	Misc.	Vandalism	Fire	Lightning	Wind	Hail	Vehicle	Water (NW)	Water (W)	Total
Misc.	24	33	1	1	0	0	39	0	5	103
Vandalism	17	267	0	0	2	0	23	0	1	310
Fire	0	2	18	3	0	0	20	2	1	46
Lightning	3	1	0	114	0	1	3	0	1	123
Wind	4	4	2	3	88	2	1	0	3	107
Hail	0	0	0	0	0	17	1	0	0	18
Vehicle	31	5	4	0	0	0	182	2	3	227
Water (NW)	2	4	0	0	0	0	5	4	52	67
Water (W)	5	1	0	0	4	0	0	1	27	38
Total	86	317	25	121	94	20	274	9	93	1039

to the search string that has been entered, but as one goes down the list, more and more irrelevant results may be observed. These junk results tend to add noise to the regression result, and hence we censor the cosine similarities with a threshold $\epsilon = 0.2$, where the choice of ϵ is an empirical question. Figure 4 in Section 4.1.2 shows the classification accuracy as a function of the threshold ϵ.

4.1.2. *Result*

The GLM model is fit using the training sample, and tested on the validation sample. Table 4 shows the confusion matrix for the result of the classification in the validation sample. For each i, the category with the largest prediction score is chosen as the predicted category. The reader may observe that the classification result is good, based on the fact that the diagonal entries are the largest. The classification for non-weather-related water damages suffered somewhat, and the confusion is primarily due to the fact that differentiating them with weather related water damages is a difficult task. Similar observations may be made for other peril types as well. Sometimes the allocation to one category may not be entirely clear to the engine, and in this case misclassification may happen. For instance, a fire due to lightning may be incorrectly categorized as fire. Including more words in the model would presumably result in a better classification engine.

The analysis of classification accuracy is often performed by the receiver operating characteristic (ROC) curve in the binary classification problem. The ROC curve is created by plotting the true positive rate against the false positive rate at various thresholds. The area under the curve is often used as a measure of how well the classification engine is performing. In the multiple class problem, numeric measures such as the average accuracy, error rate, and precision

https://doi.org/10.1017/asb.2019.28 Published online by Cambridge University Press

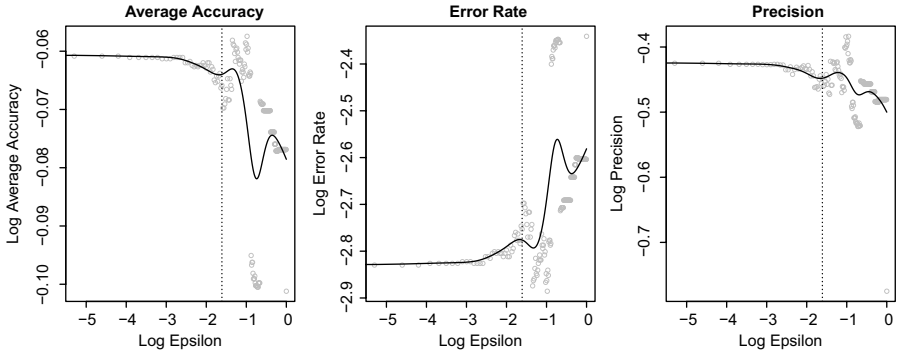

FIGURE 4: Average accuracy, error rate, and precision in log scale.

are better suited for analyzing the classification accuracy. These quantities are defined by

$$\text{Average Accuracy} = \frac{1}{j_{max}+1}\sum_{j=0}^{j_{max}} \frac{tp_j + tn_j}{tp_j + fn_j + fp_j + tn_j},$$

$$\text{Error Rate} = \frac{1}{j_{max}+1}\sum_{j=0}^{j_{max}} \frac{fp_j + fn_j}{tp_j + fn_j + fp_j + tn_j},$$

$$\text{Precision} = \frac{1}{j_{max}+1}\sum_{j=0}^{j_{max}} \frac{tp_j}{tp_j + fp_j},$$

where tp_j is the number of true positives, tn_j is the number of true negatives, fp_j is the number of false positives, fn_j is the number of false negatives; see Sokolova and Lapalme (2009). Figure 4 shows the average accuracy, error rate, and precision. With $\epsilon = 0.2$, the average accuracy is 93.62%, the error rate is 6.37%, and the precision is 63.9%. We tested if these numbers changed as the threshold ϵ changed.

Figure 4 shows that the average accuracy, error rate, and precision changes as ϵ is altered by 0.005. The solid lines show degree 10 splines fit to the experiment data. Notice that when $\epsilon = 1$ (or in other words when $\log \epsilon = 0$), the accuracy and precision reduces, and the error rate increases significantly. This is precisely the case when the cosine similarities boil down to indicator variables of whether the particular words are found in the claim descriptions (in other words, when there exists an embedding for at least one word in the description that is a constant multiple of the keyword). Figure 4 is evidence that the model matrix based on cosine similarities is outperforming that based on indicators. Our choice $\epsilon = 0.2$ is shown as a vertical dotted line at $\log(0.2) = -1.609$. We emphasize once more that the choice of ϵ is an empirical question, as the classification accuracy is not influenced much by the threshold. Moreover, since the graph is stable on the left-hand side, $\epsilon = 0$ may also be a reasonable choice, if

https://doi.org/10.1017/asb.2019.28 Published online by Cambridge University Press

prediction alone is the concern. Yet, for interpretation of the coefficients, we have chosen a nonzero threshold.

4.2. Risk mitigation

4.2.1. *Model*

In order to understand the factors that relate with high losses, we assume a sampling frame in the following form: we assume that the loss amount, the category of the loss, and the description of the loss are observed. While J_i was a random variable in Section 4.1, here we assume $J_i = j$ is fixed. Also, we assume for each category, n_j losses are observed.

Let j be the category of the loss, and let $Y^*_{j,i}$ be the underlying loss amount, and $D_{j,i}$ the description of the ith loss in the jth category. Thus, the data set consists of distinct samples with observations of the form

$$\begin{gathered}\left\{(Y^*_{0,1}, D_{0,1}), \ldots, (Y^*_{0,n_0}, D_{0,n_0})\right\} \\ \vdots \\ \left\{(Y^*_{j_{max},1}, D_{j_{max},1}), \ldots, (Y^*_{j_{max},n_{j_{max}}}, D_{j_{max},n_{j_{max}}})\right\}.\end{gathered}$$

For a given $0 < \gamma < 1$, responses $Y_{j,i}$ are formed by

$$\begin{aligned}Y_{j,i} &= I(Y^*_{j,i} > q_j(\gamma)), \\ q_j(\gamma) &= \inf\left\{y : P(Y^*_j \le y) \ge \gamma\right\},\end{aligned}$$

where $\gamma = 0.5$ is used to obtain the median for each category. For our analysis, the empirical quantile has been used for $q_j(\gamma)$. In other words, for any given loss, $Y_{j,i}$ is an indicator of whether the observed loss $Y^*_{j,i}$ is above the 50th percentile of losses in that category of loss. Any other quantile could have been used. For instance, the 95th percentile could have been used. Different quantiles would give different stories, because the definition of a large claim would be different for each case. The 50th percentile has been arbitrarily selected for demonstration.

We analyzed the losses for vandalism, fire, wind, vehicle, and the two water damage categories, omitting lightning, hail, and miscellaneous losses from the analysis because for the latter three it was difficult to identify keywords that correspond to large claims. Detailed results are shown for the vandalism peril type only, in order to limit the number of pages of the paper. The question is, whether we can understand the factors that are related to response values $Y_{j,i} = 1$, corresponding to high losses, through text analysis.

Using the word similarity metric described in Section 3.2, we can create variables for risk measures of interest. Consider a specific description of a claim in a given category. Suppose a modeler is interested in creating a risk metric corresponding to a word $w_{j,k}$, $k = 1, \ldots, K_j$ (K_j: number of risk metrics in category j). Suppose an insurance claim is described by the phrase D_i, for the ith

https://doi.org/10.1017/asb.2019.28 Published online by Cambridge University Press

TABLE 5

SUMMARY STATISTICS OF EXPLANATORY VARIABLES (VANDALISM MODEL).

	Min.	*Median*	*Mean*	*Max.*
laptop	0.000	0.000	0.130	1.000
graffiti	0.000	0.520	0.424	1.000
window	0.000	0.246	0.348	1.000
shelter	0.000	0.221	0.187	1.000
pool	0.000	0.260	0.209	1.000

observation, $i = 1, \ldots, n$, where n is the number of claims found in the data set. We create a variable by

$$u_{j,i,k} = \text{sim}_{\cos}(w_{j,k}, D_i) \cdot I\big(\text{sim}_{\cos}(w_{j,k}, D_i) \geq \epsilon_j\big)$$

for a threshold ϵ_j. In this paper, $\epsilon_j = 0.2$ is chosen for each j. Thus, $u_{j,i,k}$ is the cosine similarity between a search word $w_{j,k}$ and the description of the claim D_i, with a censoring below a certain similarity level. For example, the modeler may be interested in the risk metric *graffiti*. The modeler may be interested in the relationship between this risk metric, and a response variable of interest, say the magnitude of loss. If the metric has a high correlation with large losses, then attention should be given to the particular risk metric in order to mitigate the risk inherent in this metric. In this case, the modeler may create a column vector for *graffiti*. For a particular response, say the likelihood of a high vandalism claim, a modeler may have a set of risk metrics of interest, say *graffiti, laptop, window, shelter, pool* (In this case, $K_j = 5$). This gives a matrix of K_j explanatory variables, which can be used in standard regression models. Table 5 summarizes the explanatory variables used for the vandalism model.

For category j (this section will focus on the vandalism category in particular), given a claim i, the GAM model for each peril type can be specified as

$$g\left\{\mathbb{E}(Y_{j,i})\right\} = \alpha_j + \sum_{k=1}^{K_j} \phi_{j,k}(u_{j,i,k}),$$

where

$$g(\mu_{j,i}) = \log\left(\frac{\mu_{j,i}}{1-\mu_{j,i}}\right),$$

and α_j is the coefficient for the intercept, and $\phi_{j,k}$ are smooth functions of the covariates $u_{j,i,k}$, subject to the constraint such that $\sum_{i=1}^{n} \phi_{j,k}(u_{j,i,k}) = 0$. In other words, a logit link is used since $Y_{j,i}$ is a binary variable taking on the value of 1 or 0. We have

$$\mu_{j,i} = \frac{\exp\left(\mathbf{x}'_{j,i}\boldsymbol{\beta}_j\right)}{1+\exp\left(\mathbf{x}'_{j,i}\boldsymbol{\beta}_j\right)}, \quad V(\mu_{j,i}) = \mu_{j,i}(1-\mu_{j,i}).$$

https://doi.org/10.1017/asb.2019.28 Published online by Cambridge University Press

Let the matrix $\boldsymbol{X}_j$ include transformed columns representing the spline bases for the $\phi_{j,k}$. For this study, a second-order B-spline basis with three terms is used. In this case, the design matrix $\boldsymbol{X_j}$ can be constructed by first forming the matrix

$$\boldsymbol{\Phi}_{j,k} = \begin{bmatrix} B_1^1(u_{j,1,k}) & B_2^1(u_{j,1,k}) & B_3^1(u_{j,1,k}) \\ B_1^1(u_{j,2,k}) & B_2^1(u_{j,2,k}) & B_3^1(u_{j,2,k}) \\ \vdots & \vdots & \vdots \\ B_1^1(u_{j,n,k}) & B_2^1(u_{j,n,k}) & B_3^1(u_{j,n,k}) \end{bmatrix}, \quad k = 1, \ldots K_j,$$

where $B_l^m(u)$ are the B-spline basis functions, presented in Wood (2017). Then the columns are transformed using QR factorization, in order to impose the identifiability constraint. This involves decomposing each vector $\boldsymbol{\Phi}_{j,k}^T \mathbf{1}$ into the form

$$\boldsymbol{\Phi}_{j,k}^T \mathbf{1} = \begin{bmatrix} \boldsymbol{Q}_{j,k,1} & \boldsymbol{Q}_{j,k,2} \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_{j,k} \\ \mathbf{0} \end{bmatrix}$$

and then taking $\boldsymbol{Q}_{j,k,2}$ for $k = 1, \ldots, K_j$ to form the design matrix

$$\boldsymbol{X_j} = \left[\mathbf{1};\ \boldsymbol{\Phi}_{j,1}\boldsymbol{Q}_{j,1,2};\ \boldsymbol{\Phi}_{j,2}\boldsymbol{Q}_{j,2,2};\ \ldots\ \boldsymbol{\Phi}_{j,K_j}\boldsymbol{Q}_{j,K_j,2}\right].$$

This ensures that an intercept is included in the design, and also allows the basis functions $\phi_{j,k}$ to satisfy $\sum_{i=1}^n \phi_{j,k}(u_{j,i,k}) = 0$. The idea of P-IRLS (Penalized Iteratively Reweighted Least Squares) is that a weight matrix is adjusted each time the algorithm iterates until convergence. The algorithm follows the following steps:

1. Given the current $\boldsymbol{X_j}\boldsymbol{\beta}_j^{[h]}$, calculate the diagonal matrix $\boldsymbol{W}_j^{[h]}$

   $$W_{j,ii}^{[h]} = \left[G_{j,ii}^{[h]2} V\left(\mu_{j,i}^{[h]}\right)\right]^{-1}$$

   and

   $$\boldsymbol{z}_j^{[h]} = \boldsymbol{G}_j^{[h]}\left(\boldsymbol{y}_j - \boldsymbol{\mu}_j^{[h]}\right) + \boldsymbol{X}_j\boldsymbol{\beta}_j^{[h]},$$

   where $\boldsymbol{G}_j^{[h]}$ is a diagonal matrix satisfying $G_{j,ii}^{[h]} = g'\left(\mu_{j,i}^{[h]}\right)$ and $\mu_{j,i}^{[h]} = g^{-1}\left(\boldsymbol{X_j}\boldsymbol{\beta}_j^{[h]}\right)$.
2. Then minimize

   $$\left\| \sqrt{\boldsymbol{W}_j^{[h]}}\left(\boldsymbol{z}_j^{[h]} - \boldsymbol{X}_j\boldsymbol{\beta}_j\right)\right\|^2 + \sum_{k=1}^{K_j} \lambda_{j,k}\boldsymbol{\beta}_j^T \boldsymbol{Q}_{j,k,2}^T \boldsymbol{S}_{j,k}\boldsymbol{Q}_{j,k,2}\boldsymbol{\beta}_j$$

   with respect to $\boldsymbol{\beta}_j$.
3. Repeat Steps 1 and 2 until convergence.

https://doi.org/10.1017/asb.2019.28 Published online by Cambridge University Press

Here, h is the iteration index, and $\boldsymbol{S}_{j,k}$ are matrices designed to penalize the roughness of the smooth functions. In this paper, we used the difference penalty

$$\boldsymbol{S}_{j,k} = \boldsymbol{D}^T\boldsymbol{D}, \quad \text{where} \quad \boldsymbol{D} = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}.$$

The $\lambda_{j,k}$ are selected by generalized cross-validation. Cross-validation is applied to a subset of the training sample corresponding to each peril type. This involves minimizing the generalized cross-validation score

$$\mathcal{V}_j = \frac{n_j \sum_{i=1}^{n_j} (y_{j,i} - \hat{y}_{j,i})^2}{[n_j - tr(\boldsymbol{A}_j)]^2},$$

where $\boldsymbol{A}_j$ is the influence matrix for the jth category. For details of the theory behind cross-validation, we reference Wood (2017). The above procedure is performed for each j, where in our work each of the following peril types are considered: vandalism, fire, wind, vehicle, water (non-weather), and water (weather). GAMs are implemented in the R programming language via packages such as `gam` and `mgcv`. In this paper, parameters are estimated using the `mgcv` R package, implemented by the author of Wood (2017). We chose this package because it provides a convenient interface regarding the choice of basis functions and graphical outputs.

4.2.2. *Result*

In this section, we present the analysis results for the vandalism category. For the vandalism category, explanatory variables *laptop*, *graffiti*, *window*, *shelter*, and *pool* were included in the GAM model. Among these, *laptop* turns out to have a positive relationship with high losses. Figure 5 shows the shape of the smooth functions $\phi_{j,k}, j = 1, k = 1, \ldots, 5$, as the explanatory variable varies from 0 to 1. The shaded regions in Figure 5 illustrate the 95% credible intervals of the smooth functions at each point on the curve.

Notice that data for large values of cosines are scarce, hence the credible interval widens for large values of the explanatory variables. Figure 6 shows the words with highest cosine similarity with *laptop*, which turns out to be positively related with high losses. Presumably, vandalisms and thefts to laptops, computers, and portables turn out to result in relatively high losses within the vandalism category. Note that vandalisms are small frequent losses. Although the loss amounts in this category are small, the frequent nature of the losses may make it worthwhile to mitigate thefts to laptops. The words in Figure 6 are not necessarily found in the training data. They are words found in the word embedding matrices. What we are saying here is that since the word *laptop* has a high correlation with large losses, related words such as *laptops*, *computers*, and *phones* are also suspects for potential high losses, due to their relationship

https://doi.org/10.1017/asb.2019.28 Published online by Cambridge University Press

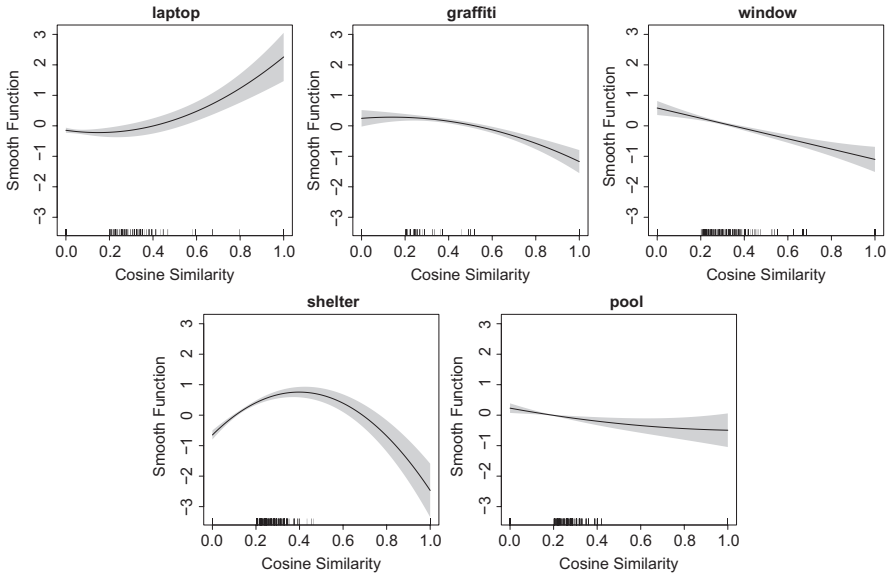

FIGURE 5: GAM model plots for vandalism.

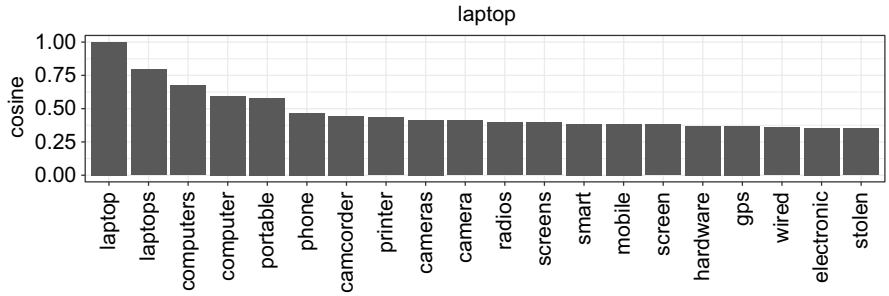

FIGURE 6: Words that relate with high vandalism losses.

to the word *laptop*. This type of chart helps an insurance claims department to learn which type of property to focus on, when mitigating risk.

4.2.3. *Model diagnostic*

Table 6 provides a summary of the GAM models. The χ^2 statistic reported for each smooth function is based on Wood (2013), or pages 305–308 of Wood (2017). Essentially, this tests the hypothesis

$$H_0 : \phi_{j,k}(u) = 0$$

for all u in the range of the cosine similarities for category j. A high p value would indicate that the function $\phi_{j,k}$ is not needed in the model. The smooth functions for vandalism all turn out to be significant, according to Table 6.

https://doi.org/10.1017/asb.2019.28 Published online by Cambridge University Press

TABLE 6

GAM MODEL SUMMARY (VANDALISM MODEL).

	Chi.sq	*p value*
s (laptop)	32.560	0.000
s (graffiti)	39.250	0.000
s (window)	47.430	0.000
s (shelter)	132.930	0.000
s (pool)	9.020	0.001
R-sq. (adj)	0.206	

5. CONCLUDING REMARKS

In this paper, we introduced a framework for incorporating textual data into insurance claims modeling, and considered its applications in claims management processes. An insurance claims representative is responsible for investigating the claim, in order to determine the handling process. In this paper, we explored the use of word similarities as a tool for modeling insurance claims and mitigating insurance risks. Our results demonstrate how text mining technology can be incorporated into a traditional regression analysis. The methodology is applicable in many different areas of applications, where textual data arises. Possible applications of our approach for an insurance risk manager may include the following:

- classification of claims based on textual descriptions of the claims,
- classification of policyholders based on textual descriptions of the policyholders,
- prediction of insurance claims at the claim level,
- prediction of insurance claims at the policyholder level,
- analysis of insurance claims and risk mitigation.

We explored the LGPIF data in the form of case studies to understand the factors that relate with high insurance losses, classify insurance claims, and model the loss amounts using parametric distributions involving covariates derived from textual information. We make some remarks on the current limitations of our framework, where potential improvements can be made.

- Under the current framework, words not found in the word embedding matrix cannot be used in the modeling.
- The threshold ϵ is selected using heuristics by a human expert, under the current framework.
- Because predetermined word embedding matrices are limited to one-grams (single words) at the time the paper is being written, the incorporation of *n*-grams (use of phrases longer than one word as a search key) remains an open question.

https://doi.org/10.1017/asb.2019.28 Published online by Cambridge University Press

- Further linguistic barriers may exist, if the textual descriptions are longer than those appearing in the data set used for this paper. Examples may be polysemy, false friends, and compound words.
- In order to use the proposed method, insurers that focus on specific insurance segments may be constrained to build their own word embedding matrices, as the terms appearing in the claim descriptions may be specific to the field. For example, a medical insurer may find GloVe insufficient, and may need a word embedding matrix trained on medical terms in order to use our proposed approach.

Economic losses due to property damage caused by perils including fire, lightning, wind, hail, or vandalism have vast implications to our society. Understanding the nature of property damage can improve our readiness and contribute to minimizing the losses. We have illustrated a way to help realize this goal using a new analysis method, which, to the best of our knowledge, has not been attempted before in the actuarial literature. We believe our methodology may help broaden the horizon of empirical research, and contribute to the advancement of the understanding of our world and the risks residing within it. In addition, we believe that our approach will improve the claim handling procedures of insurance claims departments. Illustrative R code is available from the authors upon request.

ACKNOWLEDGMENTS

We would like to express our gratitude to the three anonymous reviewers, and the editor, who have carefully read through the paper and provided constructive comments and suggestions to help improve our work.

REFERENCES

CHOLLET, F. and ALLAIRE, J. J. (2018) *Deep Learning with R*. Shelter Island, NY: Manning Publications Co.

FREES, E. W. (2009) *Regression Modeling with Actuarial and Financial Applications*. Cambridge, UK: Cambridge University Press.

FREES, E. W., LEE, G. Y. and YANG, L. (2016) Multivariate frequency-severity regression models in insurance. *Risks*, **2016**(4): 4.

GOLDBERG, Y. (2017) *Neural Network Methods for Natural Language Processing*. San Rafael, CA: Morgan & Claypool Publishers.

GOODFELLOW, I., BENGIO, Y. and COURVILLE, A. (2016) *Deep Learning*. Cambridge, MA: MIT Press.

HASTIE, T., TIBSHIRANI, R. and FRIEDMAN, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Berlin: Springer Science & Business Media.

HASTIE, T. J. and TIBSHIRANI, R. J. (1990). *Generalized Additive Models*. Boca Raton, FL: Chapman and Hall.

KEARNEY, S. (2010). *Insurance Operations*. Malvern, PA: The Institutes.

https://doi.org/10.1017/asb.2019.28 Published online by Cambridge University Press

MANNING, C. D. and SCHUTZE, H. (1999). *Foundations of Statistical Natural Language Processing, 1st Edition*. Cambridge, MA: The MIT Press.
MIKOLOV, T., SUTSKEVER, I., CHEN, K., CORRADO, G. S. and DEAN, J. (2013) Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems 26*: 3111–3119.
PENNINGTON, J., SOCHER, R. and MANNING, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, vol. 2014, pp. 1532–1543.
SOKOLOVA, M. and LAPALME, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, 45:427–437.
WOOD, S. (2013). On *p* values for smooth components of an extended generalized additive model. *Biometrika* **100**, 221–228.
WOOD, S. N. (2017). *Generalized Additive Models: An Introduction with R, Second Edition*. Boca Raton, FL: CRC Press.

GEE Y. LEE (Corresponding author)
Department of Statistics and Probability
Department of Mathematics
Michigan State University
C337 Wells Hall, 619 Red Cedar Rd, East Lansing, MI 48824, USA
E-Mail: leegee@msu.edu

SCOTT MANSKI
Department of Statistics and Probability
Michigan State University
C511 Wells Hall, 619 Red Cedar Rd, East Lansing, MI 48824, USA
E-Mail: manskisc@stt.msu.edu

TAPABRATA MAITI
Department of Statistics and Probability
Michigan State University
C424 Wells Hall, 619 Red Cedar Rd, East Lansing, MI 48824, USA
E-Mail: maiti@stt.msu.edu

https://doi.org/10.1017/asb.2019.28 Published online by Cambridge University Press