

Text Mining Methods Applied to Insurance Company Customer Calls: A Case Study

Xiyue Liao¹ PhD., Guoqiang Chen¹ MS, Ben Ku¹ MS, Rahul Narula¹ BS, and Janet Duncan¹ FCAS, FSA, MAAA.

Abstract: The purpose of this case study is to develop a process for a U.S. personal lines insurance company to improve its customer service, make call center operations more efficient, and reduce costs by analyzing customer calls. Text mining methods such as topic modeling and sentiment analysis are used to study approximately 10,000 non-claim customer calls from 2016. Results show the most frequent topics of calls and how customer sentiment differs between topics, which will allow the company to adjust their customer service accordingly.

key words: insurance, customer calls, text mining, classification, topic modeling, sentiment analysis

1. Background and Objective

Text mining is a useful tool that can be used in many industries to turn raw unstructured information into useful structured data for analysis. Text mining has become increasingly important in insurance because claim adjuster files are full of narrative notes that traditionally have not been used in any aggregate analyses. Text mining can turn adjusters' notes into useful data that can be analyzed by actuaries, claim adjusters and underwriters. Although the data underlying our analysis comes from an insurance company, the application underlying our paper is not a traditional one for insurance companies. Our application is a more general application that could be used by many service industries (including insurance companies). However, as a case study, the same approach and methods could be used for other text mining applications such as claim adjuster file notes.

A U.S. personal lines insurance company covers a wide range of financial services regarding automotive and homeowner's insurance. In this paper, we present analysis of information from customer calls using text mining methodologies and classification techniques. Our objective is to classify calls and process them more efficiently, ultimately saving the insurer time, resources, and money. Specifically, we use a variety of text mining methods to identify topics customers are

¹ Department of Statistics and Applied Probability, University of California, Santa Barbara.
Address correspondence to Xiyue Liao, Ph.D., Department of Statistics and Applied Probability, University of California, Santa Barbara, CA 93106. E-mail: liao@pstat.ucsb.edu

calling about, analyze call sentiments, and find cues which the company can use to lead customers to their online payment portal.

2. Introduction to Text Mining in R

All analysis was performed within the open source statistical software **R**. The **R** package **tm** (Feinerer et al 2008) provides a framework of commonly used text mining techniques. We used this package to create a collection of the calls, known as a corpus, and preprocessed customer calls held within the corpus, which included removing punctuation, changing text case, stemming, etc. Next, we used functions in the **tm** package to build a text document matrix for all calls for further computation and analysis. Visualization such as word cloud (Fellows 2014) and histograms of important word frequencies were utilized as a way to derive a general idea of what the customer calls were about. Details are included in Section 3. The **topicmodels** package (Grün and Hornik 2011) allowed us to fit topic models based on the text document matrix created by the **tm** package. The **tidytext** (Robinson and Silge 2016) package was used along with the **topicmodels** package to create topic charts. Text mining results are included in Section 4. Finally, we used the **SentimentAnalysis** (Feuerriegel and Proellocks 2018) package to find the average sentiment score in each call and also the sentiment distribution among the identified topics. Sentiment results are included in Section 5.

3. Description of Dataset

The insurance company received approximately one million non-claim customer calls in total during 2016. Our data was split amongst 11 Excel files of roughly 10,000 calls each. Each call is fit in one cell and there is one call per row. This dataset is unique in that it only contains one variable of text data. Additionally, all numbers (policy numbers, zip codes, phone numbers, etc.) were removed from the dataset to protect customer confidentiality.

Because all calls were transcribed using NICE software² with no human involvement, there are a variety of challenges to overcome. Words are often transcribed incorrectly, and it is often difficult to distinguish between whether the customer or representative is speaking in datasets which lack

² NICE is a company providing speech analytics software which is used for call transcription.

explicit characters to signify a change in speaker. Consequently, careful measures must be taken during analysis to ensure accuracy of results.

The number of words per call ranges from 1 to 7,442 words, with the mean being 679 words and the standard deviation being 751 words. A histogram showing the distribution of number of words per call is presented in Figure 1. Bins are of 100 words.

The histogram in Figure 1 is for all calls, but it is important to note that we have primarily been working with one data file (comprised of approximately 10,000 calls). Because calls were arranged randomly among the data files, analyzing multiple files provided indistinguishable results from analyzing only one file, and we felt comfortable proceeding with our analysis in this manner.

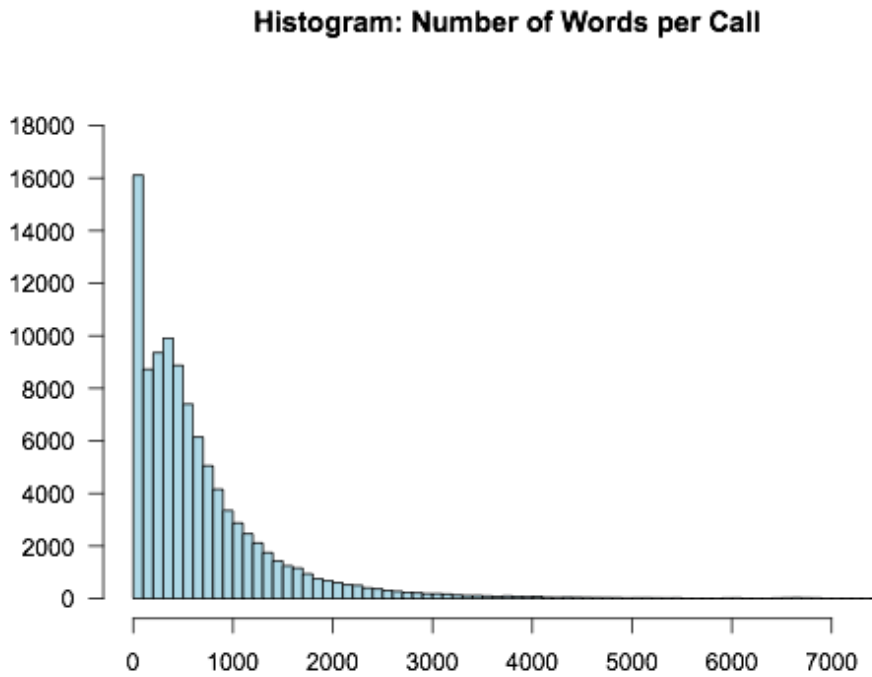


FIGURE 1. Histogram of Number of Words per CALL

4. Methodology

4.1 Preprocessing

We cleaned and prepared the data before beginning our analysis. First, we imported the customer calls from an Excel file into R and used the **tm** package to create a data structure containing these calls or documents, which is referred to as a *corpus*.

Next, we preprocessed calls held within the corpus. Common text mining preprocessing techniques we used include changing text case, removing punctuation, removing stop words, and stemming words.

Changing text case: Converting all text to a single case is necessary as the same word may be counted separately if characters are of different cases. For example, “Text” and “text” carry the same significance but would be counted differently due to the capitalization.

Removing punctuation: Similarly, punctuation must be removed as “data-mining,” “data mining,” and “data mining.” would all incorrectly be considered separate words.

Removing stop words: Stop words are words which are considered filler and hold no significance. These words clutter the dataset and ultimately slow analysis without providing much, if any, usefulness. All in all, we removed 5,375 of approximately 15,000 unique words from the 10,000 calls. Stop words were determined in the following ways:

1. **Standard stop words:** The **tm** package includes a set of common stop words which are easily removed. We filtered through the 488 words provided and determined all stop words should be removed for our analysis. Some of these stop words include: “among,” “might,” and “were.”

2. **Low-frequency words:** Additionally, we removed words which appeared infrequently in our dataset as these words hold no predictive power. Removing them also reduces the size of the dataset significantly, facilitating analysis.

3. **Manually selected words:** Even after using the aforementioned techniques to remove thousands of stop words, our data set was still cluttered. Thus, we manually sifted through the remaining words and determined which were necessary and which could be discarded. Though this process was time consuming, it was necessary and dramatically improved results.

Stemming: “Play,” “playing,” “player,” and “played” all convey the same meaning for text mining purposes, but are technically different words. Thus, it is necessary to stem words, or bring them back to their roots. In this example, each of those words would count as “play” for purposes of our analysis. This method is not infallible due to words with irregular conjugations, but is still necessary to properly prepare the data.

4.2 Exploratory Methods

4.2.1 Term Document Matrix

After preprocessing the data set, we built a term document matrix for the calls held in the corpus. The rows of a term document matrix represent documents (calls) and columns represent all the words in the corpus. Each cell within the matrix contains the frequency of each word for a particular document (call). For example, if ‘car’ appeared 5 times in document three, then the term document matrix records 5 in the intersection of row “document three” and the column for the word ‘car’. We also sorted words by importance rather than frequency. To do this, we applied the method “term frequency inverse document frequency” (TF-IDF), i.e., a weighting scheme which gives words importance as they appear more often, but also punishes words which appear in many documents. (Blei and Lafferty 2009) For example, “insur” (insure/insurance/insured, etc.) is a high-frequency word, but because it appears in almost every single call, it is not discriminative and hence is heavily punished by the TF-IDF method. In other words, it is difficult to sort “insur” into a specific topic because it can apply to a myriad of contexts.

4.2.2 Word Clouds

With the term document matrix, we further created word clouds, which are visual representations of word frequency or importance and are often used in preliminary text mining analysis. The word cloud in Figure 2 was based solely on word frequency and was created before preprocessing, and thus contains all words. Many of these words such as “okay,” “just,” and “that” are not helpful in identifying useful information or patterns. The word cloud in Figure 3 was obtained after preprocessing and it is much more informative than the word cloud in Figure 2.

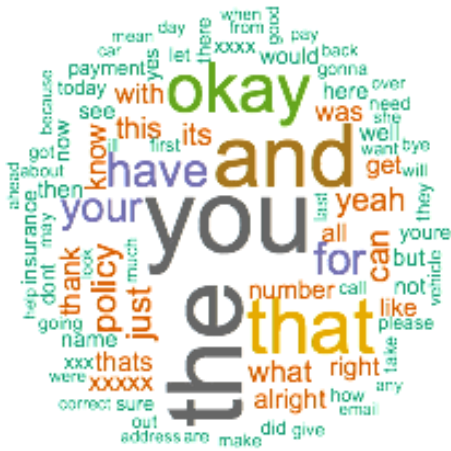


FIGURE 2. Word Cloud before Preprocessing



FIGURE 3. Word Cloud after Preprocessing

4.2.3 Frequency Histograms

Like word clouds, histograms can also be used to compare word frequencies and importance. The two histograms in Figures 4 and 5 are again for the data before and after preprocessing and shifting from raw frequency to weighting by TF-IDF. We can see a dramatic difference in the words that are output in both cases; thoroughly cleaning the data provides more valuable information.

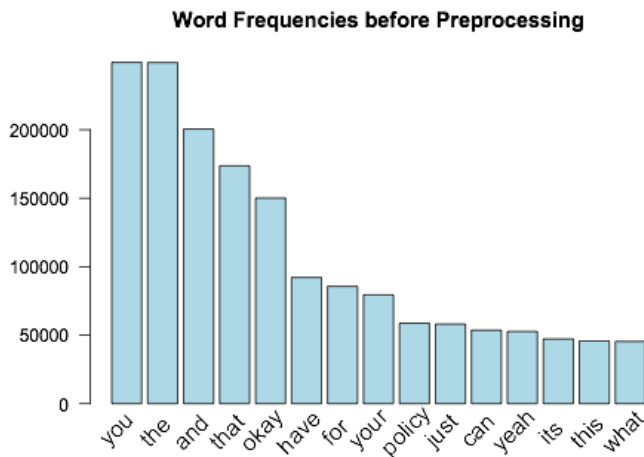


FIGURE 4. Frequency Histogram before Preprocessing

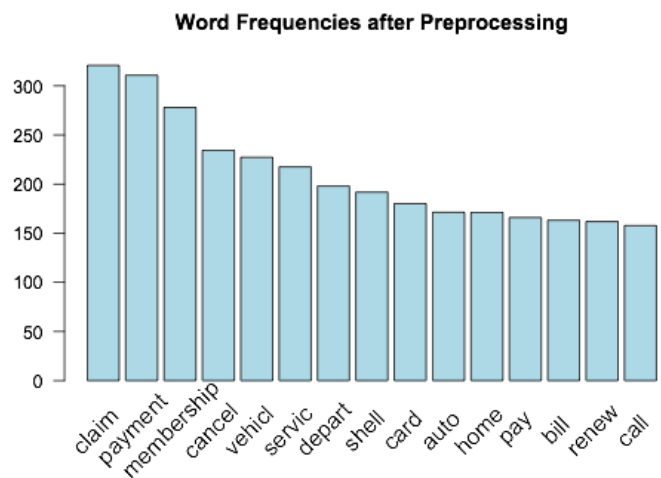


FIGURE 5. Frequency Histogram after Preprocessing

4.3 Topic Modeling

4.3.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) (Blei et al. 2003b) is a popular topic modeling method used to discover underlying topics in documents. Specifically, it is a generative statistical model which allows sets of observations to be explained by unobserved or latent groups that explain why some parts of the data are similar. It assumes that each document is a mixture of topics and each word's creation can be attributed to one of these topics. It is a Bayesian hierarchical model assuming that each topic has a Dirichlet distribution over the vocabulary in a corpus, and each word in a document is drawn from one topic assuming topic proportions in the document again following a Dirichlet distribution. Under the Dirichlet distribution, topics within a document are not correlated. See Section 3 in Blei et al. (2003b) and Section 2 in Grün and Hornik (2011) for more details.

The LDA function in the **topicmodels** package implements two algorithms to identify the latent topics in documents. The two algorithms are: variational expectation-maximization algorithm (VEM) and an algorithm using Gibbs sampling. Results using these two methods are shown later in this section.

The first step in performing LDA is to specify the number of topics (k) to be found in the corpus. There is no exact number of topics to search for in each situation, and reading through the calls and brainstorming allows one to capture the subtleties in language which quantitative measurements ignore. So k must be determined both by using statistical methods and by reading through the data. Four metrics in the **ldatuing** package (Nikita 2016) were used to help us determine the number of topics: **Arun2010** (Arun et al. 2010), **CaoJuan2009** (Cao et al. 2009), **Deveaud2014** (Deveaud et al. 2014), and **Griffiths2004** (Griffiths and Steyvers 2004). The general idea behind these methods is to choose the number of topics in a corpus to maximize the dissimilarity among topics, i.e., to make topics as distinguishable as possible. **The optimal number of topics (k) minimizes the metrics Arun2010 and CaoJuan2009, and maximizes the metrics Deveaud2014 and Griffiths2004.**

Figure 6 shows three metrics used to find the ideal value of k for the VEM algorithm and Figure 7 shows four metrics to find k for the Gibbs sampling algorithm. Although different metrics suggested we continue to inflate k past 15, we found that results were difficult to comprehend at that

point, and we ultimately decided to proceed with twelve topics as they provided a balance between specificity and applicability.

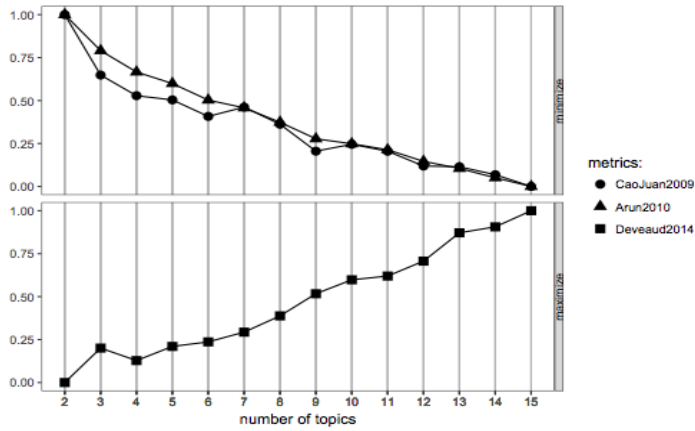


FIGURE 6. Metrics to Find the Ideal Number of Topics (k) for the VEM Algorithm

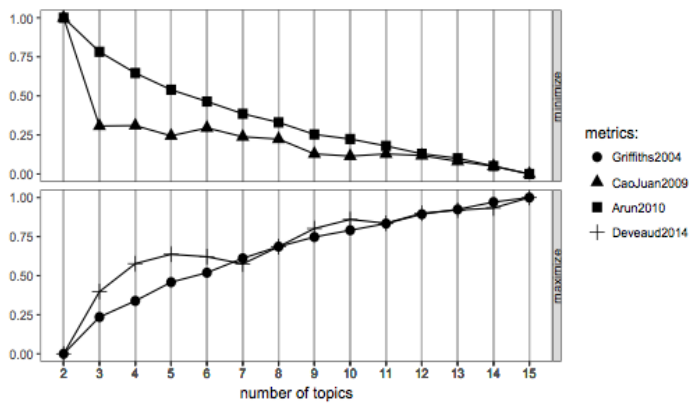


FIGURE 7. Metrics to Find the Ideal Number of Topics (k) for the Gibbs Sampling Algorithm

4.3.1.1 VEM Topics

We use the **tidytext** package to create the topic chart in Figure 8. With $k = 12$, we see the ten words under each topic with the highest beta, or probability of being generated by that respective topic. From these words, we must generate a label for the topic. For example, topic 9 contains the words “card,” “bank,” “account,” and “process,” so we can surmise that this topic concerns payments. This process required considerable judgement as well as experience with insurance company processes and services. Looking at the words under each topic, we see that there is much overlap – this is expected because all documents cover very similar subjects, but this also poses issues when accurately categorizing calls. As k increases, we can expect increased overlap but possibly have new topics emerge. Reducing k would give us more identifiable topics, but may also inappropriately merge some.

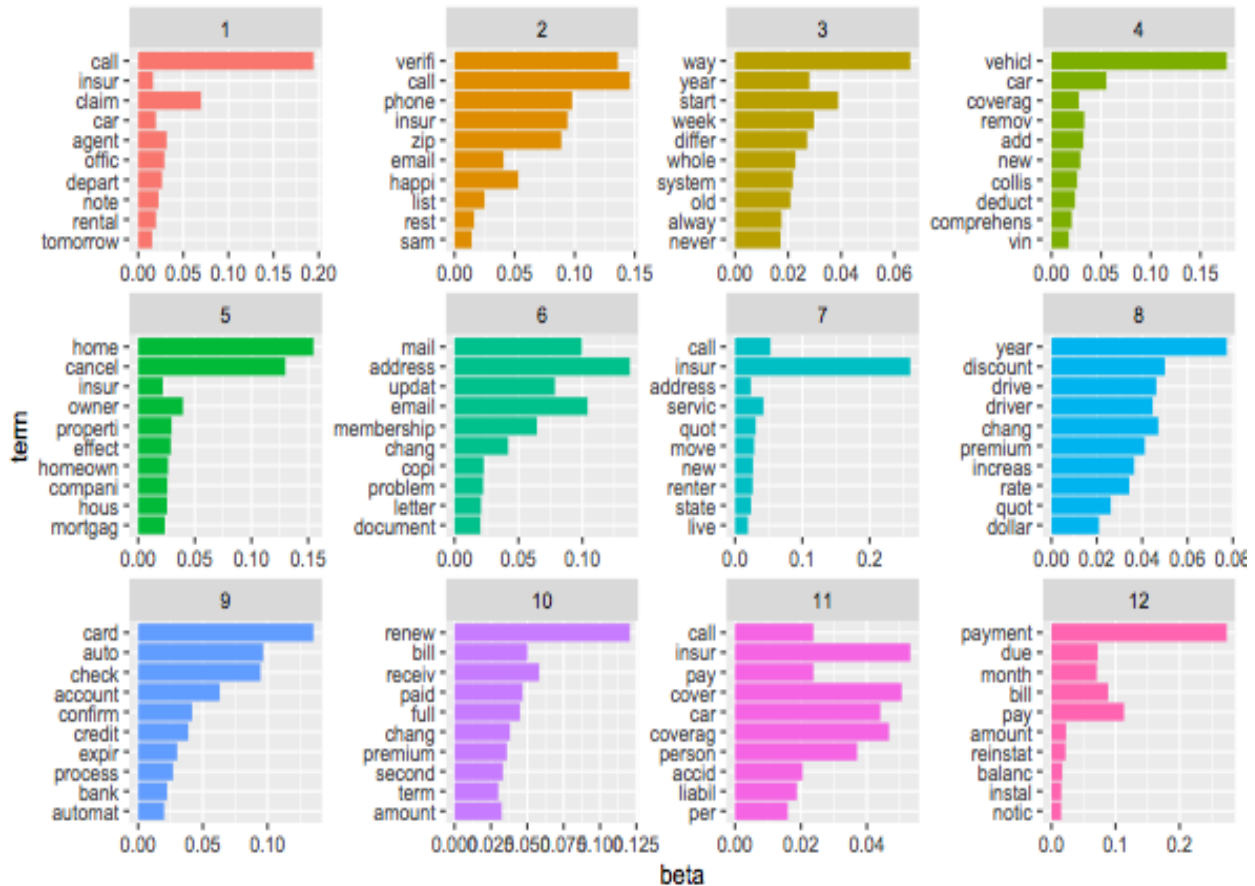


FIGURE 8. Words with Highest Beta when Identifying VEM Topics with $k = 12$

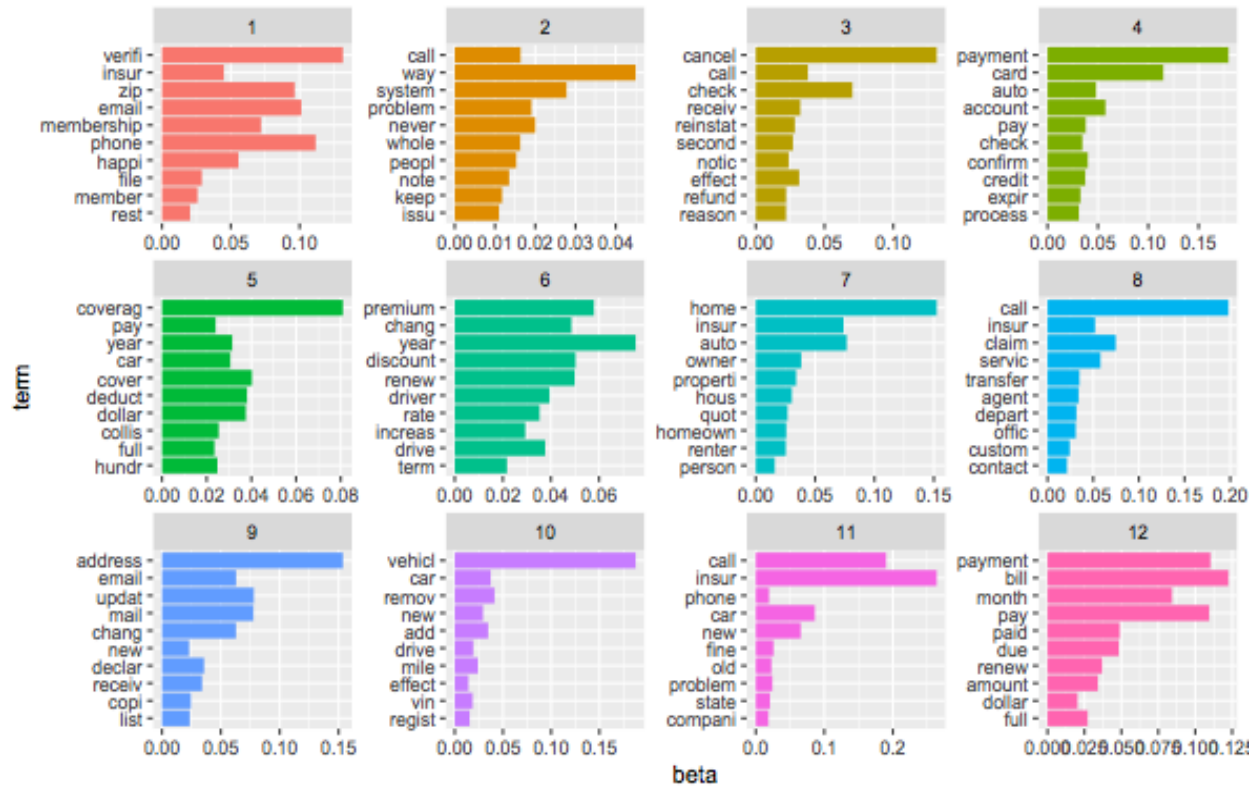


FIGURE 9. Words with Highest Beta when Identifying Gibbs Topics with $k = 12$

4.3.1.2 Gibbs Topics

A topic chart using the Gibbs sampling method is shown in Figure 9. Using the Gibbs sampling method produced very similar topics as VEM, and thus we used the VEM topics for all LDA analysis going forward.

4.3.1.3 Selected Topics

After tuning parameters and sifting through results, we found that when k was 12, the most apparent topics were “Add/Remove Vehicle,” “Administrative,” “Auto Policy,” “Billing,” “Change/Renew,” “Cancellation,” “Coverage,” and “Premium/Rates.” The remaining four topics either appeared to be combinations of the aforementioned topics or contained no discernible theme.

Table 1 shows the proportion of calls which fell under our primary topics. “[Uncategorized]” calls were those that fell under one of the four equivocal topics.

TABLE 1
VEM Topics

VEM Topic	Proportion
[Uncategorized]	29.6%
Billing	19.8%
Add/Remove Vehicle	9.7%
Cancel	9.3%
Admin	8.3%
Change/Renew	7.3%
Premium/Rates	6.0%
Coverage	5.0%
Auto Policy	5.0%

4.3.2 Correlated Topic Models

Correlated topic modeling (CTM) (Blei and Lafferty 2007) is an extension of LDA modeling. The difference is that LDA imposes the Dirichlet distribution on topic proportions in a corpus, which excludes possible correlation between presence of topics, while CTM uses the logistic normal distribution instead, which allows correlation between topics.

The CTM function in the **topicmodels** package was used to fit a CTM model with the VEM algorithm. As shown in Figure 10, when searching for twelve topics, we could categorize “personal information verification,” “add/remove vehicle,” “billing,” “policyholder services” “homeowner’s insurance,” and “policy changes.” Some topics were unclear or overlapped with others, but overall, topics derived from CTM and LDA were quite similar.

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7
[1,]	"call"	"year"	"coverag"	"insur"	"renew"	"vehicl"	"vehicl"
[2,]	"new"	"way"	"cover"	"address"	"discount"	"coverag"	"driver"
[3,]	"car"	"pay"	"home"	"cancel"	"premium"	"collis"	"drive"
[4,]	"insur"	"peopl"	"properti"	"email"	"chang"	"car"	"remov"
[5,]	"phone"	"car"	"person"	"mail"	"year"	"deduct"	"insur"
[6,]	"bill"	"month"	"insur"	"call"	"increas"	"comprehens"	"mile"
[7,]	"way"	"money"	"deduct"	"updat"	"rate"	"add"	"car"
[8,]	"pay"	"keep"	"liabil"	"verifi"	"term"	"vin"	"licens"
[9,]	"paid"	"drive"	"damag"	"home"	"dollar"	"new"	"year"
[10,]	"year"	"never"	"year"	"receiv"	"pay"	"dollar"	"add"
	Topic 8	Topic 9	Topic 10	Topic 11	Topic 12		
[1,]	"call"	"membership"	"call"	"payment"	"insur"		
[2,]	"check"	"insur"	"claim"	"pay"	"call"		
[3,]	"start"	"auto"	"insur"	"bill"	"problem"		
[4,]	"issu"	"member"	"servic"	"card"	"car"		
[5,]	"chang"	"card"	"agent"	"due"	"fine"		
[6,]	"note"	"car"	"depart"	"month"	"new"		
[7,]	"week"	"rental"	"transfer"	"insur"	"home"		
[8,]	"problem"	"servic"	"offic"	"account"	"chang"		
[9,]	"form"	"home"	"phone"	"call"	"old"		
[10,]	"open"	"renter"	"custom"	"check"	"full"		

FIGURE 10. Words with Highest Beta when Identifying 10 CTM topics

5. Sentiment Analysis

In text mining, sentiment analysis or opinion mining, refers to the use of natural language processing and other analytical methods to extract and quantify affective information. Typically, sentiment analysis is used to identify different emotions expressed in text. Insurance companies have recently become more interested in behavioral psychology and behavioral economics as it applies to customers. For most customer interfacing companies, a better understanding of the customer can result in better service levels and/or improved marketing (Oguledo and Ramsay 2015, Golden et al. 2016 and Bauer et al. 2017). This is certainly true for insurance policies which are sometimes viewed as a commodity with undifferentiated service levels. By matching customer sentiments to various services performed (e.g., billing, policy changes, cancellation), insurers can learn where there are opportunities for improvement, or where to further leverage the favorable sentiment. Understanding customer sentiment can lead to better products and higher service levels which can differentiate an insurer and improve policyholder retention.

For purposes of this project, sentimental analysis was beneficial in understanding how customers feel during their call. We use the **SentimentAnalysis package** to accomplish this. This package includes four dictionaries, “QDAP,” (Hu and Liu 2004) “HE,” (Henry 2008) “GI,” (Stone 2004) and “LM” (Loughran and McDonald 2011)³, to find and categorize each call’s sentiment. Each dictionary has been created by a different group of researchers and assigns a positivity score to different words. For example, with a scale of -1 to 1 (most negative to most positive), words such as “hate” and “love” would have scores close to -0.8 and 0.8, respectively. By referencing the words in each call transcription with the words in each dictionary, our model categorized each call as “positive,” “neutral,” or “negative.” To quantify this information, we assigned each call a score of 1, 0, or -1, respectively. Thus, we were left with four values for each call, one from each dictionary. These values were averaged, ultimately providing each call an average sentiment from -1 to 1 in increments of 0.25.

To see whether any dictionaries were redundant, we measured the correlation between the results from each dictionary. Because we saw no significant correlation between the results from each dictionary, we proceeded with all four. The correlation matrix is displayed in the Table 2.

TABLE 2

Sentiment Analysis Dictionary Correlation

Sentiment Method Correlation Matrix				
	GI	HE	LM	QDAP
GI	1			
HE	0.0630	1		
LM	0.1343	0.1369	1	

³ Dictionary QDAP: Dictionary with polarity words from qdap package (Rinker 2017)
 Dictionary HE: Dictionary with opinionated words from Elaine Henry’s Financial dictionary
 Dictionary GI: Dictionary with opinionated words from the Harvard-IV dictionary as used in the General Inquirer software
 Dictionary LM: Dictionary with opinionated words from Loughran-McDonald Financial dictionary

QDAP	0.3769	0.1115	0.1738	1
-------------	--------	--------	--------	---

To see how customers felt during their calls, we created a histogram of the average sentiment scores. As seen in Table 3, most calls tend to fall in the 0-0.5 range, indicating a general feeling somewhere between neutral and slightly positive in their interactions. The average sentiment was 0.2198 and the standard deviation was 0.4071. Because transcriptions included both customer and representative, we tried to account for confounding potential distortion in results by filtering common phrases such as “thank you for calling” or “happy to help.” Another challenge with sentiment analysis is that it is very difficult for algorithms to accurately detect jokes or sarcasm; however, we do not believe this poses a major issue for this dataset. Finally, it is important to note that these sentiment scores were assigned to calls after stop words had been removed. Most stop words would not have made a difference as they would not be referenced by any dictionary, however results may change if the calls are run again with all words included.

A call received “NA” if none of its words could be found in any of the sentiment analysis dictionaries. These calls were usually very brief and usually comprised entirely of stop words. The sentiment histogram in Figure 11 shows that they made up a negligible proportion of total calls.

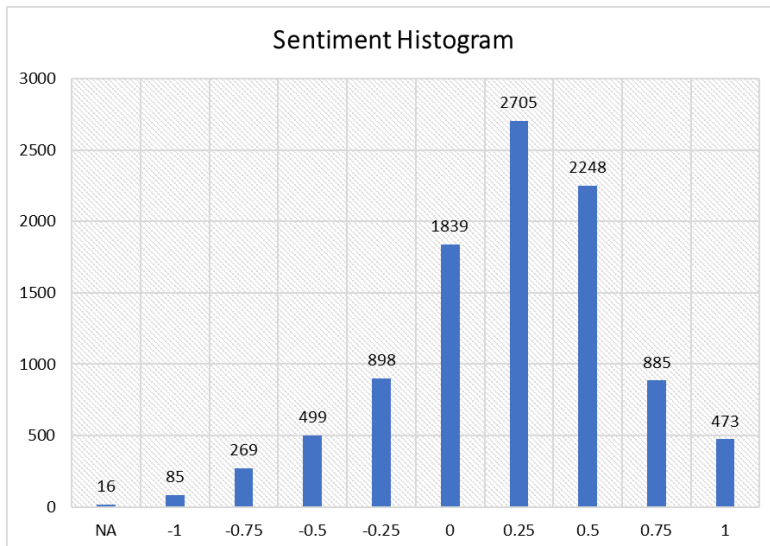


FIGURE 11. Histogram of Average Sentiment Scores

5.1 Sentiment by Topic

Finally, to see if sentiment differed across topics, we checked average scores across the major VEM topics. The result is shown in Table 3. The variation of sentiment by topic is not significant.

Results from our sentiment analysis appear over-simplified due to the data limitations of the transcription process. In particular, the text from the customers was not clearly differentiated from the text from the company representatives. We believe this neutralized our results significantly. Having said this, the results appear correct directionally, i.e., “cancellation” is showing the lowest sentiment rating.

TABLE 3
Average Sentiment and Standard Deviation by Topic

Topic	Avg Sentiment	Sentiment Std Deviation
[Uncategorized]	0.218	0.404
Add/Remove Vehicle	0.223	0.405
Administrative	0.232	0.412
Auto Policy	0.240	0.411
Billing	0.220	0.413
Cancellation	0.203	0.397
Change/Renew	0.221	0.404
Coverage	0.216	0.397
Premium/Rates	0.220	0.421

6. Discussion

6.1 Predictive modeling

We discussed a classification problem by unsupervised topic modeling, i.e., with words in customer's calls as the only variable: our goal was to identify underlying topics. When each call is labeled with a response variable, (for example, a sentiment score or claim severity), then with more predictors, numeric or qualitative, the work in this paper can be extended to be a regression problem by supervised topic modeling where the interest is in discovering which factors or topics are predictive of changes in callers' sentiment. In this case, machine learning models such as multinomial logistic regression can be used. More discussion about supervised topic modeling can be found in Blei and McAuliffe (2010). (The method in that paper is not incorporated in R yet.)

6.2 Non-Negative Matrix Factorization

Non-negative matrix factorization (NMF) is a group of algorithms in multivariate analysis and linear algebra in which a matrix is factored into (usually) two different matrices, with the property that all three matrices contain no negative elements. NMF has extensive applications and can be used in text mining as a form of topic modeling or classification. It is useful in generating a set of words which represent weighted sets of co-occurring words and can produce an efficient representation of text. However, the factorization is not unique; it is subject to the parameters and algorithm selected by users. In addition, outliers have drastic effects on the model. So far, R does not have very developed packages which allow for easy implementation of NMF, but it could be a good supplementary tool of topic modeling.

6.3 Improving transcription software

Many challenges we faced were posed by the dataset itself. Call transcriptions were difficult to read and contained many inaccurate transcriptions. Additionally, the lack of punctuation and lack of clear indication of who was speaking confounded results significantly. If the company could upgrade their transcription software to include features such as voice biometrics and real-time speech analytics, the results will be improved. More accurate transcriptions will reduce data pre-processing times as well.

6.4 Comparing identified topics to word search

To test the accuracy of our topic modeling models, we tried identifying topics and categorizing calls simply by searching for key words in the dataset. Although tedious and a misallocation of a data scientist or actuary's skills, this method works well for checking results.

7. Conclusion

We demonstrated a text mining procedure applied to customer calls data from an insurance company in the widely used statistical software R. The **tm** package provided a framework which allowed us to create a corpus, i.e., a collection of documents (calls). Using this entity, we further preprocessed the text in each call including removing stop words and stemming, and sorted the words by their importance. Visual exploration such as word frequency histogram and word clouds were used to reveal important words and underlying themes among the calls. With the term document matrix created by the **tm** package as input, we used the **topicmodels** package to identify latent topics among the calls via Bayesian hierarchical modeling. Finally, sentiment analysis was conducted with the **SentimentAnalysis** package, which showed that on average, customers calls tended to be slightly positive.

Ultimately, our results showed what customers were calling about and how they felt during their calls. This information can be used by the insurance company to prepare their representatives for common customer issues, lead customers to the online payment portal efficiently, and let them know that customers tend to be satisfied with their calling experience. When looked at in depth, this information also revealed how customers feel about specific topics, possibly helping the company tune their processes to improve caller satisfaction.

Acknowledgements

This project was jointly funded by the University of California at Santa Barbara and the Society of Actuaries.

References

- Arun R., V. Suresh, C. E. V. Madhavan, and M. N. Murthy. 2010. On Finding the Natural Number of Topics with Latent Dirichlet Allocation: Some Observations. *Advances in Knowledge Discovery and Data Mining* pp. 391-402. American Association for Artificial Intelligence Menlo Park, CA
- Bauer D., J. Gao, T. Moenig, E. R. Ulm and N. Zhu. 2017. Policyholder Exercise Behavior in Life Insurance: The State of Affairs. *North American Actuarial Journal*. 21(4): 485-501.
- Blei D.M., A. Y. Ng and M. I. Jordan. 2003b. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3: 993-1022.
- Blei D.M. and J. D. Lafferty. 2007. A Correlated Topic Model of Science. *The Annals of Applied Statistics* 1(1): 17-35.
- Blei D.M. and J. D. Lafferty. 2009. Topic Models. In A. Srivastava, M. Sahami (editors.), *Text Mining: Classification, Clustering, and Applications*. *Chapman & Hall/CRC Press*.
- Blei D.M. and J. D. McAuliffe. 2010. Supervised Topic Models. arXiv:1003.0783.
- Cao J., T. Xia, J. Li, and Y. Zhang. 2009. A Density-Based Method for Adaptive LDA Model Selection. *Neurocomputing*. 72: 1775-1781.
- Deveaud R., É. SanJuan, and P. Bellot. 2014. Accurate and Effective Latent Concept Modeling for Ad Hoc Information Retrieval. *Document numérique*. 17(1): 61-84.
- Feinerer I., K. Hornik, and D. Meyer. 2008. Text Mining Infrastructure in R. *Journal of Statistical Software* 25(5): 1-54.
- Fellows I. 2014. wordcloud: Word Clouds. R Package Version 2.5. <https://CRAN.R-project.org/package=wordcloud>.
- Feuerriegel S. and N. Proelochs. 2018. SentimentAnalysis: Dictionary-Based Sentiment Analysis. R Package Version 1.3-2. <https://CRAN.R-project.org/package=SentimentAnalysis>.
- Golden L.L., P. L. Brockett, J. Ai and B. Kellison. 2016. Empirical Evidence on the Use of Credit Scoring for Predicting Insurance Losses with Psycho-Social and Biochemical Explanations. *North American Actuarial Journal*. 20(3): 233-251.
- Griffiths T.L. and M. Steyvers. 2004. Finding Scientific Topics. *Proceedings of the National Academy of Sciences*. 101, suppl 1: 5228-5235.
- Grün B. and K. Hornik. 2011. topicmodels: An R Package for Fitting Topic Models. *Journal of Statistical Software* 40(13): 1-30.
- Henry E. 2008. Are Investors Influenced by How Earnings Press Releases Are Written? *Journal of Business Communication* 45(4): 363-407.

Hu M. and B. Liu. 2004. Mining Opinion Features in Customer Reviews. *Proceedings of the 19th National Conference on Artificial Intelligence*, pp. 755 - 760

Loughran T. and B. McDonald. 2011. When is a Liability not a Liability? Textual Analysis, Dictionaries, and 10-Ks. *Journal of Finance* 66(1): 35-65.

Nikita M. 2016. ldatuning: Tuning of the Latent Dirichlet Allocation Models Parameters. R Package Version 0.2.0. <https://CRAN.R-project.org/package=ldatuning>.

Ramsay C.M. and V. I. Oguledo. 2015. Optimal Disability Insurance with Moral Hazards: Absenteeism, Presenteeism, and Shirking. *North American Actuarial Journal*. 19(3): 143-173.

Rinker. T. W. 2017. qdap: Quantitative Discourse Analysis Package. R Package Version 2.3.0. <http://github.com/trinker/qdap>.

Silge J. and D. Robinson. 2016. tidytext: Text Mining and Analysis Using Tidy Data Principles in R. *The Open Journal* 1(3).

Stone P. 2004. Proposal for a Server-Based Inquirer. <http://www.wjh.harvard.edu/%7Einquirer>.