CS230

Sentiment Analysis of Company Earnings Conference Calls with Long Short-Term Memory

Shafiq K. Ebrahim
Stanford University
ebrahims@stanford.edu

Abstract

We examine the sentiment expressed during quarterly company earnings conference calls. Most sentiment analysis studies in the finance and accounting literature use techniques that ignore the order of words as well as the context of the information. Sequence modeling based on deep neural networks offer us the opportunity to capture the context in a more meaningful way. We compare the performance of the long short-term memory model (LSTM) with those of standard machine learning techniques. Our LSTM model struggles to outperform shallow machine learning models which suggests that a different deep learning architecture might be better suited to analyzing feature sets consisting of extremely long sequences of words.

1 Introduction

Earnings conference calls represent an important venue for companies to discuss their latest financial results and to offer insights into their expected future performance. These calls generally occur on a quarterly basis and typically consist of a presentation session during which senior management report their financial results (which is usually also available in the form of a press release) as well as a question-and-answer session involving buy-side and sell-side research analysts.

This topic has received some attention as part of a broader literature that undertakes a textual analysis of corporate disclosures.[1] Some studies of earnings conference calls find that the Q&A periods are relatively more informative than the presentation periods, while a recent study indicates that the market listens more closely to the tone of analysts' questions than that of management's responses.[2] An evaluation of sentiment from these corporate disclosures can facilitate investment decision-making and allow investors to forecast whether a company's stock price is likely to increase or decrease in the near future.

Most studies employ techniques that ignore the order of words as well as the context of the information in a document. In this paper, we use a long short-term memory (LSTM) model to assess the sentiment expressed during the Q&A session of quarterly earnings conference calls. We examine the question and answer segments separately. Our input consists of all the questions posed by analysts to company management in the former case, and all the responses by management in the latter. These documents are used to predict whether the sentiment of the conference call is either positive or negative.

The rest of this paper is organized as follows. Section 2 reviews the literature. Section 3 describes the dataset used in our study. Section 4 outlines our methods. Section 5 presents our findings, and the final section concludes.

[1] Excellent surveys of this work include Loughran and McDonald (2016), Kumar and Ravi (2016), and Kearney and Liu (2014).

[2] See Matsumoto *et al.* (2011), Price *et al.* (2012), and Brockman *et al.* (2015).

CS230: Deep Learning, Winter 2019, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

2 Related work

There are two prevalent methods of analysis in the financial textual sentiment literature: dictionary-based and machine learning. The former approach measures the polarity of texts according to the number of appearances of words from a predefined lexicon. The observed word list polarity count fractions are then used to predict sentiment following the filing date of the corporate disclosure in a regression framework.

The machine learning approach typically involves the application of the naïve Bayes and the support vector machines (SVM) classifiers to finance-related textual data. Examples include Antweiler and Frank (2004) who look at the information content of online message boards, and Huang *et al.* (2014) who analyze the market reaction to analyst report sentiment. Huang *et al.* (2014) note that the naïve Bayes approach is more effective in extracting textual opinions than dictionary-based methods.

A number of authors have successfully employed Long Short-Term Memory (LSTM) recurrent neural networks in learning long-term dependencies in textual data.[3] These methods have not been used much in the financial sentiment literature, though. An exception is Kraus and Feuerriegel (2017) who apply an LSTM model to a set of German ad hoc company announcements in English.

3 Dataset

3.1 Earnings Conference Call Transcripts

We obtain quarterly earnings conference call transcripts from S&P Capital IQ for the period from January 2008 to December 2017. We focus only on U.S. companies belonging to the Russell 1000 Index, an index of around 1000 companies that represent about 90% of the total market capitalization of the U.S. stock market. For every call transcript, we extract all the components of text corresponding to questions by analysts and combine them together for each call. We do the same for all the responses by company management. Figure 1 depicts an example of a snippet from the Q&A session of a recent earnings conference call.

Figure 1: Snippet from Q&A Session, Apple Inc. Q1 2019 Earnings Call, Jan. 29, 2019

[Steven Milunovich, Wolfe Research:] Some have the perception that you priced the new products, the new iPhones too high. What have you learned about price elasticity? And do you feel that perhaps you pushed the envelope a little bit too far and might have to bring that down in the future?

[Timothy Cook, Apple Inc.:] Steve, it's Tim. If you look at what we did this past year, we priced the iPhone XS in the U.S. the same as we priced the iPhone X the year ago. The iPhone XS Max, which was new, was $100 more than the XS. And then we priced the XR right in the middle of where the entry iPhone 8 and entry iPhone 8 Plus have been priced. So it's actually a pretty small difference in the United States compared to last year....

Figure 2 shows the numbers of conference calls in our dataset in each quarter. Our sample consists of 36,174 examples involving a total of about 134.5 million words of answers and 53 million words of questions. Table 1 provides summary statistics of the numbers of words. Each answers transcript has an average of 3718.2 words, and each questions transcript has an average of 1465.6 words.

Table 1: Summary statistics of length of earnings conference call transcripts (in words)

Component	Obs.	Mean	Std. dev.	Percentiles				
				5%	25%	50%	75%	95%
Answers	36174	3718.2	1403.4	1523.7	2791.0	3647.0	4581.0	6084.0
Questions	36174	1465.6	556.2	631.0	1088.0	1421.0	1796.0	2449.4

[3] e.g. Liu *et al.* (2015) and Wang *et al.* (2015). Zhang *et al.* (2018) surveys the literature in this area.

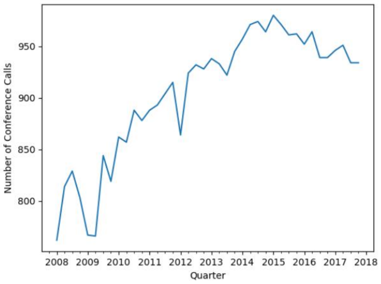

Figure 2: Number of quarterly conference calls in dataset, 2008 Q1 - 2017 Q4

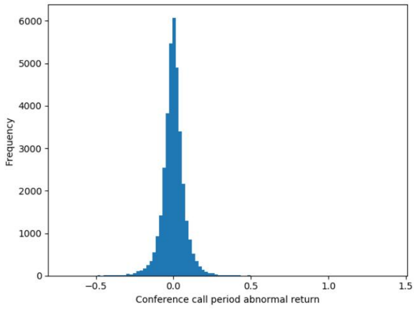

Figure 3: Frequency distribution of conference call period abnormal returns

3.2 Abnormal Stock Returns

The classification of the sentiment expressed by management or analysts on earnings conference calls is difficult because ground truth labels are not readily available (as in the case of movie review ratings, for example). Antweiler and Frank (2004) manually classify a small number of examples for training their naïve Bayes algorithm. However, this approach is subjective and susceptible to look-ahead biases related to the historical performance of companies. Also, it isn't practical for large training sets.

Another approach taken by Loughran and McDonald (2011) and Jegadeesh and Wu (2013) is to compute the abnormal stock return around the filing date of the corporate disclosure and use it as a gauge of the level of information conveyed to the market. We follow a similar approach. In particular, we obtain daily stock returns over the period from January 2008 to December 2017 from MSCI and Bloomberg and map them to transcript data using CUSIP identifiers. We compute the conference call period abnormal return, r_i, over a four-day window beginning on the date of the earnings conference call:

$$r_i = \prod_{t=0}^{3} ret_{i,t} - \prod_{t=0}^{3} ret_{index,t} \tag{1}$$

where $ret_{i,t}$ and $ret_{index,t}$ are the returns of stock *i* and the Russell 1000 Index on date *t*. We label each earnings conference call as positive or negative based on the sign of r_i. The frequency distribution of the abnormal returns is illustrated in Figure 3. Our dataset is well balanced; 49.8% of the dataset is classified as positive (49.9% of test examples are positive).

3.3 Training/Validation/Test Split

We split our dataset into training, validation, and test sets. The first 64% of the sample is our training set (23,150 examples), the next 16% is the validation set (5,788 examples), and the remaining 20% is the test set (7,236 examples). We follow Kraus and Feuerriegel (2017) and others in the literature by splitting our data in chronological order so as not to allow the training to have access to company-specific information that would have only been available ex post. Our validation and test sets are larger than is typical in deep learning studies because we want to ensure that the performance of our algorithm isn't overly dependent on an isolated period of stock market behavior.

4 Methods

4.1 Model Architecture

We use a long short-term memory model (LSTM) which is a type of recurrent neural network (RNN). This architecture is able to process information sequentially and makes use of three gates – the forget gate, the input gate, and the output gate – to control the flow of information through the cell state which acts as a memory. LSTMs are better able to deal with the problem of vanishing and exploding gradients during training than RNNs. We use a binary cross-entropy loss function:

$$L(y, \hat{y}) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})) \tag{2}$$

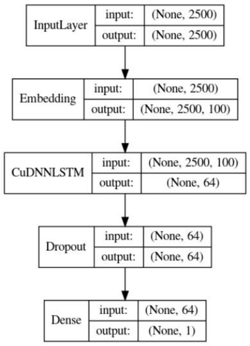

Figure 4: LSTM architecture

and accuracy as our evaluation metric. We perform our training in Keras (Chollet *et al.*) (2015)).

The inputs into this network are words from the conference call transcript. Rather than use sparse, high-dimensional word vectors based on one-hot encoding, we use word embeddings which are dense, lower-dimensional representations. Specifically, we use the pre-trained 100-dimension GloVe embeddings (400K vocabulary) without any additional fine-tuning (Pennington *et al.* (2014)). We optimize the network using the Adam algorithm with a learning rate of 0.0001 and a batch size of 32. Weights are initialized by the Xavier algorithm.

Our model consists of a single LSTM hidden layer with 64 units and a dropout rate of 0.8 followed by a dense layer with sigmoid activation function. This architecture is illustrated in Figure 4. We stopped training our network after 30 epochs as prior training up to 100 epochs indicated significant overfitting beyond this point. The accuracy of the validation set was maximized after about 30 epochs.

We compare the results of our LSTM with three models – a random guess baseline based on the training set classification probability and two standard machine learning algorithms – naïve Bayes and support vector machines. We implemented the machine learning algorithms in scikit-learn (Pedregosa *et al.* (2011)).

4.2 Hyperparameter Tuning

We tune the hyperparameters of the standard machine learning algorithms using a grid search approach with 5-fold time-series cross-validation on the first 80% of the sample (our training and validation sets). We consider various values of the alpha parameter and the maximum number of features for the naïve Bayes method, and the kernel function, cost, and maximum number of features for the SVM approach.

Given the computationally expensive nature of training the LSTM model, we tune its hyperparameters based on results observed in the validation set. We consider various values of the learning rate, batch size, number of epochs, and dropout rate on an iterative basis in our search for an effective architecture. We discuss our choices in more detail in the next section.

5 Results

5.1 Discussion

Training the LSTM model proved to be very challenging. The network appeared to be underfitting the data initially, so we added capacity by increasing the number of hidden layers from one to three and the number of hidden units up to 256. The speed of training slowed down considerably, but the model had difficulty achieving training and validation set accuracy above 50%. We reduced the learning rate, but that had only a minimal impact. We hypothesized that the input sequences may have been

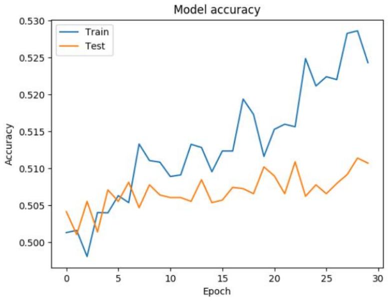

Figure 5: Model accuracy in training and validation sets - Answers segment

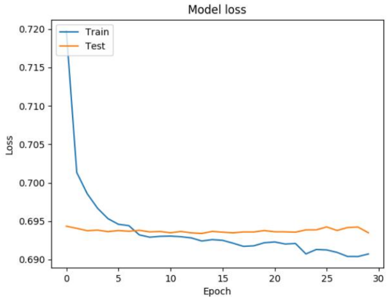

Figure 6: Model loss in training and validation sets - Answers segments

too long for the network to process effectively (with padding, they had a maximum length of over 18,000 words), so we applied a cap of 2,500 words for each document.[4]

Then, we were faced with a problem of overfitting: the training set accuracy increased to about 70%, but the validation set accuracy fell back to 50% after initially increasing to about 52%. So, we reduced the model capacity back to a single layer with 64 units and increased the dropout rate from 0.5 to 0.8. We experimented with L2 regularization and the gated recurrent unit (GRU) architecture, but our tests yielded qualitatively similar results. We also tried formulating the problem as a regression-based one with the predicted abnormal stock return as the output without much success.

5.2 Model Evaluation

Table 2 displays the classification accuracy statistics for the Answers and Questions segments of earnings conference calls. Both standard machine learning algorithms perform better than the random guess baseline with the linear support vector machines classifier performing the best. On the other hand, our LSTM performs poorly on the test sets. While the accuracy statistics for the standard machine learning models may seem low, they are likely sufficient to generate economically significant profits when used in systematic trading strategies. Interestingly, the performance of the naïve Bayes and SVM models support the finding of Brockman *et al.* (2015) regarding the market's greater attentiveness to the sentiment expressed in analyst questions than management responses.

Table 2: Sentiment classification accuracy for earnings conference call segments

	Answers		Questions	
Component	Training set	Test set	Training set	Test set
Baseline	0.500	0.497	0.496	0.496
Naïve Bayes	0.543	0.530	0.574	0.545
SVM	**0.577**	**0.547**	**0.612**	**0.571**
LSTM	0.531	0.495	0.503	0.501

Bold figures are highest values in each column.

6 Conclusion

We apply the LSTM model to the analysis of sentiment expressed in earnings conference calls. We have difficulty in training the model which suffers from significant overfitting and the inability to generalize. It is possible that a different architecture such as a convolutional neural network (CNN) or the training of even lower dimensional word embeddings based on our own corpus might yield more promising results. We leave that work for future research.

[4] We tried other cutoffs and found that the results were qualitatively similar below 5,000 words. Also, restricting the 175K transcript vocabulary to the most commonly used 50K words didn't yield materially different results.

7 Contributions

I wrote the code for data pre-processing and training of the machine learning and deep learning models. The code is available on GitHub at `https://github.com/sebrahim/cs230_project.git`.

Acknowledgements

I would like to thank Sagar Honnungar and Sarah Najmark for their helpful suggestions.

References

Antweiler, W. & Frank, M. (2004) Is all that talk just noise? The informtion content of Internet stock message boards. *Journal of Finance* **15**, 1259–1294.

Brockman, P., Li, X. & Price, S. M. (2015) Differences in conference call tones: Managers vs. analysts. *Financial Analysts Journal* **71**, 24–42.

Chollet, François and others (2015) Keras. `https://keras.io`

Davis, A. & Tama-Sweet, I. (2012) Managers' use of language across alternative disclosure outlets: Earnings press releases versus MD&A. *Contemporary Accounting Research* **29**, 804–837.

Huang, A., Zang, A. & Zheng, R. (2014) Evidence on the information content of text in analyst reports. *The Accounting Review* **89**, 2151–2180.

Huang, X., Teoh, S.H. & Zhang, Y. (2014) Tone management. *The Accounting Review* **89**, 1083–1113.

Jegadeesh, N. & Wu, D. (2013) Word power: A new approach for content analysis. *Journal of Financial Economics* **110**, 712–729.

Kearney, C. & Liu, S. (2014) Textual sentiment in finance: A survey of methods and models. *International Review of Financial Analysis* **33**, 171–185.

Kraus, M. & Feuerriegel, S. (2017) Decision support from financial disclosures with deep neural networks and transfer learning. *Decision Support Systems* **104**, 38–48.

Kumar, B.S. & Ravi, V. (2016) A survey of the applications of text mining in financial domain. *Knowledge-Based Systems* **114**, 128–147.

Liu, P., Joty, S. & Meng, H. (2015) Fine-grained opinion mining with recurrent neural networks and word embeddings. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1433–1443.

Loughran, T. & McDonald, B. (2011) When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *Journal of Finance* **66**, 35–65.

Loughran, T. & McDonald, B. (2016) Textual analysis in accounting and finance: A survey. *Journal of Accounting Research*, 1–42.

Matsumoto, D., Pronk, M. & Roelofsen, E. (2011) What makes conference calls useful? The information content of managers' presentations and analysts' discussion sessions. *The Accounting Review* **86**, 1383–1414.

Pedregosa, F. *et al.* (2011) Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830.

Pennington, J., Socher, R. & Manning, C. (2014) Glove: Global Vectors for Word Representation. *EMNLP* **14**, 1532–1543.

Price, S. M., Doran, J. S., Peterson, D. R. & Bliss, B. (2012) Earnings conference calls and stock returns: The incremental informativeness of textual tone. *Journal of Banking & Finance* **36**, 992–1011.

Wang, X., Liu, Y., Sun, C., Wang, B. & Wang, X. (2015) Predicting the polarities of tweets by composing word embeddings with Long Short-Term Memory. *Proceedings of the 53rd Annual*

Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 1343–1353.

Zhang, L., Wang, S. & Liu, B. (2018) Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **8**