Understanding Earnings Call Transcripts via Text Embeddings

Samuel Schwager and John Solitario
Stanford University, 450 Serra Mall, Stanford, CA 94305
sams95@stanford.edu and johnny18@stanford.edu
Project Github Repository

Abstract

We present a novel approach to analyzing earnings call transcript question and answer sessions wherein we seek to contextualize a given question by searching the statement section preceding the Q&A session for the "chunk" of text most similar to the question. Specifically, we choose cosine similarity as our similarity metric and experiment with a variety of text-embedding models to embed questions, statement chunks, and answers. We explore a simple bag-of-words model, a siamese neural network model to fine-tune bag-of-words embeddings, and a pre-trained BERT model. We find that the BERT model performs best both in terms of cosine similarity and qualitative performance (i.e. the quality, as determined by a human, of the statement chunk it assigns to a given question). Finally, we find that fine-tuning the bag-of-words embeddings using a simple neural network leads to significant quantitative and qualitative improvement.

1 Introduction

Since the inception of financial markets, individuals have repeatedly attempted to glean information from financial documents. From annual reports to balance sheets to news articles, investors have scoured all available financial text with the hope of gaining an investing edge. In particular, investors attempt to use financial documents to predict the future stock performance of publicly traded companies. Earnings call transcripts, which involve statements given by company officials followed by a question and answer (Q&A) session with banking analysts, have a particularly interesting structure compared to other types of financial documents. Specifically, earnings conference calls, and especially the Q&A sessions, involve rich and unscripted dialogue. Furthermore, questions and answers in these sessions typically refer back to content presented in the statement of the earnings conference call.

In this paper, we explore the relationship between question and answer pairs in the Q&A portions of earnings conference calls. In particular, we create a text similarity model that, given a question, finds the most relevant chunk of text from the statement portion of the call. We start with a simple bag-of-words model as our baseline, which allows us to quickly embed chunks of text. However, the bag-of-words model represents text as a set of independent words, causing the chunks of text to lose syntactic structure and context. Therefore, we develop a siamese neural network model with a loss function that enforces a contextual representation of questions and their corresponding answers in order to enrich the bag-of-words embeddings. Finally, we leverage a pre-trained BERT model to generate embeddings that account for both context and syntactic structure without the need for fine-tuning due to the BERT loss function and the enormous corpus on which the model has been pre-trained. BERT works extremely well on the given task, and our research acts as a promising proof-of-concept for future work on applying contextual embeddings to analyze earnings call transcripts.

2 Related Work

Due to the obvious financial incentives and applicable use cases, researchers have put a significant amount of work into understanding and modeling the effects financial documents have on the stock price and performance of a given firm. Colm Kearney and Sha Liu summarize recent, influential findings about how textual sentiment impacts individual, firm-level, and market-level behavior and performance (Kearney and Liu,

2013). Furthermore, Kearney and Liu bucket financial documents into three main categories: corporate disclosures, media articles, and internet postings. They also highlight two popular methods for extracting sentiment from financial documents: dictionary-based approaches and advanced machine learning methods. However, Kearny and Liu only focus on the predictive nature that financial documents have on firm performance, and they fail to highlight other possible use cases for the wealth of publicly available financial documents.

Following Kearney and Liu's broad analysis of the space, Lee et al. focus on using 8-K documents to help predict stock price performance. In particular, Lee et al. find that including simple linguistic features, drawn from a unigram word model, improves their baseline model, which relies solely on financial metrics (H. Lee and Jurafsky, 2014). Instead of 8-K documents, Price et al. focus on leveraging earnings conference calls to model market reactions (S. McKay Price, 2011). Leveraging a more advanced system, Price et al. determine document sentiment with a bag-of-words model, utilizing the General Inquirer for word recognition and the Harvard and Henry dictionaries for comparison. We explore the bag-of-word model more deeply in the **Methods Section**. Also to note, Price at al. pay specific attention to the question and answering portion of earnings conference calls, as the question and answering portion contains richer, unscripted information.

Chen et al. move away from predicting stock performance and examine how the tone of earnings conference calls change throughout a given day (J. Chen and Lev, 2012). They use the DICTION system, in conjunction with the Loughran & McDonald finance-oriented dictionaries, to extract document tone. Similar to Price et al., Chen et al. place a special emphasis on the question and answering portion of conference calls. Davis et al. also rely on the DICTION system to determine the sentiment of earnings conference calls, and similar to previous results, they find a positive correlation between financial document sentiment and firm performance (A. Davis and Sedor, 2012). Finally, Frankel et al. leverage more advanced machine learning methods, including support vector regressions, supervised LDA, and random forest regression trees, to extract narrative content from conference calls that correlates with useful information. In particular, they find that models of narrative structure have reasonable construct validity and that these models could be further improved by further analyzing the the unique textual characteristics of earnings conference calls.

3 Methods

3.1 Bag-of-Words

With the bag-of-words model, we represent a chunk of text as a "bag" or set of its given words. This model completely disregards word order and co-occurrence while maintaining individual word frequencies. The bag-of-words model allows us to quickly convert text into vector representations. The embeddings resulting from the bag-of-words model have lengths equal to the size of the vocabulary, which is determined by the given corpus. Note: We further explore our data in the **Dataset and Evaluation Metrics Section**. Depending on the text, the embeddings created by the bag-of-words model typically remain sparse (i.e. a short chunk of text contains a small portion of words present in the entire vocabulary). (Mueller and Thyagarajan, 2016).

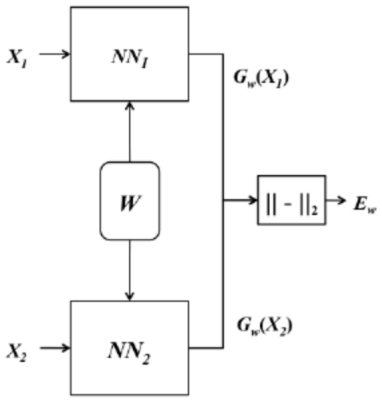


Figure 1: Architecture for a siamese neural network.

3.2 Siamese Neural Networks

First introduced by Bromley et al., siamese networks are neural networks containing two or more identical subnetwork components (J. Bromley and Shah, 1994). Furthermore, siamese networks use the same weights while working in tandem on two different input vectors to compute comparable output vectors. Thus, siamese networks have the

capacity to learn useful data descriptors that can be used when comparing output embeddings from upstream tasks (see figure (1)). Previous research has focused on using siamese networks for image retrieval and recognition tasks, but recent research has successfully leveraged these types of networks for tasks involving textual embeddings

3.3 Bidirectional Encoder Representations from Transformers (BERT)

Devin et al. introduced BERT in October 2018 to pre-train deep bidirectional language representations by jointly training on both left and right contexts of a given word in all layers of the model. Underlying BERT's architecture is a multi-layer bidirectional Transformer encoder, originally implemented by Vaswani et al. (A. Vaswani and Polosukhin, 2017). Transformers dispense entirely with recurrence and convolution mechanisms and rely solely on attention mechanisms, which significantly decreases training time.

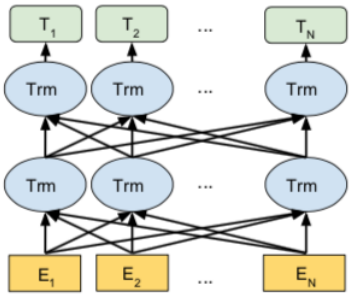
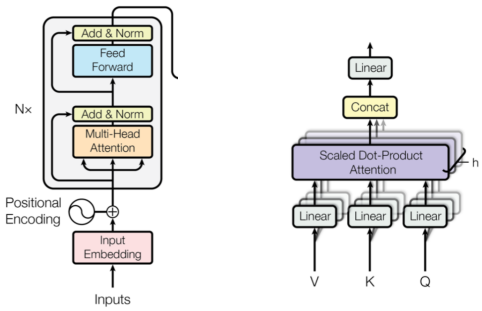


Figure 2: General BERT Architecture

Furthermore, the work of Devin et al. follows from a long history of pre-training general language representations. Specifically, it draws upon ideas central to GLoVe embeddings, developed by Pennington et al., which rely on word-to-word co-occurrence statistics, and ELMo embeddings, developed by Peters et al., which generalize traditional word embeddings by integrating context-sensitive features from language models (J. Pennington and Manning, 2014) (M. Peters and Power, 2014). Nonetheless, pre-trained embeddings have become integral to NLP systems, offering significant improvements over embeddings learned from scratch (J. Turian and Bengio, 2010). Although BERT is pre-trained on large corpus of unrelated text, there has been work showing the effectiveness of transfer learning from supervised tasks with large-scale datasets, such as natural language inference and machine translation (A. Conneau and Bordes, 2017) (B. McCann and Socher, 2017).

3.3.1 Model Overview

As noted previously, BERT's model architecture is a multi-layer bidirectional Transformer encoder based on the original implementation described by Vaswani et al. (A. Vaswani and Polosukhin, 2017). BERT relies on several layers of Transformer blocks (see figure (3)(a)), and each Transformer block consists of two sub-layers, a multi-head self-attention mechanism followed by a simple, position-wise fully connected feed-forward network. Residual connections exist around each of the two sub-layers, and dropout, following after each sub-layer, provides layer normalization.



(a) (b)

Figure 3: Key Components of the BERT Architecture (J. Devlin and Toutanova, 2018)

• **Multi-Head Attention:** We can describe an attention function as mapping a query (Q) and a set of key-value pairs (K, V) to an output vector (O), where the output is a weighted sum of the values. The weight assigned to each value is computed by a compatibility function of query (Q_i) with the corresponding key (K_i). Within BERT, the input matrix X, which either comes from the input embeddings or the previous hidden layer, is separately projected using the following weight matrices:

$$XW_i^Q = Q_i \in \mathbb{R}^{input \times d_q} \quad (1)$$

$$XW_i^K = K_i \in \mathbb{R}^{input \times d_k} \quad (2)$$

$$XW_i^V = V_i \in \mathbb{R}^{input \times d_v} \quad (3)$$

Instead of performing a single attention function, the multi-head attention mechanism linearly projects the queries, keys, and values h times with different, learned linear projections to d_v, d_k and d_q dimensions, respectively. On each of these projected versions of queries, keys, and values, the multi-head attention mechanism performs the attention function in parallel. The output vectors are then concatenated and once again projected, resulting in an output vector (O). Multi-head attention allows the model to jointly attend to information from different subspaces at different positions which bolsters effectiveness (see figure (3)(b)).

$$MHead(Q, K, V) = CAT(h_1, \ldots, h_h)W^O \tag{4}$$

$$h_i = Atten(Q_i, K_i, V_i) \tag{5}$$

• **Feed Forward:** The feed forward network, which comes after the multi-head attention mechanism, consists of two linear transformations with ReLU activitation in between.

$$FFN(O) = ReLU(OW_1 + b_1)W_2 + b_2 \tag{6}$$

• **Input Representation:** BERT can handle single sentences or pairs of sentences with a one token sequence. The input representation for a given token is constructed by summing over three different embeddings, corresponding to the token, segment, and position (see figure (4)).

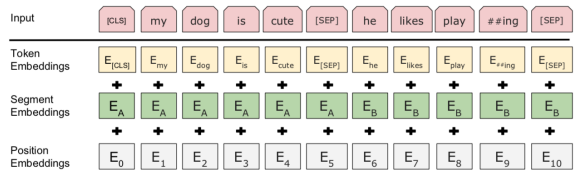


Figure 4: Input Embeddings

1. **Token Embeddings:** drawn from WordPiece embeddings (Y. Wu and Macherey, 2016).

2. **Segment Embeddings:** first token of every sequence is denoted with $[CLS]$. Sentence pairs separated with $[SEP]$. Learned sentence A embedding for every token of the first sentence and a sentence B embedding for every token of the second sentence.

3. **Position Embeddings:** learned and support sequence lengths up to 512 tokens.

• **Pre-training Procedure:** Before fine-tuning BERT, Devin et. al pre-train BERT on two novel unsupervised prediction tasks: masked language model (LM) and next sentence prediction. The masked LM task involves masking some percentage of input tokens at random, and then predicting only those masked tokens. In the next sentence prediction task, the model determines if a given sentence B proceeds the sentence A. Specifically, when choosing the sentences A and B for each pre-training example, 50% of the time B is the actual next sentence that follows A, and 50% of the time it is a random sentence from the corpus.

4 Data and Evaluation Metrics

4.1 Data

We initially received access to earnings call transcript data from the Stanford Graduate School of Business. The supplied data contains over 33,000 earnings call statements and their corresponding question and answer (Q&A) portions. The earnings conference call transcripts were originally pulled from seekingalpha.com, a crowd-sourced, content service website for financial markets. All of the earnings call transcripts come from 2017 or later. Given the relatively short time-period, approximately 2 years, we do not control for potential temporal difference between earnings call transcripts (e.g. economic downturn).

We originally received the data in the following for separate data tables:

1. *statements.csv* contains all of the dialogue from the statement portions of earnings call transcripts. Each row corresponds to the words from a given speaker, ordered chronologically. Thus, one earnings call transcript statement typically has a few associated rows, as different executives speak throughout the statement portion.

2. *qna.csv* contains all of the dialogue for the Q&A portion of the earnings conference calls. Similar to *statements.csv*, each row corresponds to the words from a given speaker. However, one Q&A session typically has numerous rows associated with it because many individuals speak during the Q&A portion, executives and analysts alike.

3. *executives.csv* tracks the executives present in each of the earnings conference calls. Thus, each row corresponds to an executive who speaks during a given call. The *executives.csv* table allows us to determine answer portions of dialogue during the Q&A session.

4. *analysts.csv* tracks the analysts present in each earnings call transcript. Similar to *executives.csv*, each row corresponds to an analysts who speaks during a given call Thus, *analysts.csv* allows us to determine the question portions of the dialogue during the Q&A session.

Since the information needed to represent a single earnings call transcript is disbursed over four different data tables, we perform a complex data aggregation procedure, in order to represent each earnings call transcript as a single data structure. Our final data structure discards all irrelevant information, and per call, our data structure retains the company name, time of call, statement dialogue, and Q&A dialogue. The company name and time of call act as a unique identifier. The statement dialogue represents all of the text associated with the statement portion of the call, divided into 64-word chunks. Furthermore, the Q&A dialogue represents all of the text associated with the Q&A portion of the call with each question and answer distinguished by a pair. Additionally, for the the statement and Q&A dialogue we remove all stop words and punctuation, and convert all of the letters to lowercase.

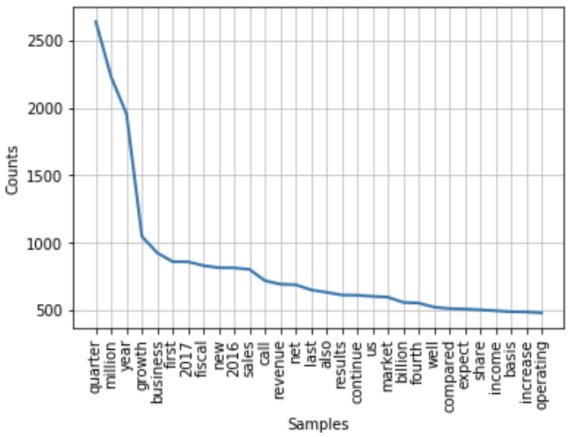


Figure 5: Word frequency distribution for **statements**.

Before performing our primary experiments, we explore the newly created corpus. Our corpus

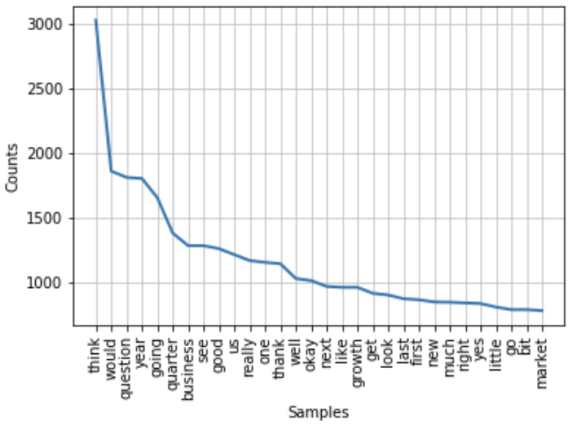


Figure 6: Word frequency distribution for **Q&A**.

contains approximately 62,000 unique words across the train-test split. As you can see, in the figure 5, words like "quarter", "million", "year", "growth", and "business" appear frequently in the statements portion of the transcript. Furthermore, as shown in figure 6, words like "think", "would", "question", "year", and "going" appear very often in the Q&A portion.

4.2 Evaluation Metrics

In order to compare embeddings of text chunks, we leverage the cosine similarity metric:

$$similarity(A, B) = \frac{A \cdot B}{||A|| \times ||B||} \qquad (7)$$

We use cosine similarity because the metric is normalized, bounded from $[-1, 1]$, which makes it easy to compare results across different models. Furthermore, cosine similarity is easy to calculate for sparse vectors, as only the non-zero dimensions need to be considered. Finally, cosine similarity works well for vectors representing discrete and continuous values. Originally, we considered using Jaccard similarity, but Jaccard similarity does not generalize as well to the continuous setting.

5 Experiments and Results

5.1 Experiments

Due to computational limitations and for the purpose of experimentation, we reduce the 33,000 earnings call transcript dataset into train and test sets of 1,586 and 680 transcripts, respectively. This corresponds to a total of 69,176 questions

and answers in the train set and 28,803 questions and answers in the test set. Although we are not able to make full use of the original data set, we believe that train and test sets with tens of thousands of questions each is more than sufficient to achieve our goal of proividing a proof-of-concept for our methodology.

Our first major experiment uses the train data to implement a bag-of-words model as follows:

1. Create bag-of-words vectors for each statement chunk and each question. Specifically, a bag-of-words vector is a vector in $\mathbb{R}^{|V|}$ where the i^{th} element is the number of times word i occurs in the text (V denotes the vocabulary for our dataset).

2. For each question, compute the cosine similarity between the question's bag-of-words vector and the bag-of-words vector for each statement chunk.

3. Choose the statement chunk from step 2 with the highest cosine similarity to be the most relevant for the current question.

Next, given the sequential nature of an earnings call transcript and the value in the Q&A session of the context provided in the statement section, we want to see how a more sophisticated model performs on our task. In order to emphasize the importance of context, we apply a contextual word embeddings model. Given the recent success of BERT, we apply an out-of-the-box pre-trained BERT model. Specifically, we use the BERT-Base uncased model and Han Xiao's "bert-as-a-service" tool to perform the following experiment:

1. For each statement chunk and question, pass the text through the pre-trained BERT model to get a 784-dimensional vector representation for each.

2. For each question, compute the cosine similarity between the question's BERT embedding and the BERT embedding for each statement chunk.

3. Choose the statement "chunk" from step 2 with the highest cosine similarity to be the most relevant for the current question.

Once again, our goal is to obtain a vector representation for each statement chunk and each question, in order to match questions to statement chunks. However, it turns out that some questions are quite long (i.e. hundreds of words), meaning that performing a single forward pass through the pre-trained BERT model is prohibitively slow. As such, we group the text of each question into 64-word chunks as has already been done for the statement text. Then we pass each of these chunks individually through the BERT model. However, since we want a single vector representation for each question, we combine each of the question chunk vectors into a single 784-dimensional vector using max pooling (i.e. given k 784-dimensional question chunk vectors, choose the final question vector to be the max of along each dimension of the k vectors).

Finally, after performing the BERT experiment, we realize that we are not using the actual answers from the Q&A sessions of the transcripts in any of these models. Although our task is inherently unsupervised, we realize that question-answer pairs afford a great deal of exploitable structure: after all, a question and its corresponding answer should be addressing the same underlying topic. In order to exploit this structure, we design a siamese neural network model that we train using the following procedure for each question-answer pair in our data set:

1. Given a valid question-answer pair and an invalid question-answer pair, compute a vector representation for each question and each answer using one of the two previously described models. Note that a valid question-answer pair is a pair that actually occurs in the data and an invalid pair is one that we have created by randomly pairing a question and an answer.

2. Pass each vector through a single-hidden-layer neural network that outputs a vector of a predetermined size.

3. Compute the cosine distance for the valid question-answer pair and the cosine similarity for the invalid question answer pairs

4. Add the cosine distance for the valid question-answer pair to the current loss (we want to minimize this distance) and add the cosine similarity for the invalid pair from the loss (we want to minimize this similarity).

The key here is that we are trying to "push" the siamese network outputs for valid question-answer pairs together and "pull" the outputs for invalid pairs apart. Ideally, our model would learn to extract the common topics being addressed by a valid question-answer pair and ignore the random signals provided by an invalid pair. This is exactly in line with our ultimate goal of creating a model that, given a question, finds the preceding chunk of text most "similar" to the question. Specifically, once we have trained this model, we can use it to identify the statement chunk that is most similar to a question in a transcript in the test set as follows:

1. For a given earnings call transcript in the test set, create vectors for each statement chunk and each question. Note that we should use the same embedding method at test time as we did during training (e.g. if we created inputs to the siamese network using bag-of-words during training, then we should use bag-of-words to create the network inputs at test time as well).

2. Pass each vector through our trained neural network, which outputs a new vector for each.

3. Choose the statement "chunk" from step 2 with the highest cosine similarity to be the most relevant for the current question.

5.2 Results and Discussion

All of our models map from statement chunks and questions to vectors that we ultimately use to determine the statement chunk most "similar" to a given question. Unfortunately, we do not have any labeled data (i.e. earnings call transcripts with labels associating statement chunks and questions). Nonetheless, we use cosine similarity between statement chunk vectors and question vectors, along with qualitative analysis, to assess the quality of the mapping we are learning from questions to statement chunks.

On the qualitative side, we examined the statement chunk-question associations for various earnings call transcripts in our test set after training our neural network model. We've displayed the outputs of our model for an interesting example in the Appendix section, and we feel that for the most part our models' outputs pick up on the major themes present in a substantial question. We are especially impressed by the outputs of the pre-trained BERT model, which is especially good at picking up on key finanical terms and proper nouns (e.g. Mexico in the example we've provided) present in the question. We also find that the outputs of the bag-of-words with siamese network fine-tuning model are significantly better than the results of the pure bag-of-words model. Finally, we note that we tried training the siamese network using the pre-trained BERT embeddings as inputs but did not observe significant improvement over the pre-trained BERT model.

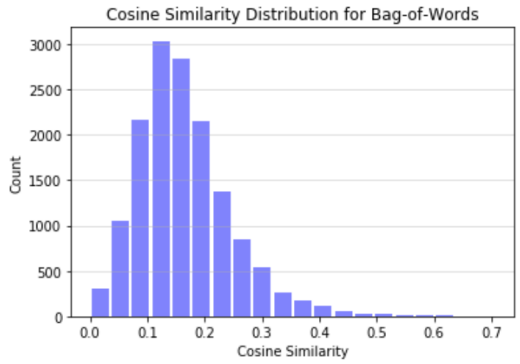
To provide a more quantitative foundation for our results, after training our final bag-of-words neural network model we performed the following process for the pure bag-of-words model, the bag-of-words model with the neural network, and the pre-trained BERT model:

1. For each question, compute the cosine similarity between its embedding under the current model and the embedding each statement chunk under the current model.

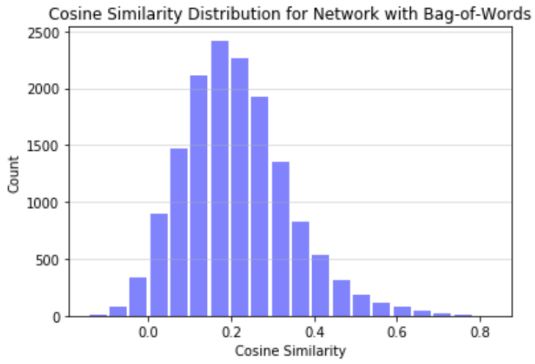
2. Plot the histogram of highest cosine similarity scores over the test set (discretize by dividing the range [-1, 1] into 20 buckets each of width 0.1).

These plots give us a sense of the quality of the embeddings outputted by a given model. For example, consistent with our qualitative analysis, we see that cosine similarity scores have shifted in the positive direction after combining the bag-of-words model with siamese neural network fine-tuning. Also, we note that this result is especially impressive since we have reduced the dimension of the embeddings from the size of the entire vocabulary to 128, a reduction by approximately a factor of 50. Thus, we appear to have compressed the meaningful information in the bag-of-words embeddings into a condensed and even contextual (due to our training procedure) representation.

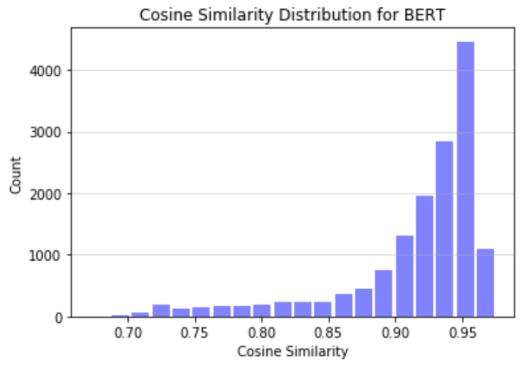
Furthermore, we see that the pre-trained BERT model's embeddings lead to exceptionally high similarity scores. The mode of the histogram for the pre-trained BERT model is 0.95, which is close to the maximum possible score of 1. We note that this is especially impressive since we performed absolutely no task-specific training



(a)



(b)



(c)

Figure 7: Cosine similarity distributions on the test set between questions and model outputs.

to produce these BERT embeddings. These embeddings are simply produced by a forward pass through a pretrained BERT model with no knowledge of our task.

Admittedly, cosine similarity was chosen somewhat arbitrarily, and in an ideal world we would have a labeled data set of question-statement chunk pairs. Nonetheless, we feel that cosine similarity is a good quantitative proxy for this ideal situation, especially when no training is performed and the model has no chance to "cheat" in order to enhance task-specific cosine similarity scores. Furthermore, we believe our intuition is supported by the alignment of our quantitative and qualitative results (i.e. our experiments indicate that models with high cosine similarity between questions and statement chunks tend to also perform well qualitatively). Finally, we acknowledge that a great deal of further exploration could be performed in terms of neural network architecture (e.g. a hyperparameter grid search) and the choice of contextual embeddings (e.g. GloVe may turn our to be a better choice than BERT). Nonetheless, we feel that we have developed an exciting and novel approach to contextualizing questions in settings in which much of the context (e.g. the statement section of an earnings call transcript) is known, even if we have only scratched the surface.

6 Conclusion

We presented a novel approach to analyzing earnings call transcript Q&A sessions through the application of textual embeddings. Our primary goal was to identify the chunk of text in the statement section most "similar" to a given question. In the absence of labeled data, we relied on cosine similarity to measure the relevance of a statement chunk to a given question. We found that a simple bag-of-words model performs relatively well at our task and that its performance is significantly enhanced via fine-tuning with a siamese neural network with a loss function enforcing similarity between valid question-answer pairs and dissimilarity between invalid (i.e. randomly assigned) question-answer pairs.

Perhaps most notably, we found that a pretrained BERT model with no additional training performed exceptionally well at our task. We believe that our work constitutes a sound proof-of-concept for the application of textual embeddings, especially those that emphasize context such as BERT, to the analysis of earnings call transcripts. Finally, we hypothesize that our results extend to the general case of associating a question with the preceding text most relevant to it in a setting where most of the applicable context is contained, and we believe that testing this hypothesis would be an exciting avenue for future research.

Authorship

Samuel and John equally participated in reviewing the associated literature, designing the experimental protocol, and compiling the final paper. John wrote the code for the bag-of-words model and Siamese neural network, while Samuel wrote the code for cleaning and formatting the data, building the BERT model, and evaluating results.

References

H. Schwenk L. Barrault A. Conneau, D. Kiela and A. Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 670680.*

J. Piger A. Davis and L. Sedor. 2012. Beyond the numbers: Measuring the information content of earnings press release language. In *In Contemporary Accounting Research Vol. 29 No. 3*.

N. Parmar J. Uszkoreit L. Jones A. N. Gomez L. Kaiser A. Vaswani, N. Shazeer and I. Polosukhin. 2017. Attention is all you need. In *In Advances in Neural Information Processing Systems*.

C. Xiong B. McCann, J. Bradbury and R. Socher. 2017. Learned in translation: Contextualized word vectors. In *In NIPS*.

B. MacCartney H. Lee, M. Surdeanu and D. Jurafsky. 2014. On the importance of text analysis for stock price prediction. In *Language Resources and Evaluation Conference*.

Y. LeCun E. Sickingerm J. Bromley, I. Guyon and R. Shah. 1994. Signature verification using a siamese time delay neural network. In *Advances in neural information processing systems*.

E. Demers J. Chen and B. Lev. 2012. Oh what a beautiful morning! the effect of the time of day on the tone and consequences of conference calls. In *SSRN Electronic Journal*.

K. Lee J. Devlin, M.W. Chang and K. Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *Arvix*.

R. Socher J. Pennington and C. D. Manning. 2014. Glove: Global vectors for word representation. In *In Empirical Methods in Natural Language Processing (EMNLP), pages 1532 1543*.

L. Ratinov J. Turian and Y. Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 384394*.

C. Kearney and S. Liu. 2013. Textual sentiment analysis in finance: A survey of methods and models. In *SSRN Electronic Journal*.

C. Bhagavatula M. Peters, W. Ammar and R. Power. 2014. Semi-supervised sequence tagging with bidirectional language models. In *In Empirical Methods in Natural Language Processing (EMNLP), pages 1532 1543*.

J. Mueller and A. Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*.

D. Peterson B. Bliss S. McKay Price, J. Doran. 2011. Earnings conference calls and stock returns: The incremental informativeness of textual tone. In *Journal of Banking and Finance*.

Z. Chen Q. V. Le M. Norouzi W. Macherey M. Krikun Y. Cao Q. Gao Y. Wu, M. Schuster and K. Macherey. 2016. Googles neural machine translation system: Bridging the gap between human and machine translation.

Appendices

Question	terrific last question give guys lot credit even trying handicap mexico risk current administration understand reduction core eps outlook balance like much currency expectation laid prepared remarks versus kind softening terms underlying fundamentals market
Answer	vivien tell unrelated fx say piece related example 17 18 fuel increase pemex january 1 mexico imports much refined product could argue result depreciation peso course goes directly transportation cost use fuel business course question happens happens mexican consumer obviously indicated fuel example real cost increase mexican consumer happens overall given uncertainty nafta wall economy ultimately goes consumer obviously know world changed lot last months clearly tried say handicap last update everybody look business end first quarter felt current world today ought give recognition uncertainty guidance
Model	**Output**
Bag-of-Words	excluding foreign currency translation transaction core eps 8 negative foreign currency transaction impact primarily driven mexico source certain inputs dollars year sales volume 1 reported net sales 3 currency neutral net sales 4 core operating income 5 yielding operating margin 60 basis points last year core eps year 9 excluding foreign currency translation transaction impacts core operating income 8 core eps 12 year solid
BERT	departure past practice believe warranted circumstances reference peso recently trading around mxn dollar rate hold balance 2017 core eps could increase current guidance certainly update throughout year furthermore somewhat related present economic uncertainty mexico potential impact consumer lowered core eps expectations another clearly better feeling progress year regarding effect bai acquisition expected 2017 core eps expect impact dilutive solely result application purchase accounting important
Network with Bag-of-Words	departure past practice believe warranted circumstances reference peso recently trading around mxn dollar rate hold balance 2017 core eps could increase current guidance certainly update throughout year furthermore somewhat related present economic uncertainty mexico potential impact consumer lowered core eps expectations another clearly better feeling progress year regarding effect bai acquisition expected 2017 core eps expect impact dilutive solely result application purchase accounting important

Figure 8: This is an example of an instance in which all three models perform relatively well, and bag-of-words converges to the same statement chunk as BERT after combining the bag-of-words model with the aforementioned Siamese network.