A BERT Perspective on Earnings Call Feature Mining

Quants Research Vertical[1, 2], Joon Woo Bae[1,3]
[1] CWRU Quants
[2] Case School of Engineering, Case Western Reserve University
[3] Weatherhead School of Management, Case Western Reserve University

This paper explores the application of sentiment analysis techniques to corporate earnings call transcripts. We aim to extract and quantify underlying sentiment and company exposure to pertinent political issues, such as government regulations, climate change policies, and international trade relations, by analyzing the language used by executives and analysts during these calls. Transcripts are decomposed into Presentation, Q&A Analyst, and Q&A Executive sections to facilitate individual and overall analysis. Our methodology begins with a baseline bag-of-words model, which is then enhanced with KeyBERT, a transformer-based model that leverages contextual word embeddings to improve the accuracy of keyword identification. We further incorporate readability metrics and sentiment analysis dictionaries to provide a comprehensive view of transcript characteristics. The resulting quantitative data will form the basis of future machine learning analysis to explore potential correlations between sentiment and stock price. This research contributes to the growing field of financial text analysis by demonstrating the value of sentiment analysis in uncovering nuanced insights from earnings calls, offering potential benefits for investors, analysts, and corporate management in understanding market perceptions.

1. Background

Earnings calls provide valuable insights into a company's performance and management's outlook. The analysis of these calls has become increasingly important for investors and analysts. Traditionally, active managers have sought an "edge" by utilizing research and in-depth analysis of various factors, including individual company specifics. Recent research has shown the potential of applying text-mining techniques to earnings call transcripts to extract meaningful information. For instance, Hassan et al. (2020) utilize textual analysis of earnings call transcripts to construct firm-level measures of political risk, showing that firms exposed to such risk tend to retrench hiring and investment, and actively lobby and donate to politicians. Chin and Fan (2022) demonstrate that these techniques can be used to derive features, including document attributes, readability, and sentiment, from earnings call transcripts. They also find that sentiment features derived from context-driven deep learning language models like BERT show promise and may have more efficacy than bag-of-words approaches. Hassan et al. (2023) also use a similar approach to construct text-based measures of firms' concerns related to epidemic diseases, such as COVID-19, SARS, and H1N1. These studies highlight the growing importance of textual analysis of earnings calls in understanding firm behavior and market dynamics. In this paper, we apply sentiment analysis techniques to earnings call transcripts to extract and quantify sentiment and political issue exposure, using a transformer-based model and additional metrics.

2. Methodology

Our initial approach was on a classical bag-of-words (BoW) framework. This approach involved a direct string-matching procedure where we compared transcript content against a curated CSV file consisting of political and risk-related keywords. Each occurance of the keyword was counted to quantify the density of politically sensitive or risk-oriented language within a given transcript. While this method provided an accessible starting point, its reliance on exact string matching limited its ability to capture semantic nuance or contextually similar expressions. As a result, it often failed to recognize politically charged or risk-related language that did not precisely match the predefined keywords.

To overcome the limitations of BoW, we used KeyBERT, a transformer- based keyword extraction model that leverages BERT embeddings. Rather than relying on exact matches, KeyBERT computes cosine similarity between embedded vectors to extract terms that are contextually relevant to user-defined seed keywords. This allowed us to surface semantically similar phrases, greatly enriching the quality of keyword identification and, consequently, the depth of sentiment and topical analysis.

We further expanded our methodology by introducing a diverse set of readability and sentimental analysis functions as additional features. For readability, we utilized metrics from the Textstat Python module, including the Dale-Chall Index, Flesch Reading Ease, and Coleman-Liau Index, among others. These scores helped us evaluate the structural complexity of transcripts, serving as a proxy for management clarity and communication effectiveness.

For sentiment analysis, we integrated dictionary-based methods using the Harvard IV-4 Psychosociological Dictionary (HIV4) and the Loughran–McDonald (LM) Master Dictionary, implemented via the pysentiment2 library. These dictionaries are well-established in financial and psychological text analytics.

3.1 KeyBERT

KeyBERT is first and foremost a transformer model. Trained on Wikipedia and BookCorpus data, and using vector embeddings, KeyBERT assigns every tokenized word in a string a high-dimensional vector. The next step is finding high similarities, KeyBERT takes seed words inputted by the user and returns words most similar to the given seed words using cosine similarity, a process that examines the angles between the embedded vectors in the high dimensional space to examine their similarity (similar vectors will have smaller angles between each other, yield a cosine similarity close to 1). Overall, KeyBERT's extract keywords function takes a text string, an n-gram range, stop words (useless words like a, the, etc.), maximum marginal relevance (essentially a cutoff value for similarity), inputted seed words and returns a list of the words selected.

Moreover, KeyBERT rectifies problems related to direct matching because it uses the aforementioned cosine similarity as opposed to exact matching, and we pick up more words that in turn produce a more accurate sentiment analysis. After appending words within an inputted range, the full function returns a dictionary that gives the phrases extracted and the seed word that the phrase is related to, this dict is then passed onto sentiment analysis functions.

3.2 Term Frequency Inverse Document Frequency (TF-IDF)

TF-IDF is a statistical method that is used to evaluate how important a word is in a document, relative to a corpus or set of documents.Its calculated by multiplying term-frequency (how often a word appears in a document) with inverse-document-frequency (how rare the word is in the set of documents).

The tf_idf function uses the scikit-learn package to calculate the TF-IDF scores on a set of documents. It returns the TF-IDF values for each unique word across all the documents and also returns the sparse matrix of the TF-IDF scores.

Although we have not specifically used this function yet in the project, it will be useful in the future for weighing the importance of risk/exposure words in relation to the rest of the corpus.

3.3 Readability Functions

A readability function calculates a readability score, which indicates how easy a given piece of text is to understand. This score is calculated using factors such as sentence length, average word length and the frequency of complex words. Reliability scores are used to determine the appropriate reading level for a target audience.

Several readability functions have been defined in readability.py using the Textstat module. `dale_chall` calculates the dale chall index for the input text. This index uses a list of 3000 words that are known by fourth grade students. Any word that is not in this list is considered difficult. It provides a numerical estimate of how difficult a piece of text is to comprehend.

The `flesch_ease` returns the flesch reading ease score. In this system, higher scores indicate easier readability while lower scores tell us that the text is easier to comprehend. Other functions such as `coleman_liau, overall, automated_readability, flesch_kinkaid, gunning_fog and smog_index` estimate the U.S grade level required to understand the text. For example, if the function outputs a score of 6.5, it means that the text is suitable for someone at the sixth to seventh grade reading level.

3.4 Sentiment Functions

The sentiment functions written are derived from the features list from the Chin et. al. paper. These functions pertain to two specific dictionaries, Harvard Psychosociological Dictionary (HIV4) and the Loughran−McDonald Master Dictionary (LM). These libraries were specifically developed for sentiment analysis of newspapers, documents, speeches, and other long form texts. The Python library used was pysentiment2, which is based on these LM and HIV4 dictionaries and has methods for each feature of these libraries: Positive, Negative, Net Sentiment, Polarity, and Subjectivity. We then made individual functions for each of these five features for both LM and HIV4.

LM refers to the Loughran-McDonald dictionary, with all of the words contained within the said dictionary having predefined sentiments of positive and negative. The dictionary was developed with specifically the purpose of use in financial contexts, making it a good choice for our purposes in

sentiment analysis. The more basic functions, LM_Positive and LM_Negative, first tokenize the strings given from a dict, and then return the number of positive or negative words, depending on the function. The net sentiment function returns the difference between the positive and negative score, so a negative score would imply a negative overall sentiment, with a positive return integer being a positive overall sentiment. The polarity score is defined as the difference between the positive and negative scores divided by their sum, similar to the net sentiment score, a polarity score implies positive sentiment when the score is positive, and negative sentiment when the score is negative. Finally, the Subjectivity score is defined as the sum of positive and negative words divided by the cumulative sum of positive, negative, and neutral words, with a score closer to 1 being more subjective than a score closer to 0.

For the HIV4 functions, HIV4 standing for Harvard IV 4, the calculations remain the same, with the only difference being that of the dictionary choice. The Harvard IV 4 dictionary was created in the 1960's to be more for general use of systematic transcript analysis to isolate emotional and cognitive aspects of language use. Even though the dictionary was made some time ago, and not explicitly for financial analysis, it still has quality capability to return positive and negative words, and the more broad analysis may even be beneficial when compared to the LM dictionary since it may pick up words that the LM functions would pass over.

4. Results

The final result is compiled to a csv file that contains the quantitative data from the integrated functions mentioned above. The dataset incorporated around 80 features which will lay the foundation for future development of the machine learning analysis to explore potential correlations between sentiment and stock price. Based on the current result, the political-risk discourse is present but sparse; it appeared mainly in prepared remarks and executive responses. The sentiment varies on different segments where it is chiefly positive in scripted statements and neutral to negative in Q&A. KeyBERT materially broadens coverage, furnishing a richer input set for the machine learning stage of the project.

5. Future Directions

The future continuation of this project will focus on assessing the correlation of the effect of sentiment on the prediction of the stock market behaviors using machine learning analysis on the result we have generated in this paper, which equips investors, analysts, and corporate management with a useful metric in understanding the market perceptions. Building on the present findings, machine learning analysis with suitable models deepens both the methodological rigour and the practical relevance of sentiment analysis in earnings-call research.

6. Acknowledgements

We would like to thank Professor Joon Woo Bae for advising our project, and the CWRU Quants club.