67-212 Project Final Report

Malika Dixit, mdikshit@andrew.cmu.edu
Erin Thomas, esthomas@andrew.cmu.edu

Abstract
Our project is a task of financial claim detection from managers' speeches in Earnings Conference Calls (ECCs). It is an information retrieval task and a binary classification task. The primary goal of our project is to determine which numerals in the ECCs are in-claim (relevant to the company's financial performance) or out-of-claim (not relevant to the company's financial performance). To solve this task, we propose the neural networks model which was balanced using the SMOTE technique as it yielded the best results of performance. The results of the model are F1 scores of 0.873 and 0.865 for the validation and test sets respectively.

1 Introduction

Analyzing financial data helps a business make better economic decisions. Published financial statements, earnings conference call events, and call transcriptions are all designed most importantly to meet the needs and answer the questions of external decision makers of the company, including present and potential stockholders, investors, creditors, and analysts.

In this paper, we focus on analyzing managers' speeches in Earnings Conference Calls (ECCs). All the numerals in the speeches might not directly be related to the financial performance of the company. For example, the speech might mention 2018, which is a year, which does not bear direct relevance to how the company has performed financially. This particular numeral is of no relevance to investors and creditors. On the other hand, a speech might also have numerals such as 5% which refers to the net profit margin of the company. This would be of great interest to investors who will make decisions of whether to invest in the company or not based on these figures. Numerals of direct relevance to the investors of a company and its financial performance are called 'in-claim' and numerals not related to the company's financial performance are called 'out-of-claim'.

In this paper, we present the tools we have developed to automatically label a numeral in an ECC speech as in-claim or out-of-claim.

This paper is organized as follows: Section 2 represents the related work done with the same dataset and task. In Section 3, we describe the data and the pre-processing steps we undertook in detail. The methodology and model description are elaborated on in Section 4. The evaluation and results we have observed are consolidated in Section 5. Section 6 gives a conclusion to the paper along with the scope for future work.

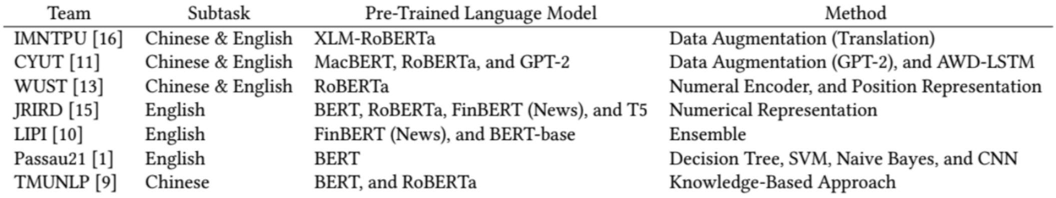
2 Related Work

The task we have worked on is part of a competition in which various teams participated and used several methodologies to get the best results. Chen et. al (2022) released the NumClaim (Chinese) and the NTCIR-16 FinNum-3 (English) datasets which comprised numerals present in financial texts and along with annotated labels (in-claim or out-of-claim). All of the teams in the competition focused on the F1 metric as the best indicator of the results of the task.

Alhamzeh et. al (2022), who participated in the competition with the team name Passau21, proposed the BERT (Bidirectional Encoder Representations from Transformers) base model, which had been pre-trained on a large corpus of English data. The results of their model were 86.48% macro-F1 Score in the validation split and 87.12% macro-F1 score in the test data.

Ghosh et al (2022). also made use of the BERT base model and combined the pre-trained language model with Logistic Regression, Random Forest, and Gradient Boosting Machine models. However, they got the best results with the BERT base model combined with Logistic Regression. Ghosh et al. achieved a Macro F1 score of 0.8223 on the validation set.

The various results obtained by the participants of the competition and the methodologies they used have been summarized in the figure below:



Team	Subtask	Pre-Trained Language Model	Method
IMNTPU [16]	Chinese & English	XLM-RoBERTa	Data Augmentation (Translation)
CYUT [11]	Chinese & English	MacBERT, RoBERTa, and GPT-2	Data Augmentation (GPT-2), and AWD-LSTM
WUST [13]	Chinese & English	RoBERTa	Numeral Encoder, and Position Representation
JRIRD [15]	English	BERT, RoBERTa, FinBERT (News), and T5	Numerical Representation
LIPI [10]	English	FinBERT (News), and BERT-base	Ensemble
Passau21 [1]	English	BERT	Decision Tree, SVM, Naive Bayes, and CNN
TMUNLP [9]	Chinese	BERT, and RoBERTa	Knowledge-Based Approach

Figure 1: Summary of participants' methods (Chen et al., 2022)

3 Data

The data was provided from the "NTCIR-16 FinNum-3: Investor's and Manager's Fine-grained Claim Detection" dataset (NTCIR-16 - finnum-3, n.d.). We explored the English dataset which consisted of data collected from Earnings Conference Calls, which are events where investors can hear detailed information about the most recent quarterly reports and ask questions for more background (Beers, 2022). The dataset contains 3 different files labeled as train, dev, and test files and they were in JSON format. The training file contained 8,337 records and the validation/development set file had 1,191 records, while the test set file had 2383 records. The files contained 6 columns, which are as follows :

- 'paragraph': the input sentence from the manager, recorded during the Earnings Conference Call
- 'target_num': the target numeral in the sentence
- ‘category': the category to which the target numeral belongs to, such as money, quantity_absolute, ranking
- 'offset_start': The start index of the target numeral in the sentence
- 'offset_end': The end index of the target numeral in the sentence
- 'claim': 1 means the target numeral is in-claim, and 0 means the target numeral is out-of-claim.

(*"NTCIR-16 - finnum-3"*, n.d.)

We also found that our training data had imbalanced classes for the target variable ('claim'), with 7298 records having 'claim' as 1, and only 1039 records having 'claim' as 0. We have taken measures to deal with this imbalance in the classes, which is explained in the methodology and models section of the report.

In terms of data preparation, firstly, we converted the JSON files to pandas data frames for ease of working with the data. For the data preprocessing, we performed four steps, which are as follows:

- Lowercasing the text: This was motivated by the intent of reducing variation in the entire sentence
- Punctuation removal: This was motivated by the intent of removing unnecessary tokens in the text. But we took care into preserved punctuations such as $, % and . as they were valuable to our task in deciding if the numeral had financial claim relevance
- Tokenization: This was motivated by the intent of breaking down the text into smaller chunks and was performed using nltk's word_tokenize() method
- Lemmatization: This was yet another pre-processing method performed to reduce variation in the text we had. Working with the lemma of words seemed more appropriate to our task than the stem of the works, which is why we chose lemmatization over stemming.

4 Methodology and Model Description

To approach this financial claim detection task, firstly we converted the earnings conference call text present in the files to Bag Of Words (BOWs) representation since it is the simplest form of text representation, and is widely used for information retrieval tasks ("Bag of words model. Engati.", n.d.).

To take care of the imbalanced classes in the data, the following two techniques were used:

- Balancing class weights: This approach "modifies the class weights of the majority and minority classes during the model training process to achieve better model results" (Amy, 2022).
- SMOTE (Synthetic Minority Oversampling Technique): creates synthetic data points based on the original data points by utilizing a k-nearest neighbor algorithm (Korstanje, 2021).

Then, we tested with four different models, which are the Naive Bayes model, the Decision Tree classifier model, the Logistic regression model, and the Neural Networks model, with Naive Bayes as the baseline model for our task.

Naive Bayes Model (Baseline): As our baseline in this task, we chose the Naive Bayes model. This model is generally expected to perform well in binary classification tasks. It also doesn't require a great amount of training data to learn from as long as it can infer the probabilistic relationship between the target variable and each of the attributes in the dataset individually, which was suitable for us as our dataset is quite small.

We fit our Naive Bayes model on the training data transformed through Bag of Words and the corresponding labels (0 or 1) of the transformed training data. The Naive Bayes model was then used to make predictions on the transformed data from the development set. The output of the predict method was the predictions of the model on the development set which were then compared to the actual labels of the development set to obtain the metrics of the model. We then replicated the process of predicting and comparing for the test set.

Decision Tree Classifier Model: For our first model to compete with the baseline, we chose to use a decision tree classifier model. This model is expected to be a good one for classification, because of "its ability to use different feature subsets and decision rules at different stages of classification" (Du & Sun, 2008).

Logistic Regression Model: For our second model to compete with the baseline, we chose to use the logistic regression model, since it is known to work well for binary classification tasks.

For both the decision tree classifier and the logistic regression model, we created three different variations of each of the models. First, we fit both these models on the BOW transformed training data. Next, we tried to balance the class weights on both these models, using the parameter class_weight='balanced' while building the model. And, lastly, we used the SMOTE technique to balance the classes. Each of these models was then used to make predictions on the transformed data from the development set and the test set. The output of the predict method was the predictions of the model on the development set which were then compared to the actual labels of the development set to obtain the metrics of the model. We then replicated the process of predicting and comparing for the test set.

Neural Networks Model: For our final model, we decided to use the powerful Vanilla Neural Network model on our dataset. We defined a sequential model with one hidden layer as part of our Neural Network. The hidden layer made use of the Relu activation function. The output layer used the sigmoid activation function, since our task is a binary classification one. We compiled the model using the binary_crossentropy function, and used the Adam optimizer function as it generally performs better than other optimizer functions such as Scholastic Gradient Descent (SGD). We then fit the model on our training dataset (transformed through BOW and balanced using SMOTE), and we specified the number of epochs as 10. We then predicted the model on the development dataset (which was also (transformed through BOW and balanced using SMOTE) and got the predictions for the development set as probabilities. We converted the probabilities to labels (0 and 1) and then compared the predictions against the actual labels of the development set to obtain the metrics of our model. We replicated the same process for our test set.

5 Evaluation and Results

In order to evaluate and compare the performances of our models, we chose the F1 score as the evaluation metric due to 2 reasons:

- It was the evaluation metric used by the competition of which this task is a part.
- F1 score was used because it is a metric wherein the “relative contribution of precision and recall are equal”, which was of importance to the task of financial claim detection itself.

We used the f1_score() function from sklearn.metrics library and the results for each of the models are as displayed in the table below:

Model	F1 score (for dev set)	F1 score (for test set)
Naive Bayes (baseline)	0.486	0.425
Decision Tree Classifier with imbalanced classes	0.330	0.266
Decision Tree Classifier with balanced class weights	0.396	0.321
Decision Tree Classifier with balanced classes using SMOTE	0.818	0.660

Logistic Regression with imbalanced classes	0.362	0.317
Logistic Regression with balanced class weights	0.451	0.410
Logistic Regression with balanced classes using SMOTE	0.829	0.852
Neural Networks with balanced classes using SMOTE	0.873	0.865

Table 1: Summary of F1 scores of all models for the validation and test datasets

Since our classes were highly imbalanced for the Naive Bayes (baseline) model, we expected poor results from the model and that is exactly what we observed. The F1 score for the test datasets was 0.425. We also observed high recall and low precision scores for both the development and test datasets, which can be explained by the imbalance of the datasets. The positive class is in the minority. Since the number of negative examples is much larger, the false positive results overwhelm the true positives, leading to high recall and low precision scores.

Once we balanced the classes using SMOTE, we saw an increase in the F1 scores for the decision tree classifier and the logistic regression model as compared to the baseline model, which were 0.660 and 0.852 respectively.

For the test datasets for the neural networks model, we observed the F1 score as 0.865. This was a huge improvement from our baseline, as well as a slight improvement from our previous models, showing the effectiveness of Neural Networks and the learning it had acquired with 10 epochs.

6 Conclusion and Discussion

With all our models for our task of financial claim detection, we have managed to improve the F1 scores from our baseline model. We also learned that balancing the classes using the SMOTE technique helped improve performances within the same model itself. Finally and most importantly, we observed that the Neural Networks model shows the greatest improvement in all metrics from our baseline, and hence, is currently our proposed best-performing model to solve the task.

In terms of future work, performing the task of financial claim detection using BERT and model combinations with BERT (like BERT & logistic regression) would be the next step. Since we have only looked into the English dataset of manager’s claim detection through conference calls, working with the Chinese dataset containing Analyst Report data to study the investor’s claim detection would also be a potential future extension of our work on a different language and dataset with the same task motivation.

References

Alhamzeh, A., Lacin, M. K., & Egyed-Zsigmond, E. (2022). Passau21 at the ntcir-16 finnum-3 task: Prediction of numerical claims in the earnings calls with transfer learning. In Proceedings of the 16th NTCIR Conference on Evaluation of Information Access Technologies (pp. 121-125).

Amy. (2022, June 28). Balanced weights for imbalanced classification. Medium. Retrieved November 6, 2022, from https://medium.com/grabngoinfo/balanced-weights-for-imbalanced-classification-465f0e13c5ad#:~:text=The%20balanced%20weight%20is%20one,to%20achieve%20better%20model%20results .

Bag of words model. Engati. (n.d.). Retrieved November 6, 2022, from https://www.engati.com/glossary/bag-of-words#toc-where-is-bag-of-words-used-

Beers, B. (2022, September 11). *What is an earnings conference call?* Investopedia. Retrieved November 9, 2022, from https://www.investopedia.com/small-business/what-is-an-earnings-conference-call/

Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2019. 2020. NumClaim: Investor's Fine-grained Claim Detection. In Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM 2020), Virtual Event, Ireland.

Du, C., & Sun, D. (2008). Decision Tree Classifier. Decision Tree Classifier - an overview | ScienceDirect Topics. Retrieved November 8, 2022, from https://www.sciencedirect.com/topics/computer-science/decision-tree-classifier

Ghosh S, Naskar SK. Detecting context-based in-claim numerals in Financial Earnings Conference Calls. Int J Inf Technol. 2022;14(5):2559-2566. doi: 10.1007/s41870-022-00952-7. Epub 2022 May 15. PMID: 35602417; PMCID: PMC9107601.

Korstanje, J. (2021, August 30). Smote | towards data science. Retrieved November 6, 2022, from https://towardsdatascience.com/smote-fdce2f605729

NTCIR-16 - finnum-3. (n.d.). Retrieved November 8, 2022, from https://sites.google.com/nlg.csie.ntu.edu.tw/finnum3