



FiNCAT: Financial Numeral Claim Analysis Tool

Sohom Ghosh and Sudip Kumar Naskar

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

March 18, 2022

FiNCAT: Financial Numeral Claim Analysis Tool

Sohom Ghosh
Fidelity Investments
Bengaluru, Karnataka, India
sohom1ghosh@gmail.com

Sudip Kumar Naskar
Jadavpur University
Kolkata, West Bengal, India
sudip.naskar@gmail.com

ABSTRACT

While making investment decisions by reading financial documents, investors need to differentiate between in-claim and out-of-claim numerals. In this paper, we present a tool which does it automatically. It extracts context embeddings of the numerals using one of the transformer based pre-trained language model called BERT. After this, it uses a Logistic Regression based model to detect whether the numerals is in-claim or out-of-claim. We use FinNum-3 (English) dataset to train our model. After conducting rigorous experiments we achieve a Macro F1 score of 0.8223 on the validation set. We have open-sourced this tool and it can be accessed from https://github.com/sohomghosh/FiNCAT_Financial_Numeral_Claim_Analysis_Tool

CCS CONCEPTS

• **Applied computing** → *Economics*; • **Information systems** → **Information retrieval**; • **Computing methodologies** → **Information extraction**.

KEYWORDS

numeral claim detection, financial text processing, natural language processing

1 INTRODUCTION

Call transcripts, financial documents relating to stocks, funds and organizations enable investors to make data-driven investment decisions. However, to persuade investors, narratives present in such documents may be just claims and not actual facts. Chen et. al released the NumClaim (Chinese) [1] and the NTCIR-16 FinNum-3 (English) [2] datasets which comprised numerals present in financial texts and along with the annotated labels (in-claim or out-of-claim). We use the English dataset [2] to develop **FiNCAT** - a tool to analyse numerals present in financial texts.

Our contributions

- We develop a tool to automatically detect whether numerals present in financial texts are in-claim or out-of-claim. To the best of our knowledge, we are the first one to develop such a tool.
- We have open-sourced¹ this tool as well as the embeddings and labels for further developments.

2 EXPERIMENTS AND RESULTS

We initiated by exploring the “NTCIR-16 FinNum-3 (English): Investor’s and Manager’s Fine-grained Claim Detection” dataset [2].

¹https://github.com/sohomghosh/FiNCAT_Financial_Numeral_Claim_Analysis_Tool

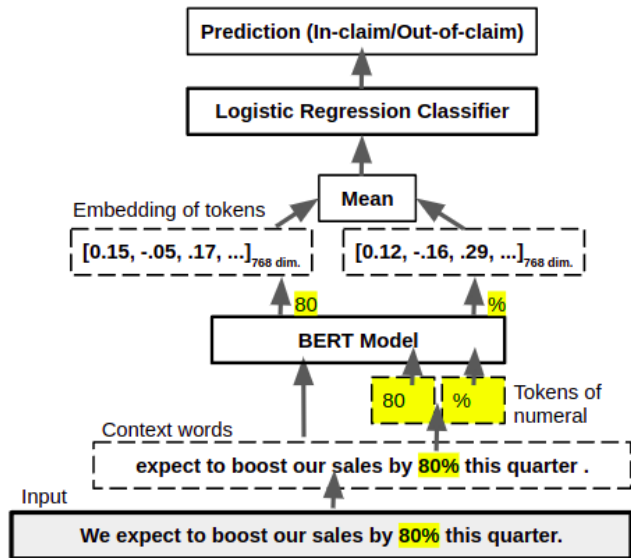


Figure 1: System Diagram of FiNCAT

Table 1: Model Performance on Training and Validation sets (LR=Logistic Regression, RF=Random Forest, GBM=Gradient Boosting Machine, LGBM=LightGBM, XGB=XG-Boost)

Model	Training		Validation	
	F1-Micro	F1-Macro	F1-Micro	F1-Macro
BERT + LR	0.9698	0.9283	0.9295	0.8223
BERT + RF	0.9922	0.9826	0.9211	0.7869
BERT + GBM	0.9996	0.9992	0.9270	0.7738
BERT + LGBM	0.9996	0.9992	0.9286	0.8009
BERT + XGB	0.9996	0.9992	0.9295	0.8054
RoBERTa + LR	0.9478	0.8694	0.9261	0.8034
RoBERTa + RF	0.9681	0.9318	0.8992	0.7461
RoBERTa + GBM	0.9996	0.9992	0.9219	0.7248
RoBERTa + LGBM	0.9996	0.9992	0.9270	0.7699
RoBERTa + XGB	0.9993	0.9983	0.9244	0.7588

The training and validation set had 8,337 and 1,191 records respectively. Furthermore, each of the target numerals was labelled as in-claim or out-of-claim by experts. Most of these financial texts had more than one target numeral. In order to deal with this, we tried to define a context window around the target numeral by considering a certain number of words before and after it. We empirically decided to use 6 words before and after the target numeral as the context window.

FiNCAT

Financial Numeral Claim Analysis Tool

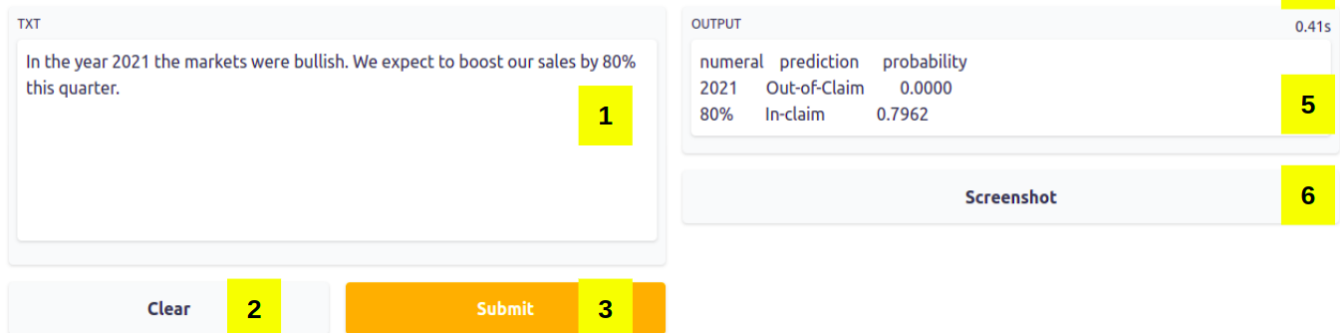


Figure 2: FiNCAT: Financial Numeral Claim Analysis Tool

We primarily experimented with two kinds of embeddings BERT-base [4] and RoBERTa-large [8]. We extracted the mean of the embeddings of the constituent tokens of the target numeral given the words in the context window. We trained several machine learning models using the mean embeddings as features to detect whether the target numeral was in-claim or not. These models include Logistic Regression, Random Forest [6], Gradient Boosting Machine [5], LightGBM [7] and XG-Boost [3]. Keeping the threshold at 0.5 and we used F1 score for evaluation.

Analysing the results presented in Table 1, we finally decided to move ahead with the logistic regression based model trained using BERT [4] embeddings (768 dimensions). It performed the best and was more efficient, explainable than the others. We present the final architecture in Figure 1.

3 TOOL DESCRIPTION

We deploy the tool using gradio² on Google Colab³. We present a screenshot of it in Figure 2. It comprises six parts: 1) **input text box**, 2) **clear button**, 3) **submit button**, 4) **execution time**, 5) **output text** and 6) **screenshot button**. The **input text box** takes any text as input. However, since this tool is specifically built for the financial domain, we recommend users provide texts related to finance like financial conversations, annual reports of organizations and so on. On pressing the **submit button** we look for words in the input text which contains at least one digit. For each such word, we evaluate the model described in section 2. This consists of computing the mean of contextual BERT [4] embeddings of the constituent tokens present in the target numeral. This mean (768 dimensions) is used as features to score the Logistic Regression model. Finally, we generate a **output table** which consists of three columns: i) numerals present in the input text ii) prediction stating whether the numerals are in-claim or out-of-claim and iii) probability predicted for each of them. The **screenshot button** and the **clear button** allow users to take screenshots and clear the entered texts respectively.

²<https://gradio.app/>

³<https://colab.research.google.com/>

We use Google Colab (free version CPU) to assess if it can detect in-claim numerals in real-time. We observe that the average time needed to generate predictions (**execution time**) for a given financial text consisting of 18 words and having 2 numerals is 0.25 seconds.

4 CONCLUSION

In this paper, we present a tool **FiNCAT** which uses context-based embeddings and machine learning to detect in-claim numerals present in financial texts. Presently, it takes only texts as input and checks for all the numerals present in the given text.

In future, we want to take the target numeral as an input from the user. This is supposed to reduce the computational time. Further tuning of hyper-parameters of the tree-based models and threshold used for prediction may yield better results. Based on the popularity we shall consider hosting it permanently using Hugging Face Spaces⁴. Another interesting direction for future research would be to explore different methods for generating embeddings of the target numerals as a whole rather than taking the mean of embeddings of its constituent tokens.

REFERENCES

- [1] Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2020. NumClaim: Investor’s Fine-Grained Claim Detection. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management (Virtual Event, Ireland) (CIKM ’20)*. Association for Computing Machinery, New York, NY, USA, 1973–1976. <https://doi.org/10.1145/3340531.3412100>
- [2] Chung-Chi Chen, Hen-Hsen Huang, Yu-Lieh Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2022. Overview of the NTCIR-16 FinNum-3 Task: Investor’s and Manager’s Fine-grained Claim Detection.
- [3] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (San Francisco, California, USA) (KDD ’16)*. ACM, New York, NY, USA, 785–794. <https://doi.org/10.1145/2939672.2939785>
- [4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>

⁴<https://huggingface.co/spaces>

- [5] Jerome H. Friedman. 2001. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics* 29, 5 (2001), 1189–1232. <http://www.jstor.org/stable/2699986>
- [6] Tin Kam Ho. 1995. Random decision forests. *Proceedings of 3rd International Conference on Document Analysis and Recognition* 1 (1995), 278–282 vol.1. <https://doi.org/10.1109/ICDAR.1995.598994>
- [7] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017), 3146–3154.
- [8] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs.CL]