FiNCAT: Financial Numeral Claim Analysis Tool

Sohom Ghosh
Fidelity Investments
Bengaluru, Karnataka, India
Jadavpur University
Kolkata, West Bengal, 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 can do this task automatically. It extracts context embeddings of the numerals using a transformer based pre-trained language model – BERT. Subsequently, it uses a Logistic Regression based model to detect whether a numeral is in-claim or out-of-claim. We use the FinNum-3 (English) dataset to train our model. We conducted rigorous experiments and our best model achieved a Macro F1 score of 0.8223 on the validation set. We have open-sourced this tool which 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

ACM Reference Format:
Sohom Ghosh and Sudip Kumar Naskar. 2022. FiNCAT: Financial Numeral Claim Analysis Tool. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3487553.3524635

1 INTRODUCTION

Call transcripts, financial documents relating to stocks, funds and organizations enable investors to make data-driven investment decisions. However, to persuade the 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 in which the numerals present in the financial texts are annotated with in-claim and out-of-claim labels. We use the English dataset [2] to develop **FiNCAT** - a tool to analyse numerals present in financial texts.

WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9130-6/22/04…$15.00
https://doi.org/10.1145/3487553.3524635

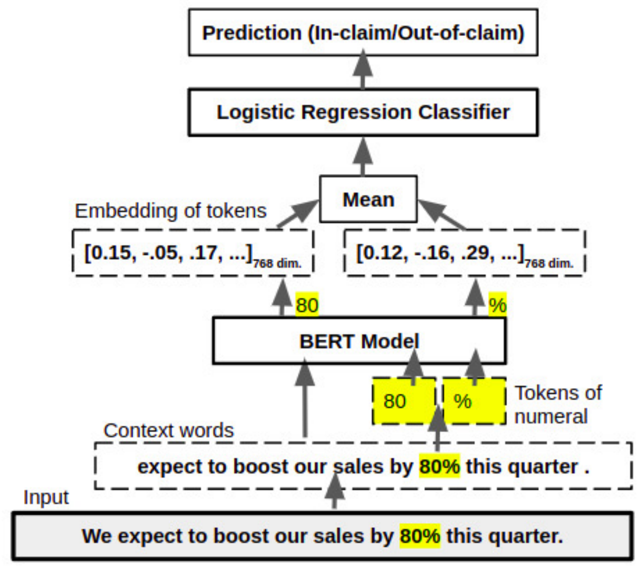

Figure 1: System Diagram of **FiNCAT**

Our contributions

- We developed 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-sourced[1] this tool as well as the embeddings and labels for further developments by the research community.

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]. The training and validation set had 8,337 and 1,191 records, respectively. Each of the target numerals in this dataset is labelled as in-claim or out-of-claim by experts. Most of these financial texts had more than one target numeral. We tried to define a context window around the target numeral by considering a certain number

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

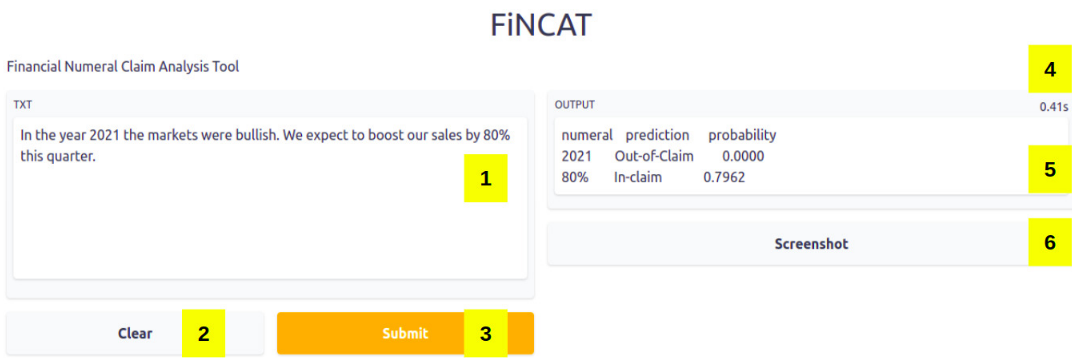

Figure 2: FiNCAT: Financial Numeral Claim Analysis Tool

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

	Training		Validation	
Model	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

of words before and after it. We empirically decided to use 6 words before and after the target numeral as the context window.

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 is in-claim or not. The models include Logistic Regression, Random Forest [6], Gradient Boosting Machine [5], LightGBM [7] and XG-Boost [3]. We kept the threshold at 0.5 and 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 is more efficient, explainable than the others. We present the final architecture in Figure 1.

3 TOOL DESCRIPTION

We deploy the tool using gradio[2] on Google Colab[3]. Figure 2 presents a screenshot of the tool. It comprises of six parts: 1) **`input text box`**, 2) **`clear button`**, 3) **`submit button`**, 4) **`execution time`**, 5) **`output`**, 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, etc. On pressing the **`submit button`** the tool looks for words in the input text which contains at least one digit. Each such word is evaluated using the model described in section 2. This consists of computing the mean of the contextual BERT [4] embeddings of the constituent tokens present in the target numeral. This mean (768 dimensions) of the contextual embeddings is used as features to score the Logistic Regression model. Finally, the tool presents the **`output`** in a tabular format consisting 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.

We used Google Colab (free version CPU) to assess if it can detect in-claim numerals in real-time. We observed 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 the hyper-parameters of the tree-based models and threshold used for prediction may yield better results. Depending on the popularity we shall consider hosting it permanently using Hugging Face Spaces[4]. Another interesting direction for future research would be to explore different methods for generating the embeddings of the target numerals as a whole rather than taking the mean of embeddings of its constituent tokens.

[2] https://gradio.app/
[3] https://colab.research.google.com/
[4] https://huggingface.co/spaces

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

[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]