Detecting context-based in-claim numerals in Financial Earnings Conference Calls

Sohom Ghosh 1,2 • Sudip Kumar Naskar 2

Abstract Most investors tend to make decisions after analysing financial documents of organizations available online. These documents include financial reports, conversations, brochures, etc. While reading these documents investors need to ensure that they rely only on facts and do not get swayed away by claims which representatives of organizations make. Thus, it is essential to have an automated system for detecting whether numerals present in financial texts are in-claim. In this paper, we discuss a system for evaluating whether numerals present in financial texts are in-claim or out-of-claim. It is trained on the English version of the FinNum-3 corpus using two variants of the FinBERT model and a BERT model augmented with handcrafted features. Our best model, an ensemble of these 3 models, produces a Macro-F1 score of 0.8671 on the validation set and outperforms the existing baselines.

Keywords Financial texts • Natural language processing • Claim detection • Transformers

1 Introduction

With the advent of the Internet and digitalization, most financial services and investment platforms have moved online. Organizations publish their performance reports and brochure digitally. Earnings conference calls of executives get transcribed and digitally preserved. Most investors rely on this information to make investment decisions. Numbers present in such information may be claims or not-claims (i.e. facts). Facts are always true whereas claims may be true or false. It is expected that investors will rely only on facts and not be allured by false claims. However, making such a distinction is not easy specifically for novice investors. Thus, we need to have an automated system that would be able to detect whether numbers in financial texts are claims (in-claims) or not (out-of-claims/facts). Figure 1 presents two instances. The number ‘23’ present in the text “For the full year we continue to expect an adjusted effective tax rate of 23–24%” is a claim. The number ‘1.1’ in the text “Free cash flow a really good start to the year at $1.1 billion.” is not a claim.1

1.1 Our contributions

• We developed a system that can detect whether a numeral present in a given financial text is a claim or not. For this, we used the English version of the publicly available dataset FinNum-3 [11]. On the validation set, our system achieved macro F1 score of is 0.8671.
• We studied how adding handcrafted features and information regarding the category of a target numeral affect the performance of the model.

This remaining paper is structured as follows. In Sect. 2 we discuss some of the existing works. We formally state the

1 Our team LIPI participated in the FinNum-3 (English) shared task [11] of NTCIR-16 workshop. This paper is an extension of our tool FiNCAT [14] and the corresponding short system description paper [15].
problem statement in Sect. 3 and describe the dataset in Sect. 4. In the subsequent Sects. 5, 6 and 7, we discuss the methodology, the experiments we performed and their results, respectively. Section 8 concludes and mentions some future work directions.

2 Related works

Detecting claims from texts using Natural Language Processing (NLP) has been one of the trending areas of research. This has been applied in various domains like NEWS [13, 26], Twitter [6], legal [24], etc. Hassan et al. [16] developed a system, ClaimBuster, to detect claims present in the 2016 US presidential primary debates. They evaluated ClaimBuster on statements selected for fact-checking by CNN and PolitiFact. They found that their system was able to detect several sentences with claims which were not selected for fact-checking by the above mentioned organizations. [13] created a new dataset by manually labelling the debates. They also proposed SVM and neural based systems to rank claims for prioritizing fact-checking. Subsequently, a similar application was presented by Konstantinovskiy et al. [19]. They used universal sentence representations for classification and out-performed existing claim ranking system [13] and ClaimBuster [16]. Furthermore, they proposed an annotation schema and a crowdsourcing methodology. This enabled them to create a dataset having 5571 sentences with labels as claims or non-claims. Reddy et al. [26] released a new dataset NewsClaim which consisted of 529 manually annotated claims collected from 103 news articles mostly relating to COVID-19. They showed that zero-shot and prompt-based approaches perform well in detecting claims from news articles.

Aharoni et al. [1] developed a dataset for detecting claims in controversial topics. It consisted of 2683 arguments which were collected from 33 controversial topics. Sundriyal et al. [29] proposed a novel framework called DESYR. It consisted of a gradient reversal layer and attentive orthogonal projection over poincare embeddings. They evaluated it on informal datasets like online comments, web disclosures, Twitter, etc. Chakrabarty et al. [6] created a corpus from Reddit consisting of 5.5 million self-labelled claims which contain “IMO/IMHO (in my (humble) opinion)” tags. They fine-tuned ULMFiT [18] on this corpus. They further demonstrated how fine-tuning helped in argument detection tasks. Wright et al. [31] proposed a unified model called Positive Unlabelled Conversion. It constituted of a positive unlabelled classifier and a positive-negative classifier. They evaluated their model on three datasets namely Wikipedia citations, Twitter Rumours and Political Speeches.

Levy et al. [20] trained context-dependent classifiers for detecting claims on Wikipedia corpus. It primarily consisted of three components—Sentence Component, Boundaries Component and Ranking Component. Subsequently, Levy et al. [21] proposed an unsupervised framework to detect claims and evaluated its’ performance on the same corpus. Lippi et al. [23] used Partial Tree Kernels to generate features for detecting claims irrespective of the context. The inner nodes of these trees consisted of POS tags of the words in the leaf nodes. Furthermore, Lippi et al. [24] validated the effectiveness of this approach in the legal domain. To do this, they manually annotated claims from fifteen decisions of the European Court of Justice. Bar-Haim et al. [4] expanded the initial set of manually curated sentiment lexicons and added some contextual features (like headers, claim sentences, neighbouring sentences and neighbouring claims) to improve the existing claim stance classification systems. Botnevik et al. [5] proposed a browser-based extension BRENDa that helped users to verify facts within claims which are present in different web pages.

Recently, with the increase in the availability of financial textual data, researchers have been focusing on detecting claims in financial texts as well [9, 10]. Chen et al. [9] presented a novel dataset NumClaim in Chinese which comprised financial texts, their categories and whether a target number within a text is in-claim or out-of-claim. They further proposed some neural architecture based baselines. Their best performing model CapsNet resulted in a macro-F1 score of 82.62% on the NumClaim Corpus. Recently, they released a similar dataset in English while organizing the FinNum-3 workshop [11].

3 Problem statement

Given a set $F = \{ (t_1, n_1, s_1, e_1, c_1, m_1), (t_2, n_2, s_2, e_2, c_2, m_2), \ldots, (t_k, n_k, s_k, e_k, c_k, m_k) \}$ of $k$ elements, the $i_{th}$ element of $F$ consists of a financial text ‘$t_i$’, a number ‘$n_i$’ present within the text having starting and ending index positions ‘$s_i$’ and ‘$e_i$’ respectively. Moreover, each element also contains $c_i$ which denotes the category $t_i$ belongs to and $m_i$ which represents whether $n_i$ is in-claim or out-of-claim. $m_i \epsilon \{0,
1], 0 and 1 representing out-of-claim and in-claim, respectively. $c_i \in \{’date’, ’other’, ’money’, ’relative’, ’quantity absolute’, ’absolute’, ’product number’, ’ranking’, ’change’, ’quantity relative’, ’time’\}$. Our target is to develop a system for classifying an unknown numeral ‘n’. We evaluate the performance of our models using macro-averaged F1-score.

4 Dataset

Our experimental dataset comprises transcripts from earnings conference calls in English. They are formal financial documents. A similar dataset in Chinese consisting of reports written by analysts has been described in more detail [9]. Recently, a shared task, “NTCIR-16 FinNum-3: Investor’s and Manager’s Fine-grained Claim Detection” [11], is being held where participants are provided with this dataset. We registered in the shared task and obtained the training and validation data. The training data consists of 8337 records whereas the validation data consists of 1191 records. Of all these records the train and validation set has 1,039 and 114 in-claim instances respectively. There are 2627 and 409 unique financial texts in the training set and validation set respectively. This indicates that most of the texts present in the training and validation sets have multiple numbers present in them. We present the category-wise distribution in Table 1.

5 Methodology

Our final system consists of an ensemble of 3 sub-systems. The first two sub-systems consist of fine-tuning pre-trained language model FinBERT [2] and are almost identical. The third one is a logistic regression based model built using contextual BERT embedding [12] of the numerals and other engineered features. BERT (Bidirectional Encoder Representations from Transformers) [12] is one of the state of the art language models. It has been pre-trained using masked language modelling (MLM) and next sentence prediction (NSP) objectives. We use the base and uncased version of it which consists of 768 hidden units, 12 attention heads and encoder blocks. It has a total of 110 million parameters and can be used to generate contextual embeddings of 768 dimensions. FinBERT [2] is a version of BERT which has been subsequently pre-trained on Financial text and fine-tuned for financial sentiment classification task. We fine-tune the FinBERT model even further for the text classification task to detect in-claim numerals. Since the given training set has multiple numbers that are present in the same text, we try to narrow down the context of the target numeral. For the first sub-system, we define context as 8 words before and after the numeral. For the second and third sub-system, we further narrow it down to 6 words around the numeral. The entire process is depicted in Fig. 2.

5.1 Sub-system-1(S1)

Firstly, we tokenize the financial texts and extract 8 words before and after the target numeral. We follow the standard method of fine-tuning a FinBERT model (768 dimensions) so that its [CLS] token learns to predict whether the target numeral is in-claim or out-of-claim. We run this model in batches of size 256 for 40 epochs with a learning rate of 0.00002. We consider a maximum of 64 tokens. Finally, we select the model which is tuned till 15th epoch as it performs the best on the validation set (Macro F1 score = 0.8585).

5.2 Sub-system-2(S2)

This sub-system is similar to the first one. The only differences are we narrow down the context around the target numeral from 8 to 6 and consider a maximum of 16 tokens.

Table 1 Category wise distribution of the training and validation set

| Set    | Category       | Count |
|--------|----------------|-------|
| Train  | Absolute       | 683   |
| Valid  | Absolute       | 85    |
| Train  | Change         | 398   |
| Valid  | Change         | 84    |
| Train  | Date           | 1616  |
| Valid  | Date           | 221   |
| Train  | Money          | 1496  |
| Valid  | Money          | 266   |
| Train  | Other          | 406   |
| Valid  | Other          | 106   |
| Train  | Product number | 235   |
| Valid  | Product number | 26    |
| Train  | Quantity absolute | 1193 |
| Valid  | Quantity absolute | 122  |
| Train  | Quantity relative | 178  |
| Valid  | Quantity relative | 45   |
| Train  | Ranking        | 35    |
| Train  | Relative       | 2089  |
| Valid  | Relative       | 236   |
| Train  | Time           | 8     |
We do this to focus specifically on the target numeral. This model performs the best just after the 14th epoch (Macro F1 score = 0.8439).

5.3 Sub-system-3 (S3)

This sub-system is different from the previous two. In this sub-system, given a context window of 6 words, we first extract BERT base uncased embedding (768 dimensions) of the target numeral. Since we have been using sub-word tokenization, for many cases the target numerals resulted in more than one token. This is one of the drawbacks of transformer based models. It has also been mentioned by Wallace et al. in the paper [30]. To deal with such instances, we take the mean of the embeddings of all the constituent tokens. Moreover, being inspired by [3, 22, 28] and [8] we engineer several features from the target numerals. These features include

- number of digits before the decimal
- number of digits after the decimal
- one-hot vectors of different categories extracted using Microsoft Recognizers for Text
- one-hot vectors of parts of speech of the target numeral as well as the just succeeding and preceding words

Finally, we develop a logistic regression model which takes the embeddings and engineered features as input and predicts whether a given numeral is in-claim or out-of-claim. The hyper-parameters\(^4\) of the logistic regression model are: C = 1.0, fit_intercept = True, intercept_scaling = 1, max_iter = 100, penalty = l2, solver = lbfgs, tolerance = 0.0001. The macro F1 score of this model is 0.8318.

5.4 Final system

The final system is an ensemble model. It selects results from the three subsystems (S1, S2 and S3) using majority voting. The macro F1 score of this model is 0.8671.

6 Experiments

We performed the experiments in four phases as mentioned below.

6.1 Defining the context window

At first, while exploring the data we noticed that 1867 and 285 financial texts from the training set and the validation set respectively had more than one target numeral. Thus, it was essential to define a context around the target numeral. We tried to extract the sentences in which the target numerals were present. This did not solve the problem as more than half of the data had multiple target numerals in a given sentence. We further tried to extract the portion of the text on which the target numeral was dependent using the dependency parser provided by spaCy.\(^5\) However, the performance did not improve. Finally, we performed several experiments by varying the context window size from 2 to 10. Context window of size k means we consider k words before and after the target numeral. Context window of size 8 gave us the best results.

6.2 Exploring various embeddings and classification algorithms

We explored various ways to numerically represent texts starting from TF-IDF to sentence transformer [27] based embeddings generated using BERT [12], RoBERTa [25] and FinBERT [2]. We further trained several classifiers over it. These classifiers included Logistic Regression, Random Forest [17], XG-Boost [7], etc. The performances of these models were not good enough. Thus, we added several engineered features as mentioned in Sect. 5. This improved the performance slightly but the improvement was not notably high.

6.3 Fine-tuning pre-trained transformer based models

We tried to fine-tune several variants of BERT [12] model for the task of classification. A FinBERT [2] model trained with batches of 256, for 15 epochs with a learning rate of 0.00002 gave the best performance. This model was trained on a context window of size 8.

6.4 Adding information regarding category and handcrafted features

We experimented by adding the categories to which the target numeral belonged as one hot vectors. We further

---

\(^3\) https://github.com/microsoft/Recognizers-Text.

\(^4\) https://scikit-learn.org/stable/modules/generated/sklearn.linear_ model.LogisticRegression.html.

\(^5\) https://spacy.io/.
engineered several features as mentioned in Sect. 5.3. These actions improved the macro F1 score to 0.8315 and 0.8318 respectively.

6.5 Ensembling individual models

Finally, we tried to combine outputs of the individual models using majority voting. On combining the individual models which are described in Sect. 5, the macro F1 score improved from 0.8585 to 0.8671.

6.6 Implementation details

These experiments were performed in a node having Nvidia Tesla V100 GPU with 32 GB RAM. We used Python (3.7) for all the computations. The main libraries used here includes PyTorch, SentenceTransformers, pandas, NumPy, scikit-learn, and Microsoft recognizers-text-number.

7 Results and discussion

We present the overall results in Table 2. We observe that machine learning based classifiers built with TF-IDF (with n-grams ranging from 1 to 4 and ignoring terms with document frequency strictly lower than 0.0005) based features (Sl. No. 1 to 3) did not perform as good as those which were built with FinBERT embeddings as features (Sl. No. 4 to 6). We tried extracting the portion of the financial text on which the target numeral was dependent. We further fine-tuned a FinBERT model using only the words on which the target numeral was dependent. This did not yield any improvement in the model performance (Sl. No. 7, Macro F1 score = 0.7250). However, on adding handcrafted engineered features and using context words within a window of 6 for fine-tuning the FinBERT model, the Macro F1 score improved to 0.8244 (Sl. No. 8). On adding information relating to categories as one hot vectors the F1 score further improved to 0.8315 (Sl. No. 9). Details regarding models S1, S2, S3 and their ensemble have already been mentioned in Sect. 5. The ensemble model (Sl. No. 14) performed the best (Macro F1 score = 0.8671 on validation set, 0.8473 on the test set). This is a significant improvement over the existing baseline CapsNet [9] (Sl. No. 10, Macro F1 score = 0.5736 on the test set). The results on the test set have been provided by the organizers in the paper [11].

Next, we evaluate the performance of the ensemble model across different categories. We present this in Table 3. It is interesting to note that the model performs well for almost all categories except ‘product number’ and ‘date’. This is because the training set did not have a single in-claim instance of the category ‘date’ and only 9 such instances of the category ‘product number’.

7.1 Ablation study

We conduct a detailed ablation study to assess the importance of each component present in the ensemble model. We present the results of this in Table 4. We observe that the ensemble model performs better than the constituent models. While testing the hypothesis that the ensembled model is better than S1, we obtained a p-value of 0.18. We modified S3 by removing engineered features and considered only the largest sub-word token of the target numeral. It resulted in the reduction of macro F1-score. This proves the effectiveness of every part of the final model. We further tried varying the context window size. We conclude that the context window of size 8 gives the best performance.

7.2 Qualitative error analysis

Subsequently, we performed a qualitative evaluation for instances where our model made wrong predictions. We
present a sample of it in Table 5. We observe that more than 66% of the miss-classified target numerals have a dollar (‘$’) symbol and 17% of them have a percentage (‘%’) symbol associated. Microsoft digit recognizer was able to effectively put these instances into categories ‘currency’ and ‘percentage’ respectively. Thus, creating a classifier to first predict the categories and then training separate classifiers for each category may have helped in achieving better performance.

### Table 3: Category wise performance of the ensemble model

| Category           | Micro-F1 | Macro-F1 |
|--------------------|----------|----------|
| Product number     | 0.9615   | 0.4902   |
| Date               | 0.9864   | 0.4966   |
| Quantity absolute  | 0.9426   | 0.8002   |
| Money              | 0.8759   | 0.8214   |
| Relative           | 0.9788   | 0.8474   |
| Absolute           | 0.9059   | 0.8590   |
| Change             | 0.8810   | 0.8776   |
| Other              | 1        | 1        |
| Quantity relative  | 1        | 1        |

### Table 4: Ablation study

| Model                                          | Macro-F1 | Micro-F1 |
|------------------------------------------------|----------|----------|
| S1 (CW = 8, only)                              | 0.8585   | 0.6345   |
| S2 (CW = 6, only)                              | 0.8439   | 0.6603   |
| S3 (only)                                      | 0.8318   | 0.6646   |
| S3 (-EF)                                       | 0.8238   | 0.7900   |
| S3 (-EF, only largest token)                   | 0.7934   | 0.7763   |
| FinBERT classifier (CW = 4)                    | 0.8408   | 0.7994   |
| FinBERT classifier (CW = 5)                    | 0.8318   | 0.7250   |
| FinBERT classifier (CW = 7)                    | 0.8381   | 0.8244   |
| FinBERT classifier (CW = 9)                    | 0.8247   | 0.8262   |
| FinBERT classifier (CW = 10)                   | 0.8407   | 0.8585   |
| Ensemble S1, S2, S3                            | 0.8671   | 0.9479   |

CW context window, EF engineered features

In future, we would like to build a custom tokenizer that will tokenize other words into sub-tokens while keeping the target numeral as it is. We also want to experiment by changing the ensembling method from majority voting to a meta-classifier. Furthermore, a Convolutional Neural Network (CNN) or a Long Short Term Memory (LSTM) model trained using the context embeddings may yield better results. Another interesting direction would be to experiment if we could leverage knowledge from a similar kind of datasets [9, 10] available in other languages like Chinese. Finally, we want to improve the algorithm being

### Table 5: Qualitative analysis of miss-classified instances

| Paragraph                                                                 | Numeral | Target (claim) | Predicted |
|---------------------------------------------------------------------------|---------|----------------|-----------|
| Fourth quarter free cash flow was $1.2 billion taking our full year cash generation to $4.4 billion | $4.4    | 0              | 1         |
| On the synergies front Collins Aerospace team is on track to deliver approximately $150 million of cost synergies this year and we still see at least $4.45 billion of total cost synergy potential | $500    | 0              | 1         |
| Thanks Matt. Verizon's strategic priorities for 2019 are clear. I have outlined $4.45 priorities with my team that focus on our customers financial performance 5G leadership our valued employees and Verizon's role in creating benefits for our society | 5       | 1              | 0         |
| Additionally United Technologies launched a hybrid-electric propulsion technology demonstrator program. This demonstrator program is expected to yield an average fuel savings for regional-sized aircraft of $4.30% and we’re targeting first flight within 3 years | 30%     | 1              | 0         |
used to extract words from the given texts on which the target numerals are dependent.

References

1. Aharoni E, Polnarov A, Laveen T, Hershovich D, Levy R, Rinott R, Gutfreund D, Slonim N (2014) A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics. In: Proceedings of the first workshop on argumentation mining, Baltimore, Maryland. Association for Computational Linguistics, pp 64–68. https://doi.org/10.3115/v1/W14-2109. https://aclanthology.org/W14-2109

2. Araci D (2019) Finbert: Financial sentiment analysis with pre-trained language models

3. Azzi AA, Bouamor H (2019) Fortial1@ the ntcir-14 finnum task: enriched sequence labeling for numeral classification. In: Proceedings of the 14th NTCIR conference on evaluation of information access technologies, pp 526–538. http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings14/pdf/ntcir/02-NTCIR14-FINNUM-AzziA.pdf

4. Bar-Haim R, Edelstein L, Jochim C, Slonim N (2017) Improving claim stance classification with lexical knowledge expansion and context utilization. In: Proceedings of the 4th workshop on argument mining, Copenhagen, Denmark, Sept. Association for Computational Linguistics, pp 32–38. https://doi.org/10.18653/v1/W17-5104. http://aclanthology.org/W17-5104

5. Botnevik B, Sakariassen E, Setty V (2020) BRENDA: browser extension for fake news detection. Association for Computing Machinery, New York, pp 2117–2120. 10(1145/3397271:3401396

6. Chakrabarty T, Hidey C, McKeown K (2019) IMHO fine-tuning improves claim detection. In: Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: human language technologies, vol 1 (Long and Short Papers), Minneapolis, Minnesota, June. Association for Computational Linguistics, pp 558–563. https://doi.org/10.18653/v1/N19-1054. https://aclanthology.org/N19-1054

7. Chen T, Guestrin C (2016) XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, KDD ‘16, New York, ACM, pp 785–794. https://doi.org/10.1145/2939672.2939785

8. Chen YY, Liu CL (2020) Mig at the ntcir-15 finnum-2 task: use the transfer learning and feature engineering for numeral attachment task. In: Proceedings of the 15th NTCIR conference on evaluation of information access technologies, https://research.nii.ac.jp/ntcir/workshop/OnlineProceedings15/pdf/ntcir/02-NTCIR15-FINNUM-ChenY.pdf

9. Chen CC, Huang HH, Chen HH (2020) Numclaim: investor’s fine-grained claim detection. In: Proceedings of the 29th ACM international conference on information & knowledge management, CIKM ‘20, New York. Association for Computing Machinery, pp 1973–1976. https://doi.org/10.1145/3340531.3412100

10. Chen CC, Huang HH, Chen HH (2021) Evaluating the rationales of amateur investors. In: Proceedings of the web conference 2021, WWW ’21, New York. Association for Computing Machinery, pp 3987–3998. https://doi.org/10.1145/3442381.3449664

11. Chen CC, Huang HH, Huang YL, Takamura H, Chen HH (2022) Overview of the ntcir-16 finnum-3 task: investor’s and manager’s fine-grained claim detection. In: Proceedings of the 16th NTCIR conference on evaluation of information access technologies, Tokyo, Japan (forthcoming)

12. Devlin J, Chang M.-W, Lee K, Toutanova K (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), Minneapolis. Association for Computational Linguistics, pp 4171–4186. https://doi.org/10.18653/v1/N19-1423. https://aclanthology.org/N19-1423

13. Gencheva P, Nakov P, Márquez L, Barrón-Cedeño A, Koychev I (2017) A context-aware approach for detecting worth-checking claims in political debates. In: Proceedings of the international conference recent advances in natural language processing, RANLP 2017, Varna, Bulgaria, Sept. INCOMA Ltd, pp 267–276. https://doi.org/10.26615/978-954-452-049-6-037

14. Ghosh S, Naskar SK (2022a) Fincat: Financial numeral claim analysis tool. In: Companion proceedings of the web conference 2022 (WWW ’22 Companion), New York. Association for Computing Machinery. https://doi.org/10.1145/3487553.3524635. https://arxiv.org/abs/2202.00631

15. Ghosh S, Naskar SK (2022b) Lipi at the ntcir-16 finnum-3 task: ensemble transformer based models to detect in-claim numerals in financial conversations. In: Proceedings of the 16th NTCIR conference on evaluation of information access technologies, Tokyo Japan (forthcoming)

16. Hassan N, Tremayne M, Arslan F, Li C (2016) Comparing automated factual claim detection against judgments of journalism organizations. In: Computation+ journalism symposium, pp 1–5. https://journalism.stanford.edu/cj2016/files/Comparing

17. Ho TK (1995) Random decision forests. In: Proceedings of 3rd international conference on document analysis and recognition, vol 1, pp 278–282. https://doi.org/10.1109/ICDAR.1995.598994. https://ieeexplore.ieee.org/document/598994

18. Howard J, Ruder S (2018) Universal language model fine-tuning for text classification. In: Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: long papers), Melbourne. Association for Computational Linguistics, pp 328–339. https://doi.org/10.18653/v1/P18-1031. https://aclanthology.org/P18-1031

19. Konstantinovskiy L, Price O, Babakar M, Zubiaga A (2021) Toward automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection. In: Digital threats: research and practice. https://doi.org/10.1145/3412869

20. Levy R, Bilu Y, Hershovich D, Aharoni E, Slonim N (2014) Context dependent claim detection. In: Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers, Dublin. Dublin City University and Association for Computational Linguistics, pp 1489–1500. https://aclanthology.org/C14-1141

21. Levy R, Gretz S, Szajdier B, Hummel S, Aharonov R, Slonim N (2017) Unsupervised corpus-wide claim detection. In: Proceedings of the 4th workshop on argument mining, Copenhagen. Association for Computational Linguistics, pp 79–84. https://doi.org/10.18653/v1/W17-5110. https://aclanthology.org/W17-5110

22. Liang CC, Su KY (2019) Asnlu at the ntcir-14 finnum task: incorporating knowledge into dnn for financial numeral classification. In: Proceedings of the 14th NTCIR conference on evaluation of information access technologies, vol 92. https://research.nii.ac.jp/ntcir/workshop/OnlineProceedings14/pdf/ntcir/04-NTCIR14-FINNUM-LiangC.pdf

23. Lippi M, Torroni P (2015) Context-independent claim detection for argument mining. In: Proceedings of the 24th international conference on artificial intelligence, IJCAI’15. AAAI Press,
pp 185–191. https://www.ijcai.org/Proceedings/15/Papers/033.pdf

24. Lippi M, Lagioia F, Contissa G, Sartor G, Torroni P (2018) Claim detection in judgments of the EU court of justice. In: Pagallo U, Palmirani M, Casanovas P, Sartor G, Villata S (eds) AI approaches to the complexity of legal systems. Springer, Cham, pp 513–527

25. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Roberts SV (2019) A robustly optimized bert pretraining approach

26. Reddy RG, Chinthakindi S, Wang Z, Fung YR, Conger KS, Elsayed AS, Palmer M, Ji H (2021) Newsclaims: a new benchmark for claim detection from news with background knowledge. https://blender.cs.illinois.edu/paper/newsclaims2022.pdf

27. Reimers N, Gurevych I (2019) Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), Hong Kong. Association for Computational Linguistics, pp 3982–3992. https://doi.org/10.18653/v1/D19-1410. https://aclanthology.org/D19-1410

28. Spark A (2019) Brnir at the ntcir-14 finnum task: scalable feature extraction technique for number classification. In: Proceedings of the 14th NTCIR conference on evaluation of information access technologies. https://research.nii.ac.jp/ntcir/workshop/OnlineProceedings14/pdf/ntcir/03-NTCIR14-FINNUM-SparkA.pdf

29. Sundriyal M, Singh P, Akhtar MS, Sengupta S, Chakraborty T (2021) DESYR: definition and syntactic representation based claim detection on the web. Association for Computing Machinery, New York, pp 1764–1773. 10(1145/3459637):3482423

30. Wallace E, Wang Y, Li S, Singh S, Gardner M (2019) Do NLP models know numbers? probing numeracy in embeddings. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), Hong Kong. Association for Computational Linguistics, pp 5307–5315. https://doi.org/10.18653/v1/D19-1534. https://aclanthology.org/D19-1534

31. Wright D, Augenstein I (2020) Claim check-worthiness detection as positive unlabelled learning. In: Findings of the association for computational linguistics: EMNLP 2020, Online. Association for Computational Linguistics, pp 476–488. https://doi.org/10.18653/v1/2020.findings-emnlp.43. https://aclanthology.org/2020.findings-emnlp.43