# Empowering the Fact Checkers: Automatic Identification of Claim Spans on Twitter

**Joseph Picchi, Shiyao Guo, Yuning Yang**
University of California, Los Angeles
jpicchi22@g.ucla.edu, sguo18@g.ucla.edu, yuningyang@g.ucla.edu

## Abstract

During the COVID pandemic, the prevalence of fake news and misinformation caused fear, anxiety, and social unrest, underscoring the need for wide-scale fact checking. We seek to assist human fact checkers by automatically extracting spans of text that represent claims in COVID-related tweets. To do so, we developed large language models to perform automatic claim span detection. We attemped the task with several models. Our encoder models (Bert, DeBERTa, DistilBert, and BERT+LSTM) yielded state-of-the art results for the dataset provided, while our encoder-decoder models (T5 and BART) achieved more modest performance. The superior performance of DeBERTa relative to all of the other models we tested is consistent with previous literature addressing this task and datset.

## 1 Introduction

The rise of social media and internet-based communications mediums has accelerated the spread of (mis)information. Prior research finds that the often-novel nature of fake news encourages users to share it more frequently than real news, causing misinformation to spread more rapidly than true information online (Vosoughi et al., 2018).

Fact checkers aim to combat misinformation. Most claim verification is performed manually, making it imperative to assist human fact-checkers as much as possible (Guo et al., 2022). Our research seeks to assist them by training natural language models to automatically identify claims made in social media posts.

In particular, we fine-tune encoder and encoder-decoder models on a Twitter corpus of 7.5K tokenized tweets annotated with the corresponding claims they make. An example is shown in figure-1. Each example can have one or more claims.

The task was originally defined by (Sundriyal et al., 2022), which created the dataset, formalized the task, and defined a claim as "an assertion that deserves our attention" (Toulmin, 2003) and "is the key component of any argument" (Trautmann et al., 2020). The dataset was then posted publicly in a Codalab challenge[1].


Figure 1: A COVID-related tweet with the claim span highlighted in blue. See appendix figure- for more examples

### 1.1 Previous Work

Previous work has focused on 3 main claim-oriented research areas: detection (Gupta et al., 2021), check worthiness (Wright and Augenstei, 2020), and verification (Soleimani et al., 2020). Within the claim detection area, researchers have developed course-grained sentence-level claim detection models (Chakrabarty et al. (2019); (Gupta et al., 2021; Chakrabarty et al., 2019). Few works have focused on fine-grained span detection within s entences for COVID-related tweets.

Previous conferences (Nakov et al., 2021) have underscored the importance of claim span detection in fact checking by identifying 4 fundamental steps in the pipeline of end-to-end AI fact checking: (1) identifying claims, (2) detecting previously fact-checked claims, (3) retrieving evidence, and (4) verifying the claim. Within these areas, most research has focused on downstream tasks (2)-(4), such as (Shaar et al., 2020) that takes an input sentence and generates a ranked list of previously fact-checked claims to verify it.

In the area of span identification, researchers have developed fine-grained models to extract spans in other domains like toxic language (Pavlopoulos et al., 2021), propaganda (Martino

[1] "Empowering the Fact-checkers! Automatic Identification of Claim Spans on Twitter"

et al., 2020), and hate speech (Mathew et al., 2021). Such studies have tested various methods, including transformers, data augmentation, and ensembling.

For our specific task of fine-grained span identification for COVID tweets, encoder-only models were developed by the paper that developed the dataset and defined the task (Sundriyal et al., 2022), including DistilBERT, BERT, SpanBERT, RoBERTA, DABERTa, and a BiLSTM. Our research builds upon this progress by fine-tuning more recent checkpoints of BERT and its variants, and by attempting the task with a BERT+LSTM model and several encoder-decoder models.

## 2 Methods

### 2.1 Encoder-Decoder Models

Although the claim span identification task is more naturally formulated as a token identification task, (Raffel et al., 2020) have demonstrated that many NLP tasks can be reformulated in a sequence to sequence format. So we designed multiple seq2seq input-output formats for our task and used them to fine tune T5 and BART. The motivation was to test whether transfer learning from T5 could improve predictions for claim span detection.

We fine-tuned T5 with 2 different input formats. The first format structures claim span detection as a Q&A task to take advantage of T5's text extraction capabilities in its pre-trained Q&A task. The second format structures claim span detection as a new task type that wasn't seen in pre-trianing. Inputs are structured as "Extract the claims: [tweet text]". Both input formats have the same output format: claims are listed as consecutive sentences.

BART was fine-tuned only on the second input format, since it's not pre-trained on T5's Q&A task.

To evaluate the sequence-to-sequence models, we used Rouge 1, 2, L, and Lsum metrics that score predictions based on the number of shared words with the example label (Lin, 2004). Analysis of the rougeL and Lsum scores also ensures that the overlapping words occur in the same consecutive sequence in both the prediction and the label, thus ensuring that a coherent subset or superset of the target claim is properly extracted.

### 2.2 Encoder Models

This is essentially a name entity recognization task, which is also known as NER. NER requires BIO tagging for each token, but the training and test data do not have BIO tagging. So we preprocessed the training and test data by adding BIO tagging to each token. For token that is the start of the claim, we labelled it using B. For token that is inside the claim, we labelled it using I. For all the rest of the tokens, we label them as O.

After data preprocessing, we created a function called convert_to_input, which returns the input to different BERT models. We used a tokenizer to tokenize each token and then converted them into ids. We also set the attention mask to focus on tokens with B and I tagging.

We tried several BERT models, namely BERT, DistilBert, and DABerta. For optimizer, we used AdamW. There are 3 inputs into the models, namely tokens in the form of ids, the labels, and the attention masks.

In addition to the BERT models, we also experimented with a hybrid LSTM+BERT model. In this model, we first used BERT to generate context-aware embeddings for each token in the input. These embeddings were then passed to a bidirectional LSTM, which processed the sequence of embeddings over time. The LSTM, being a type of recurrent neural network, is effective at capturing long-term dependencies in sequence data, which is beneficial in NER tasks where the classification of a word can depend on the words that precede it in a sentence.

However, LSTMs can suffer from a short-sighted vantage wherein they place greater consideration upon prior outputs that were generated more recently when predicting new outputs. To mitigate this, we used the bidirectional variant of LSTM, which processes the data in both directions (forward and backward) and can capture information from both past and future states. The goal of this hybrid LSTM+BERT model was to leverage the strengths of both models and provide a robust approach to the NER task.

## 3 Results

### 3.1 Encoder-Decoder Models

Table-1 indicates that T5 performed nearly equivalently on both input formats, with slightly better performance on the Q&A format across all metrics. This indicates that Q&A pre-training of T5 yielded minimal benefits for this task.

BART performed negligibly worse than T5 on the same non-Q&A input format. Their near-equivalent performance indicates that transfer learn-

| Metric | T5 QA | T5 | BART |
|---|---|---|---|
| Rouge1 Fmeasure | 0.780 | 0.775 | 0.775 |
| RougeL Fmeasure | 0.776 | 0.772 | 0.771 |

Table 1: Sequence-to-sequence model performance: All 3 encoder-decoder models achieved approximately equivalent performance, irregardless of architecture or input formulation. Note that "T5" is the T5 model fine-tuned without the Q&A input formulation. See appendix table-9 for the full results.

| Metric | 1 Claim | 2+ Claims |
|---|---|---|
| Rouge1 Fmeasure | 0.829 | 0.600 |
| Rouge1 Precision | 0.875 | 0.876 |
| Rouge1 Recall | 0.817 | 0.481 |
| RougeL Fmeasure | 0.826 | 0.592 |
| RougeL Precision | 0.872 | 0.864 |
| RougeL Recall | 0.814 | 0.475 |

Table 2: One claim vs. multiple claims: The T5 Q&A model achieves significantly higher Fmeasure and recall scores on examples where the tweet makes one claim compared to examples where the tweet makes more than one claim. See appendix table-10 for the full results.

ing with T5 produced almost no performance boost for this task.

All 3 encoder-decoder models performed considerably worse than BERT, indicating that any benefits of transfer learning were outweighed by the suboptimal sequence-to-sequence formulation of the task. The lower performance can be explained by the difference in objective between encoder and encoder-decoder models.

Encoder-decoder models optimize the probability of each output token given the input sequence and preceding output tokens. This objective is suboptimal because our task seeks to classify each token as a claim or non-claim solely based on the input sequence and irrespective of other tokens previously outputted by the decoder. This is particularly erroneous because some tweets have multiple claim spans. Since each claim is independent, tokens outputted for the second claim should not probabilistically depend on tokens outputted for the first claim. Table-2 evidences this explanation by showing that the T5 model achieved much higher F1 scores on tweets with one claim compared to tweets with multiple claims.

In contrast, the encoder objective optimizes the contextual representation of each token. This is more desirable for our task because it optimizes the independent classification of each token as a claim

| **Label:** |
|---|
| US & China Collaborated to Make a Deadly #Coronavirus. Dr Fauci knew it was being developed as a #bioweapon. |
| **Prediction:** |
| US & China Collaborated to Make a Deadly #Corona |

Table 3: Labels vs predictions for multi-claim tweets: T5 consistently predicts a subset of the claim text from the labels for examples in which the tweet makes more than one claim. See appendix table-11 for more examples.

or non-claim based on the input sequence. This explains the superior performance of our BERT models.

Interestingly, table-1 shows that the Rouge1 Fmeasure is approximately equivalent to the RoughL Fmeasure for all models. This indicates that nearly all words shared between the model predictions and reference labels appear consecutively in the prediction text. Thus, the model is at least able to extract a subset or superset of the actual claim text, as opposed to randomly-sequenced words that happen to appear in the reference text.

Lastly, table-2 indicates that the Rouge precision was much greater than recall for examples with multiple claims, whereas precision and recall were approximately equivalent for examples with one claim. This indicates that the ratio of prediction-label shared words to prediction words is much greater than the ratio of prediction-label shared words to label words. Thus, for examples with multiple claims, T5 routinely predicts a subset of the claim text in the labels. We verify this by manually examining poorly predicted examples that make multiple claims. One such example is shown in table-3. Moreover, the subset that T5 predicts typically begins at the start of the label. This further evidences the fact that the learning objective of encoder-decoder models is suboptimal for our task because each claim in the same tweet is independent of the others. Thus, tokens outputted for the second claim should not depend on tokens outputted earlier for the first claim.

### 3.2 Encoder Models

We used f1 score, precision, and recall as metrics to measure the performance of our model. Since the test dataset is not provided and the only two datasets available are train dataset and dev dataset,

| Model | Precision | Recall | F1 |
|---|---|---|---|
| BERT | 0.998 | 0.998 | 0.998 |
| DistilBERT | 0.997 | 0.997 | 0.997 |
| DeBERTa | 0.999 | 0.999 | 0.999 |
| BERT+LSTM | 0.760 | 0.760 | 0.760 |

Table 4: Encoder model performance: The LSTM detracted from BERT's predictions, and all other BERT models achieved state-of-the-art performance for the given dataset.

we used dev dataset as our test dataset. All of our BERT models have really high F1 score, precision, and recall. But the best performing model is DABerta, which correspond to the paper's finding that DABerta yielded the best results in this automatic identification of claim spans on twitter task. We tuned the hyper parameters and found out that setting the learning rate to 1e-5 and setting the batch size to 32 had the best performance.

We also experimented with a hybrid LSTM+BERT model. However, this model achieved a lower F1 score of 0.7601. This suggests that the addition of the LSTM layer did not improve performance for this particular task. In fact, it may have introduced additional complexity that hindered the model's ability to accurately identify claim spans.

While BERT is able to consider the context from both directions for each token, LSTM processes the sequence in a linear fashion. This difference in handling sequence data might have contributed to the lower performance of the LSTM+BERT model. Additionally, the LSTM layer adds more parameters to the model, which could make the model more prone to overfitting, especially if the amount of training data is limited.

Despite the lower performance of the LSTM+BERT model compared to the BERT and DeBerta models, it's worth noting that an F1 score of 0.7601 is still reasonably good. The LSTM+BERT model could potentially be improved with further tuning and optimization. Future work could explore different ways of combining BERT and LSTM, or investigate other types of recurrent layers or architectures.

## 4 Conclusion

In this study, we explored various models for the task of Claim Span Detection, aiming to accurately identify the beginning and end of a claim within a sentence. Our models included both sequence-to-sequence models (T5 and BART) and encoder models (BERT, DABerta, DistilBert, and a hybrid LSTM+BERT model).

Our results showed that the encoder models generally outperformed the sequence-to-sequence models. This is likely due to the fact that the encoder models optimize the contextual representation of each token, which is more suitable for our task of classifying each token as a claim or non-claim based on the input sequence. On the other hand, sequence-to-sequence models optimize the probability of each output token given the input sequence and preceding output tokens, which is not ideal for our task as it does not consider the independence of each claim in the same tweet.

Among the encoder models, DABerta achieved the highest performance, with an F1 score close to 0.99. This aligns with existing literature that suggests DABerta's superior performance in the task of automatic identification of claim spans on Twitter. The hybrid LSTM+BERT model, while not performing as well as the other encoder models, still achieved a respectable F1 score of 0.7601, suggesting potential for further optimization and tuning.

We release our code[2] for the purposes of reproducibility, demonstration, and further experimentation.

### 4.1 Future Work

While our models achieved high performance, there is still room for improvement and exploration. For the sequence-to-sequence models, future work could investigate different input-output formats or explore other types of sequence-to-sequence architectures. For the encoder models, future work could explore different ways of combining BERT and LSTM, or investigate other types of recurrent layers or architectures.

Furthermore, the performance of the models could potentially be improved by fine-tuning the hyperparameters, such as the learning rate and batch size. It would also be interesting to investigate the use of other optimization algorithms besides AdamW.

## Acknowledgements

Our team would like to acknowledge the teaching staff of the UCLA COM SCI 263 course for

[2] NLP-claim-span-detection

helping us learn the skills and knowledge demonstrated in this project. This includes instructor Kai-Wei Chang and teaching assistants Tanmay Parekh, Elaine Wan, and Masoud Monajatipoor.

We also acknowledge (Sundriyal et al., 2022) for laying the groundwork and creating the dataset for this task.

## References

Tuhin Chakrabarty, Christopher Hidey, and Kathy McKeown. 2019. Imho fine-tuning improves claim detection. *Association for Computational Linguistics*, 1:558—-563.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *MIT Press Direct*, (10):178–206.

Shreya Gupta, Parantak Singh, Megha Sundriyal, Md. Shad Akhtar, and Tanmoy Chakraborty. 2021. Lesa: Linguistic encapsulation and semantic amalgamation based generalised claim detection from online content. *Association for Computational Linguistics*, pages 3178–3188.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Association for Computational Linguistics*, Text Summarization Branches Out(1):74—-81.

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. Semeval-2020 task 11: Detection of propaganda techniques in news articles. *International Committee for Computational Linguistics*, Proceedings of the Fourteenth Workshop on Semantic Evaluation:1377—-1414.

Binny Mathew, Punyajoy Saha, Chris Biemann Seid Muhie Yimam, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17):14867—-14875.

Preslav Nakov, David Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barrón-Cedeño, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. 2021. Automated fact-checking for assisting human fact-checkers. *arXiv*.

John Pavlopoulos, Jeffrey Sorensen, Léo Laugier, and Ion Androutsopoulos. 2021. Semeval-2021 task 5: Toxic spans detection. *Associa- tion for Computational Linguistics*, Proceedings of the 15th International Workshop on Semantic Evalua- tion (SemEval-2021):59—-69.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

Shaden Shaar, Nikolay Babulkov, Giovanni Da San Martino, and Preslav Nakov. 2020. That is a known lie: Detecting previously fact-checked claims. *Association for Computational Linguistics*, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics:3607—-3618.

Amir Soleimani, Christof Monz, and Marcel Worring. 2020. Bert for evidence retrieval and claim verification. *Advances in Information Retrieval*, 12036(359).

Megha Sundriyal, Atharva Kulkarni, Vaibhav Pulastya, Md Shad Akhtar, and Tanmoy Chakraborty. 2022. Empowering the fact-checkers! automatic identification of claim spans on twitter. *arXiv*.

Stephen E Toulmin. 2003. The uses of argument. *Cambridge university press*.

Dietrich Trautmann, Johannes Daxenberger, Christian Stab, Hinrich Schütze, and Iryna Gurevych. 2020. 15th international workshop on semantic evalua- tion (semeval-2021). *Asso- ciation for Computational Linguistics*, pages 521–526.

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science*.

Dustin Wright and Isabelle Augenstei. 2020. Claim check-worthiness detection as positive unlabelled learning. *Association for Computational Linguistics*, pages 476–588.

## Appendix

## A Task Structure

This section contains additional information about the structure of claim span identification task and the nature of the inputs that our models processed.

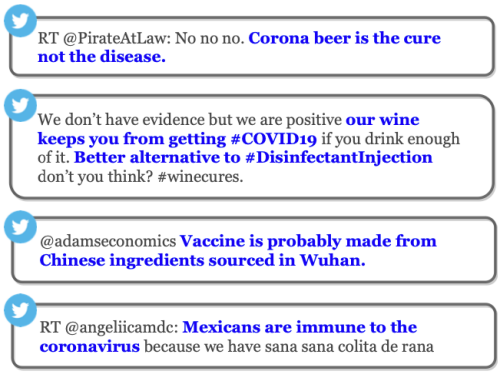


Figure 2: COVID-related tweets from the training dataset with claim spans highlighted in blue.

| |
|---|
| **Tokens:** ['If', ' you', ' swam', ' in', ' buck-eye', ' lake', ' as', ' a', ' kid', ' you', ' are', ' immune', ' to', ' the', ' coronavirus']<br>**Start Indices:** [0]<br>**End Indices:** [14] |
| **Tokens:** ['RT', ' FirebaughNorman', ': Tom', ' Cotton', ' was', ' right', ': The', ' Wuhan', ' virus', ' probably', ' came', ' from', ' a', ' bioweapons', ' lab']<br>**Start Indices:** [2, 6]<br>**End Indices:** [5, 14] |
| **Tokens:** ['In', ' all', ' fairness', ', injecting', ' #lysol', ' or', ' #bleach', ' will', ' protect', ' you', ' from', ' #COVID19', ' for', ' the', ' rest', ' of', ' your', ' life']<br>**Start Indices:** [3]<br>**End Indices:** [11] |
| **Tokens:** ['Top', ' 10', ' Trump', ' quotes', ':\n1', '. #coronavirus', ' is', ' fake', '"\n2', '. "I', ' am', ' the', ' chosen', ' one', '"\n3', '. "I', ' am', ', like', ', really', ' smart', '"\n4', '. "I', ' call', ' her', ' Pocahontas', '"\n5', '. "Impeachment', ' is', ' a', ' hoax', '"\n6', '. "No', ' quid', ' pro', ' quo', '"\n7', '. "No', ' collusion', '"\n8', '. "I', ' am', ' a', ' stable', ' genius', '"\n9', '. "Climate', ' change', ' is', ' a', ' hoax', '"\n10', '. "Windmills', ' because', ' cancer']<br>**Start Indices:** [5, 9, 21, 26, 39, 45]<br>**End Indices:** [7, 13, 24, 29, 43, 49] |

Table 5: Raw inputs: examples in the train and dev datasets consist of a tokenized tweet, along with indices indicating the tokens at which each claim span begins and ends.

## B Encoder Results

| Tagging | Precision | Recall | F1 |
|---|---|---|---|
| B | 0.97 | 0.94 | 0.96 |
| I | 0.99 | 1.00 | 0.99 |
| O | 1.00 | 1.00 | 1.00 |

Table 6: BERT performance

| Tagging | Precision | Recall | F1 |
|---|---|---|---|
| B | 0.95 | 0.94 | 0.94 |
| I | 0.99 | 0.99 | 0.99 |
| O | 1.00 | 1.00 | 1.00 |

Table 7: DistilBERT performance

| Tagging | Precision | Recall | F1 |
|---|---|---|---|
| B | 0.98 | 0.96 | 0.97 |
| I | 0.99 | 1.00 | 1.00 |
| O | 1.00 | 1.00 | 1.00 |

Table 8: DeBERTa performance

## C Encoder-Decoder Results

This section contains additional tables related to the encoder-decoder experiments.

| Metric | T5 QA | T5 | BART |
|---|---|---|---|
| Rouge1 Fmeasure | 0.780 | 0.775 | 0.775 |
| Rouge1 Precision | 0.875 | 0.872 | 0.869 |
| Rouge1 Recall | 0.745 | 0.742 | 0.740 |
| Rouge2 Fmeasure | 0.755 | 0.750 | 0.749 |
| Rouge2 Precision | 0.855 | 0.850 | 0.847 |
| Rouge2 Recall | 0.722 | 0.719 | 0.715 |
| RougeL Fmeasure | 0.776 | 0.772 | 0.771 |
| RougeL Precision | 0.870 | 0.868 | 0.864 |
| RougeL Recall | 0.742 | 0.738 | 0.736 |
| RougeLs Fmeasure | 0.776 | 0.772 | 0.771 |
| RougeLs Precision | 0.870 | 0.868 | 0.864 |
| RougeLs Recall | 0.742 | 0.739 | 0.736 |

Table 9: Sequence-to-sequence model performance: Performance metrics for our encoder-decoder models for the entire development data set. Note that "T5 QA" represents the T5 model that was fine-tuned using the Q&A input format and "T5" is the T5 model fine-tuned without the Q&A input format. "RougeLs" is an abbreviation for "RougeLsum". All encoder-decoder models achieved approximately equivalent performance on all metrics.

| Metric | 1 Claim | 2+ Claims |
|---|---|---|
| Rouge1 Fmeasure | 0.829 | 0.600 |
| Rouge1 Precision | 0.875 | 0.876 |
| Rouge1 Recall | 0.817 | 0.481 |
| Rouge2 Fmeasure | 0.809 | 0.555 |
| Rouge2 Precision | 0.859 | 0.840 |
| Rouge2 Recall | 0.798 | 0.440 |
| RougeL Fmeasure | 0.826 | 0.592 |
| RougeL Precision | 0.872 | 0.864 |
| RougeL Recall | 0.814 | 0.475 |
| RougeLsum Fmeasure | 0.826 | 0.592 |
| RougeLsum Precision | 0.872 | 0.864 |
| RougeLsum Recall | 0.814 | 0.475 |

Table 10: One claim vs. multiple claims: The T5 Q&A model achieves significantly higher Fmeasure and recall scores on development set examples where the tweet makes one claim compared to development set examples where the tweet makes more than one claim.

| |
|---|
| **Label:** |
| It is impossible to get equal pay without the #ERA. It is impossible to shift the rape culture without #ERA. States allow police to rape someone in their custody & claim it was consensual. |
| **Prediction:** |
| It is impossible to get equal pay without the #ERA. It is impossible to shift the |
| **Label:** |
| People are dropping like flies. Wuhan is the epicenter for the worlds most dangerous pathogens. |
| **Prediction:** |
| People are dropping like flies. Wuhan is the epicenter for the worlds |
| **Label:** |
| Drink Lysol/Dettol to get cured frm Coronavirus Outbreak. BANG THALIS, LIT DIYAS & MOMBATTI TO GET CURED FRM CORONA VIRUS. |
| **Prediction:** |
| Drink Lysol/Dettol to get cured frm Coronavirus Outbreak |
| **Label:** |
| US & China Collaborated to Make a Deadly #Coronavirus. Dr Fauci knew it was being developed as a #bioweapon. |
| **Prediction:** |
| US & China Collaborated to Make a Deadly #Corona |

Table 11: Labels vs predictions for multi-claim tweets: T5 consistently predicts a subset of the claim text from the labels for examples in which the tweet makes more than one claim.