Six Approaches to Improve BERT for Claim Verification as Applied to the Fact Extraction and Verification Challenge (FEVER) Dataset

Jonathan Ling
Department of Management
Science and Engineering
Stanford University
Stanford, CA 94305
jonling@stanford.edu

Daniel Jun
Department of Management
Science and Engineering
Stanford University
Stanford, CA 94305
djun36@stanford.edu

Anica Oesterle
Department of Management
Science and Engineering
Stanford University
Stanford, CA 94305
oestea@stanford.edu

Abstract

BERT has been used in various research for fact extraction and verification tasks, such as tweet classification, hate speech detection and fake news detection. However, BERT suffers from various issues when applied to claim verification, which can help detect and classify misinformation. The goal of our project is to implement the BERT model on the FEVER (Fact Extraction and Verification) task, specifically for claim verification, as well as suggest and implement six improvement approaches to the original BERT model. We aim to gain valuable insights into the effectiveness of various model improvements for claim verification and hope to support the conquest to combat the spread of misinformation on the internet with our experiments. We conducted an end-to-end analysis of improvements on BERT for claim verification specifically for the FEVER task, from pre-processing evidence via data augmentation (synonym replacement and back-translation), changing the transformer settings (BERT vs DistilBERT and number of epochs), and post-processing its results neurally. Our modifications did not result in significant changes to the FEVER score and BERT baseline remained as the best performing model. Applying our neural aggregation layer, however, did improve performance on the DistilBERT model. This may be because BERT is a large model with a lot of pre-trained knowledge, and so our changes in the fine-tuning process and aggregation layer may not have a large impact on the model's performance as much as on the smaller DistilBERT model.

1 Introduction

The Internet provides a dangerous breeding ground for misinformation from unreliable sources. The FEVER (Fact Extraction and Verification) challenge aims to tackle the spread of misinformation by working on verifiable knowledge extraction with research teams all across the world in a workshop and shared task format. Models are trained and tested on the related FEVER dataset, which consists of 185,000 generated claims labelled as "SUPPORTS", "REFUTES" or "NOT ENOUGH INFO", based on the introductory sections of a 50,000 popular Wikipedia pages dump (Thorne u. a., 2018). Based on this data, the language model classifies the veracity of textual claims and extracts the correct evidence sentences necessary to support or refute the claims. One piece of evidence can contain several sentences that only if examined together result in the stated label - for example, for the claim "Oliver Reed was a film actor", one piece of evidence can be the set "Oliver Reed starred in the Gladiator", "Gladiator is film released in 2000". The FEVER leaderboard keeps track of each team's results on the FEVER score - the label accuracy conditioned on providing the correct evidence sentences. The current top score on the FEVER leaderboard is 75.87% (appendix A.1). Given a claim needs to be compared against an enormous amount of information in order to be verified, the computational challenge is massive. Therefore, the FEVER task is usually divided into a three-step pipeline: document retrieval, sentence retrieval, and claim verification. We aim to contribute to the important cause of tackling misinformation by further investigating the BERT transformer model (Devlin u. a., 2018) with several experiments to improve claim verification.

Primarosa (2020) uses a BERT model for each of steps two and three of the pipeline - evidence retrieval and claim verification. As we saw potential for further improvement to claim verification performance, we used Primarosa's implementation as a baseline model and experimented with several modifications to the fine-tuning process, including data augmentation and varying epoch numbers to avoid both underfitting (not enough epochs) and overfitting (too many epochs). We are also investigating the performance of using DistilBERT (Sanh u. a., 2020) on the task - a smaller, faster, cheaper and lighter version of BERT. Finally, Primarosa (2020) only uses a simple if-then logic to classify a claim based on the five retrieved possible evidence sentences without taking advantage of any synergistic information between them. Hence, we applied a neural aggregation layer based on Yoneda u. a. (2019) to combine this knowledge. Our contributions include:

- Implementing the BERT baseline model per Primarosa (2020), which uses the same high-level architecture as Soleimani u. a. (2019) for sentence retrieval and claim verification with document retrieval per Hanselowski u. a. (2018)
- Comparing our baseline results to BERT modifications: (1) implementing DistilBERT (Sanh u. a., 2020); (2) and (3) data augmentation via adding synonyms and back-translation over five languages to make retrieved sentences more robust; (4) and (5) amending the number of training epochs; (6) adding a neural aggregation layer to BERT

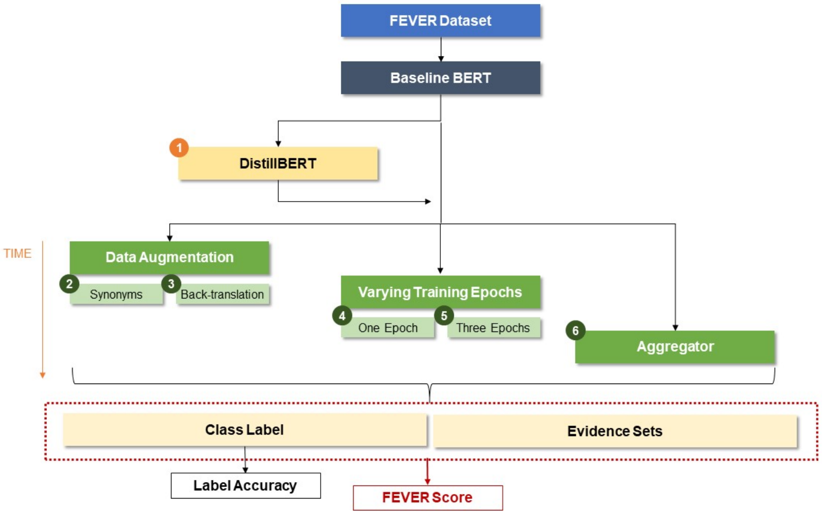


Figure 1: Summary of approaches to improve BERT for claim verification on the FEVER dataset

2 Related work

Both Soleimani u. a. (2019) and Primarosa (2020) use BERT in an evidence retrieval and claim verification pipeline on the FEVER dataset. The underlying task is to classify the correctness of textual claims and extract the correct evidence sentences required to support or refute the claims. Yoneda u. a. (2019) chose a different approach to the FEVER task. The team relies on a standard logistic regression model without transformers for sentence retrieval and the Enhanced Sequential Inference Model (ESIM) - a Natural Language Inference (NLI) Model - with a bidirectional LSTM for the claim verification task. In the last step, the NLI model is connected to an aggregation network, which aggregates the predicted NLI labels for each claim-evidence pair and outputs the final prediction ("aggregation stage"). This approach resulted in a FEVER score of 62.52% on the provisional test set and 65.41% on the development set - an improvement to the underlying baseline model. Hence, we were inspired to connect an aggregation network to our chosen baseline BERT model in order to assess whether aggregation improves not only the NLI model, but also BERT.

In order to not only improve the model itself, but also enhance the quality of the input data set, we were inspired by Wei und Zou (2020), as well as Longpre u. a. (2019). The former investigated synonym replacement by randomly choosing n words from a sentence that were not stop words and replaced these words with a randomly selected synonym. We applied this synonym replacement approach to our retrieved sentences in the FEVER pipeline. Moreover, Longpre u. a. (2019) as well as Yu u. a. (2018) enhanced their training data by translating the original sentences from English to another language and then back to English, which enhanced the number of training instances and diversified the phrasing. We emulated and extended this promising approach by translating from English to five different languages (German, French, Japanese, Hindi, Russian) with the goal to diversify the phrasing even further.

Many research teams have developed alternative and enhanced BERT models (e.g. RoBERTa, Liu u. a. (2019)). We chose to evaluate the DistilBERT model's performance on the FEVER task, as a comparison to Baseline BERT. DistilBERT was introduced by Sanh u. a. (2020) in order to tackle the challenge of operating large Natural Language Processing (NLP) models under constrained computational training or inference budgets. DistilBERT is a smaller, pre-trained general purpose language representation model with a smaller parameter count, which can be fine-tuned on a broad range of tasks (appendix A.2). Given the FEVER task's extremely high computational requirements, DistilBERT was a good fit for our improvement experiments.

3 Approach

3.1 Task

The "FEVER task" - classifying the correctness of textual claims - is approached in a three-step pipeline, consisting of 1) document retrieval, 2) sentence retrieval, and 3) claim verification (fig. 2). "Document retrieval" shortlists a set of documents, which could possibly contain relevant information to support or refute a claim, from the Wikipedia set. "Sentence retrieval" extracts five sentences out of the retrieved documents as potential evidence. Lastly, "claim verification" verifies the claim against the retrieved evidence sentences.

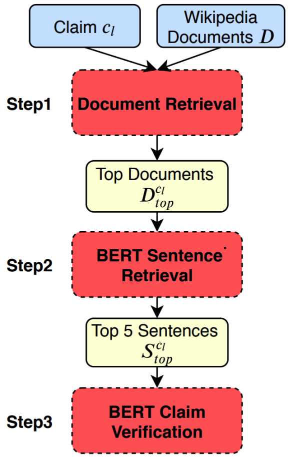


Figure 2: Three-step pipeline for evidence extraction and claim verification (Soleimani u. a., 2019)

3.2 Baseline model - BERT

As described in Soleimani u. a. (2019), the FEVER dataset provides N_D Wikipedia documents $D = \{d_i : i = 1, ..., N_D\}$. A document d_i consists of sentences $S^{d_i} = \{s^i_j : j = 1, ..., N_{S^{d_i}}\}$. The model's goal is two-fold: first, it has to classify the claim c_l for $l = 1, ..., N_C$ (where $N_C \approx$ 145,000 for the FEVER benchmark) as “SUPPORTS”, “REFUTES” or “NOT ENOUGH INFO”. Second, to consider a prediction true, a complete set of evidence $E^{c_l} = \{s^i_j\}$ has to be retrieved for the claim c_l. Claims labelled with “not enough info” do not have an evidence set.

3.2.1 Document retrieval

For this task, we ran the code from Hanselowski u. a. (2018) as used in the BERT implementation by Primarosa (2020). This uses the proposed entity linking approach for document retrieval in finding entities in the claims that match the titles of Wikipedia articles. The subsequent document retrieval component has three main steps: mention extraction, candidate article search, and candidate filtering.

- Mention extraction: AllenNLP’s constituency parser from Gardner u. a. (2019) is used for this first step to find entities of different categories. After the claim is parsed, every noun phrase is considered a potential entity mention.
- Candidate article search: Hanselowski u. a. (2018) use the MediaWiki API to search the Wikipedia database for the claim noun phrases extracted in task one. The top match of the API is the article whose title has the largest overlap with the query.
- Candidate filtering: As the MediaWiki API retrieves articles whose titles overlap the query, the resulting articles may have a longer or shorter title than the entity mentioned in the query. Hanselowski u. a. (2018) removed results that are no longer than the entity mentioned and do not overlap with the remaining claim. We collect a set of top documents $D^{c_l}_{\text{top}}$ for claim c_l.

3.2.2 Sentence retrieval and claim verification

Here, we use code from Primarosa (2020), with both of these steps using a BERT model each. The architecture of the BERT model follows that of Soleimani u. a. (2019) and is illustrated in appendix A.3. In the sentence retrieval step, for each claim c_l, all sentences S_{d_i} retrieved from the documents $D^{c_l}_{\text{top}}$ in the document retrieval step that match the claim c_l ($S^{c_l}_{\text{all}} = \{S_{d_i} | d_i \in D^{c_l}_{\text{top}}$) are scored, and the top five potential evidence sentences $S^{c_l}_{\text{top}}$ by this sentence score are retrieved. The training set consists of $\sim$145,000 claims for which this is done. Here, $S^{c_l}_{\text{all}}$ may or may not include the actual evidence sentences that are known from the ground truth labels. In the claim verification step, these top five potential evidence sentences $S^{c_l}_{\text{top}}$ for each claim are independently compared against the claim c_l and each is labeled. By aggregating these five individual labels, the final label is assigned (Primarosa, 2020).

3.3 Dataset

We worked with pre-trained models and did not need a pre-training dataset. For fine-tuning and evaluation, we used the FEVER dataset (Thorne u. a., 2018) due to its large size, text-only claims (no metadata), and live public leaderboard (Cocarascu, 2018). FEVER consists of 185,445 claims generated from altered sentences extracted from Wikipedia. Each claim is tied to a label and a list of evidence sets. The labels are one of “SUPPORTS”, “REFUTES” or “NOT ENOUGH INFO”, depending on what can be concluded from the Wikipedia data. Each evidence set is made up of one or more sentences that come from one or more Wikipedia articles. Any evidence set in the list of evidence sets for a claim can independently verify the claim.

3.4 Improvement 1 - DistilBERT

DistilBERT is a smaller, faster and lighter model version of BERT with significantly fewer parameters. It has the same general architecture as BERT. However, the token-type embeddings, as well as the pooler are removed, and the number of layers is reduced by a factor of 2 (Sanh u. a., 2020). The team has identified that the number of layers has the comparably largest impact on computation efficiency and hence, focused on optimizing this aspect.

3.5 Improvement 2 - Data augmentation via addition of synonyms

As per Wei und Zou (2020), we randomly chose n words from the retrieved sentences that were not stop words (a defined list of common words such as “the" and “and" that don’t contribute much to the sentence's meaning) nor proper nouns (considered as words starting with a capital letter) and replaced each of these words with one of its synonyms chosen at random from WordNet, a lexical database for English. The replacement process took about two days.

3.6 Improvement 3 - data augmentation via back-translation

Based on the Python 3 library "TextAugment", we imported the "Translate" function, which used Google's translation API to translate retrieved sentences first from English to German, French, Japanese, Hindi and Russian and second, back to English. We aimed to achieve similar improvements to Longpre u. a. (2019) and Yu u. a. (2018) by generating context paraphrases via back-translation. Back-translation was applied to 2% of the sentences in the dataset due to computational/time limitations. However in general, even when using a neural machine translation model (NMT) instead of an Internet-based NMT like Google's API, translation is very slow, on the order of seconds per sentence. This is because each target word requires looping over all source words for the attention calculation, and for NMTs that use recurrent neural networks, the target words can only be generated sequentially rather than in parallel; further, the use of large vocabularies exacerbates the slow speed as it results in expensive softmax normalization computations (Zhang u. a., 2020).

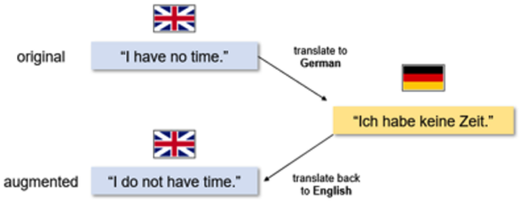


Figure 3: Back-translation example

3.7 Improvements 4 and 5 - BERT trained over one and three epochs respectively

An epoch refers to the process of passing an entire dataset forward and backward through a neural network once. A dataset is usually passed multiple times or in multiple mini-batches through the same neural network because optimizing the model's weights ("learning") is an iterative approach via gradient descent or stochastic gradient descent. The challenge is to find the optimal number of epochs that neither results in an underfitting, nor overfitting model (appendix A.4). As our baseline BERT model trains over two epochs, we experimented with one epoch and three epochs, respectively.

3.8 Improvement 6 - aggregating instead of using if-then logic

Baseline BERT uses if-then logic to classify a claim based on the five provided possible evidence sentences without taking advantage of any synergistic information between them. If there is any sentence that SUPPORTS, then the prediction is SUPPORTS; otherwise if there is any sentence that REFUTES, then the prediction is REFUTES; otherwise the label is NOT ENOUGH INFO.

We replaced this classification process with a neural aggregation step as per Yoneda u. a. (2019) to combine the knowledge of our retrieved sentences using a neural network as a more powerful architecture to learn any important relationships between input sentences and labels. The aggregation layer is a classifier neural network, with cross-entropy as its loss function. Each retrieved input sentence is assigned a score for each of the three labels. Then, the neural network calculates a score for each of the three labels and chooses the one with the highest score as the label for the claim (fig. 4). As the input in the baseline model from Primarosa (2020) did not have the granularity of label scores like Yoneda u. a. (2019) did, we added code to make these scores available for use in the neural network's input.

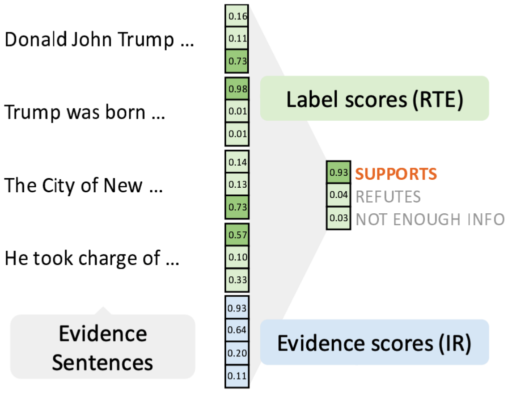


Figure 4: Overview of the aggregation network

4 Experiments

4.1 Data and code

The format of the data at the claim verification step for a single claim is illustrated in appendix A.5. For each claim, predicted sentences from Wikipedia were scored for their relevance to the claim, had a true (ground-truth) label for the training set (S = SUPPORTS, R = REFUTES, N = NOT ENOUGH INFO), and the baseline model's predicted label.

Baseline code, code changes and scripts to run are given in appendix A.6.

4.2 Evaluation method

We evaluated our models based on the following metrics:

- FEVER Score: the model's label accuracy conditioned on providing evidence sentences. The predicted evidence set needs to include a true evidence set for a high FEVER score.
- Label Accuracy: the model's accuracy to label correctly for "SUPPORTS", "REFUTES" or "NOT ENOUGH INFO".
- Evidence Precision: the macro-precision of the evidence for supported/refuted claims.
- Evidence Recall: the macro-recall of the evidence for supported/refuted claims where an instance is scored if and only if at least one complete evidence group is found.
- Evidence F1: harmonic mean of precision and recall.

4.3 Experimental details

The neural model configurations used are given in table 1.

Table 1: Neural model configurations

Model	Training Time (h)	Optimizer	# Parameters	# Training Epochs	Learning Rate
Baseline BERT	19	AdamW	∼110M	2	$2 \cdot 10^{-5}$*
DistilBERT	10	AdamW	∼66M	2	$2 \cdot 10^{-5}$*
Aggregator	0.1	Adam	24.6K	5	10^{-3} (default)

*Default learning rate from baseline model (Primarosa, 2020)

Training time refers to how long it took to run each model on a Tesla K80 GPU for fine-tuning to the FEVER dataset. Additionally, the aggregator is a classifier neural net with two hidden layers (100 neurons each) with a ReLU after each hidden layer. The input is of size 20 = 5 sentences x (1 sentence score + 3 class/label scores) and output size is 3 (3 class/label scores). Cross-entropy loss with class weights was used as the inverse of class dataset frequency. Training time quoted is only for this neural network rather than for BERT/DistilBERT and the aggregator combined.

4.4 Results

In table 2, we compare the evaluation metrics on the claim verification task for all seven experiments.

Table 2: Results

	FEVER score	Label accuracy	Evidence precision	Evidence recall	Evidence F1
BERT					
1 - Baseline	**0.6918**	**0.7415**	0.8906	0.7090	**0.7895**
2 - Data Augmentation (Synonyms)	0.689	0.7376	0.8921	0.7008	0.7849
3 - Data Augmentation (Back-Translation)	0.6872	0.7371	0.8915	0.7080	0.7892
4 - Training over One Epoch	0.6843	0.7345	**0.8952**	0.6938	0.7818
5 - Training over Three Epochs	0.6843	0.7374	0.8856	0.7056	0.7854
6 - Neural Aggregation	0.6864	0.7376	0.7262	**0.8405**	0.7792
DistilBERT					
7 - Baseline	0.5896	0.6415	0.8599	0.6420	**0.7351**
8 - Data Augmentation (Synonyms)	0.5859	0.6383	**0.8612**	0.6330	0.7297
9 - Data Augmentation (Back-Translation)	0.5849	0.6388	0.8552	0.6437	0.7346
10 - Neural Aggregation	**0.6081**	**0.6606**	0.6560	**0.8276**	0.7318

5 Discussion

Our experiments showed that modifications of the data augmentation and fine-tuning steps resulted in only minimal changes to the model's performance on key metrics. The most promising changes were the BERT training over one epoch and the addition of an aggregator, which resulted in better performance on evidence precision and evidence recall, respectively, compared to the baseline BERT model. High evidence precision is particularly relevant in the identification of misinformation on the Internet. We would prefer more diligence in selecting the evidence sentences that support a claim than letting incorrect evidence sentences "slip through" that could wrongly support claims and reduce the quality of our verification mechanism, which should be reliable and trustworthy. While our expectation was to see more significant improvements, we recognize that our ability to implement the modifications at a larger scale (e.g. architectural changes to the transformer, pre-training, or different pipeline steps) were limited by computational capacity. We would be curious to see which impact our suggestions could have when applied with more computational resources. Also, the reason for the small impact of our modifications could be that the input data and fine-tuning steps do have a small impact on the overall model performance, while the model architecture and pre-training process may be more meaningful and impactful. An ablation analysis on the different pipeline steps could provide additional insights.

In our baseline model, we used the large pre-trained BERT model and fine-tuned it on the smaller FEVER dataset for the claim verification task. Kou u. a. (2020) stated that this process often leads to the model being overfit on the smaller dataset. We were interested in combating this overfitting by augmenting the FEVER dataset. By adding more variance to the fine-tuning dataset, we hoped to make the BERT model more robust and generalizable. To see if the BERT model was actually overfitting on the FEVER dataset, we ran a simple check by adding an additional epoch (three instead of two) to the fine-tuning process. This resulted in a negligible decrease in performance, indicating that the BERT model was likely not underfitting to the FEVER dataset, but may be overfitting. To possibly reduce the phenomena of overfitting, we also fine-tuned the model with just one epoch, which also just resulted in a negligible decrease in performance in comparison to the baseline model.

Our data augmentation by synonym replacement and back-translation added more variance to the fine-tuning dataset. However, these experiments also did not noticeably change the model's performance in comparison to the baseline. This may be because BERT is simply too large (110M parameters) with a massive pre-trained learned knowledge base that our data augmentations to the small FEVER dataset did not have any impact on the model's performance. This may also be the case for why our experiment in using an aggregation layer, instead of the baseline if-then classifier, had negligible impact on performance - the BERT model being too large meant that the change applied to the classifier does not significantly impact its performance. Of all the experiments that we ran with the BERT model, none of them changed the FEVER score, label accuracy, or evidence F1 metrics by more than 0.01 except for the aggregation layer that decreased the evidence F1 score by 0.0103 compared to the baseline.

Next, we applied the synonym replacement and aggregation layer to the DistilBERT model (66M parameters) to evaluate if these changes would have a larger impact on a smaller model. Data augmentation with synonyms did not have a noticeable effect, but the aggregation layer did improve the FEVER score and label accuracy by 0.0185 and 0.0191 respectively, compared to the baseline DistilBERT model. This improvement may be due to DistilBERT being a much smaller model than BERT and so a change to the classifier, using an aggregator instead of simple if-then logic, has a larger impact on the model's performance.

6 Conclusion

As discussed previously, our modifications to fine-tuning did not result in significant model improvements, compared to baseline BERT. While we saw small improvements in evidence precision and recall with the epoch modification and aggregator approaches, baseline BERT still performed best on all other metrics. We recognize that augmentations to fine-tuning may only have a minimal impact on the overall model performance due to BERT's large size that contains a vast amount of knowledge, or the relatively small size of the tweaks made. Our modification experiments did have a larger impact on DistilBERT, given that with only 66M parameters the model is much smaller than BERT and is thus less prone to overfitting after fine-tuning. Also, our modification experiments were limited to claim verification. Assessing the impact of changes to the document and sentence retrieval steps could be an interesting area for future research.

For research teams with more computational resources, we would recommend 1) an ablation analysis across the three FEVER pipeline steps, 2) improvements to back-translation and 3) improvements to the final aggregation layer. Regarding an ablation analysis, this would help to identify which step has the largest impact on overall model performance, and thus where to direct focus to improvements. Regarding backtranslation, we suggest translating the entire dataset, rather than a subset of about 2% of sentences as we did due to computational limitations. We also suggest translating sentences using multiple languages per sentence to build up a training dataset with greater variation. Further improvements on our aggregation layer include training the aggregator to distinguish between where evidence is coming from by predicting the "not enough info" label when only a part of the evidence, and not the full evidence, is present. If such partial examples are included in training the aggregator, the models discriminative power should be higher. Additionally, to improve the FEVER score, which depends on the accuracy of the predicted evidence set, the aggregation network can be expanded to also output which retrieved sentences are the most likely to have contributed to the predicted label. Here, an RNN (recurrent neural network) could be used as an alternative model architecture with the initial input being the claim, and subsequent inputs and outputs being the sentences and sentence relevance, respectively. This architecture also has the flexibility to take as input a varying number of sentences.

A Appendices

A.1 Current FEVER challenge results

The FEVER challenge results at the time of this paper's submission (March 12th, 2021) are given in table 3.

Table 3: FEVER challenge results as of March 12th, 2021

#	User	Entries	Data of Last Entry	FEVER Score	Label Accuracy	Evidence F1
1	h2oloo	3	01/05/21	0.7587	0.7935	0.3955
2	nudt_nlp	17	08/29/20	0.7442	0.7738	0.3890
3	dominiks	6	07/09/20	0.7427	0.7660	0.3669

A.2 Parameter counts of recently released and pretrained language models

The parameter counts of some recent models are given in fig. 5.

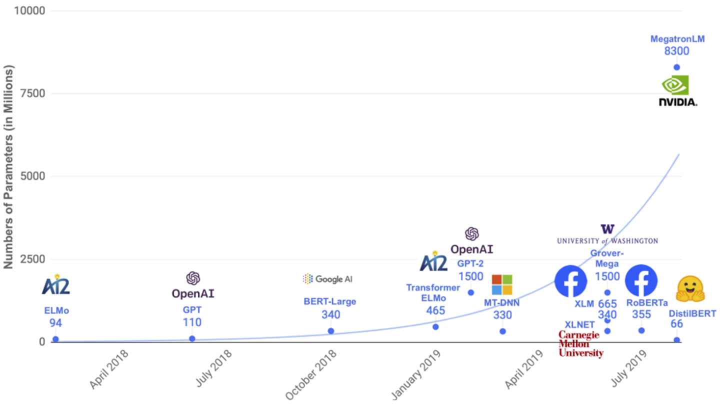
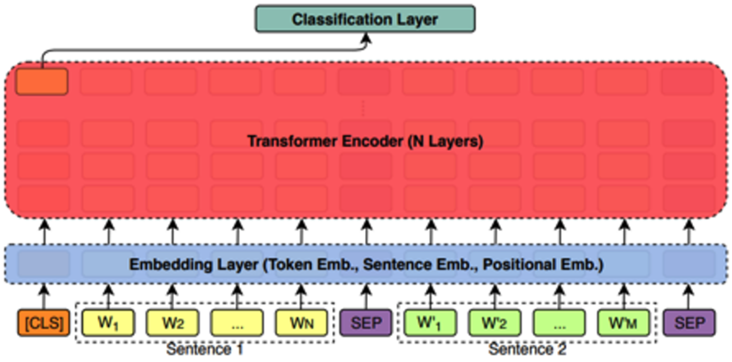


Figure 5: Parameter counts of recent models (Sanh u. a., 2020)

A.3 BERT model architecture



Classification Layer
Transformer Encoder (N Layers)
Embedding Layer (Token Emb., Sentence Emb., Positional Emb.)
[CLS]
W1
W2
...
WN
SEP
W'1
W'2
...
W'M
SEP
Sentence 1
Sentence 2

Figure 6: BERT model architecture (Soleimani u. a., 2019)

The BERT model architecture used is given in fig. 7. The input representation starts with a special classification embedding ([CLS]) and is followed by the tokens' representations of the first and second sentences, separated by another specific token ([SEP]). The model input of the form [CLS] + sentence 1 + [SEP] + sentence 2 is then passed through the embedding layer, where token, sentence, and positional embedding are applied, as well as through N transformer encoder layers. A classification layer predicts the output from the first neuron of the last layer.

A.4 Overfitting and underfitting

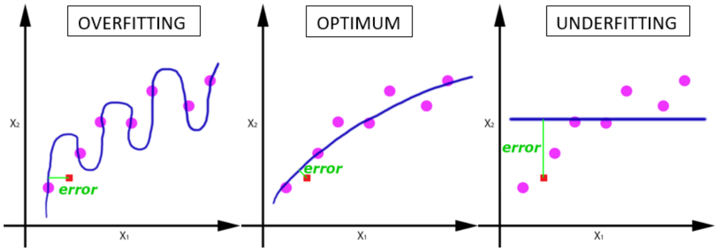


Figure 7: Impact of different epoch numbers on model results (Sharma, 2017)

A.5 Claim verification data format

The format of the data used at the claim verification step for a single claim is given in table 4.

Table 4: Data sample from FEVER for a single claim at the claim verification step

Claim ID	Claim	Page name	Sentence ID	Sentence	Sentence score	True label	Predicted label
75397	Nikolaj Coster-Waldau worked with the Fox Broadcasting Company.	Nikolaj_Coster-Waldau	7	He then played Detective John Amsterdam in the short-lived Fox television series New Amsterdam -LRB- 2008 -RRB- , as well as appearing as Frank Pike in the 2009 Fox television film Virtuality , originally intended as a pilot .	0.76	S	S
75397	Nikolaj Coster-Waldau worked with the Fox Broadcasting Company.	Fox_Broadcasting_Company	0	The Fox Broadcasting Company -LRB- often shortened to Fox and stylized as FOX -RRB- is an American English language commercial broadcast television network that is owned by the Fox Entertainment Group subsidiary of 21st Century Fox .	0.08	S	N
75397	Nikolaj Coster-Waldau worked with the Fox Broadcasting Company.	Nikolaj_Coster-Waldau	8	He became widely known to a broad audience for his current role as Ser Jaime Lannister , in the HBO series Game of Thrones .	0.56	N	N
75397	Nikolaj Coster-Waldau worked with the Fox Broadcasting Company.	Nikolaj_Coster-Waldau	9	In 2017 , he became one of the highest paid actors on television and earned # 2 million per episode of Game of Thrones .	0.33	N	N
75397	Nikolaj Coster-Waldau worked with the Fox Broadcasting Company.	Nikolaj_Coster-Waldau	3	Since then he has appeared in numerous films in his native Scandinavia and Europe in general , including Headhunters -LRB- 2011 -RRB- and A Thousand Times Good Night -LRB- 2013 -RRB- .	0.05	N	N

A.6 Codebase and scripts to run

The code for this paper is at `https://github.com/jonathan-ling/cs-224n-final-project`. Baseline code is given in table 5, with our changes to it in table 6, and scripts to run in table 7.

Table 5: Baseline code from existing papers

File / repository	Link	Associated paper
Baseline BERT	Primarosa (2020)	Readme file at Primarosa (2020)
Synonym replacement	https://github.com/jasonwei20/eda_nlp/blob/5d54d4369fa8db40b2cae7d490186c057d8697f8/experiments/nlp_aug.py	Wei und Zou (2020)
Aggregator for claim verification labelling	https://github.com/takuma-ynd/fever-uclmr-system/blob/interactive/neural_aggregator.py	Yoneda u. a. (2019)

Table 6: Files added to or created in a copy of the baseline model's repository

Experiments	File	Summary of changes made
Synonym replacement and back-translation	src/pipeline/claim-verification/generate.py	Added synonym replacement and back-translation code
Aggregator	src/pipeline/claim-verification/model.py	Added prediction scores for each class (refutes, supports, not enough information) for each retrieved sentence
	src/pipeline/claim-verification/aggregator.py	Created neural network model to aggregate claims to replace the original if-else model
Common to all	scripts/pipeline.sh	Set up experiments to be able to be run at the command line with appropriate flags

Table 7: Commands to run

Experiment	Commands
Baseline	`bash scripts/pipeline.sh claim_verification -model-type bert -model-name bert-base-cased`

Synonym replacement	`bash scripts/pipeline.sh replace_synonyms`
	`bash scripts/pipeline.sh claim_verification -model-type bert -model-name bert-base-cased`

Back-translation	`bash scripts/pipeline.sh backtranslation`
	`bash scripts/pipeline.sh claim_verification -model-type bert -model-name bert-base-cased`

Aggregation layer	`bash scripts/pipeline.sh claim_verification -model-type bert -model-name bert-base-cased`
	`bash scripts/pipeline.sh write_predictions -model-type bert -model-name bert-base-cased`
	`bash scripts/pipeline.sh aggregator`

References

[Cocarascu 2018] COCARASCU, Oana: *Fact Extraction and VERification (FEVER) Challenge*. 2018. – URL `https://competitions.codalab.org/competitions/18814#results`

[Devlin u. a. 2018] DEVLIN, Jacob ; CHANG, Ming-Wei ; LEE, Kenton ; TOUTANOVA, Kristina: Bert: Pre-training of deep bidirectional transformers for language understanding. In: *arXiv preprint arXiv:1810.04805* (2018)

[Gardner u. a. 2019] GARDNER, Matt ; GRUS, Joel ; NEUMANN, Mark ; TAFJORD, Oyvind ; DASIGI, Pradeep ; LIU, Nelson F. ; PETERS, Matthew ; SCHMITZ, Michael ; ZETTLEMOYER, Luke: AllenNLP: A Deep Semantic Natural Language Processing Platform, 2019

[Hanselowski u. a. 2018] HANSELOWSKI, Andreas ; ZHANG, Hao ; LI, Zile ; SOROKIN, Daniil ; SCHILLER, Benjamin ; SCHULZ, Claudia ; GUREVYCH, Iryna: Ukp-athene: Multi-sentence textual entailment for claim verification. In: *arXiv preprint arXiv:1809.01479* (2018)

[Kou u. a. 2020] KOU, Xiaoyu ; YANG, Yaming ; WANG, Yujing ; ZHANG, Ce ; CHEN, Yiren ; TONG, Yunhai ; ZHANG, Yan ; BAI, Jing: Improving BERT with Self-Supervised Attention. In: *arXiv preprint arXiv:2004.03808* (2020)

[Liu u. a. 2019] LIU, Yinhan ; OTT, Myle ; GOYAL, Naman ; DU, Jingfei ; JOSHI, Mandar ; CHEN, Danqi ; LEVY, Omer ; LEWIS, Mike ; ZETTLEMOYER, Luke ; STOYANOV, Veselin: Roberta: A robustly optimized bert pretraining approach. In: *arXiv preprint arXiv:1907.11692* (2019)

[Longpre u. a. 2019] LONGPRE, Shayne ; LU, Yi ; TU, Zhucheng ; DUBOIS, Chris: An Exploration of Data Augmentation and Sampling Techniques for Domain-Agnostic Question Answering, 2019

[Primarosa 2020] PRIMAROSA, Simone: *FEVER Transformers*. 2020. – URL `https://github.com/simonepri/fever-transformers`

[Sanh u. a. 2020] SANH, Victor ; DEBUT, Lysandre ; CHAUMOND, Julien ; WOLF, Thomas: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In: *arXiv preprint arXiv:1910.01108v4* (2020)

[Sharma 2017] SHARMA, Sagar: *Epoch vs Batch Size vs Iterations*. 2017. – URL `https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9`

[Soleimani u. a. 2019] SOLEIMANI, Amir ; MONZ, Christof ; WORRING, Marcel: BERT for Evidence Retrieval and Claim Verification. In: *arXiv preprint arXiv:1910.02655v1* (2019)

[Thorne u. a. 2018] THORNE, James ; VLACHOS, Andreas ; CHRISTODOULOPOULOS, Christos ; MITTAL, Arpit: FEVER: A large-scale dataset for fact extraction and verification, 2018

[Wei und Zou 2020] WEI, Jason ; ZOU, Kai: EDA: Easy data augmentation techniques for boosting performance on text classification tasks, 2020

[Yoneda u. a. 2019] YONEDA, Takuma ; MITCHELL, Jeff ; WELBL, Johannes ; STENETORP, Pontus ; RIEDEL, Sebastian: UCL Machine Reading Group: Four Factor Framework For Fact Finding (HexaF), 2019

[Yu u. a. 2018] YU, Adams W. ; DOHAN, David ; LUONG, Minh T. ; ZHAO, Rui ; CHEN, Kai ; NOROUZI, Mohammad ; LE, Quoc V.: QaNet: Combining local convolution with global self-attention for reading comprehension, 2018

[Zhang u. a. 2020] ZHANG, Wen ; HUANG, Liang ; FENG, Yang ; SHEN, Lei ; LIU, Qun: Speeding up neural machine translation decoding by cube pruning, 2020