CS224N | Final Project

1. KEY INFORMATION:

- Title: **Six Approaches to Improve BERT for Claim Verification as Applied to the Fact Extraction and Verification Challenge (FEVER) Dataset**
- Team: Daniel Jun, Jonathan Ling, Anica Oesterle {djun36, jonling, oestea}@stanford.edu
- Project Type: Custom
- Mentor: Megan Leszczynski
- Other information: no external collaborators; no external mentors; not a shared project

2. ABSTRACT:

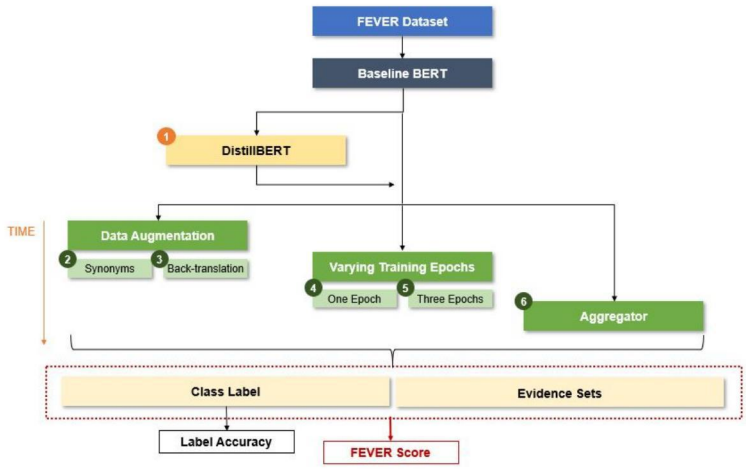
BERT has been used in various research for fact extraction and verification tasks, such as tweet classification, hate speech detection and fake news detection. However, BERT suffers from various issues when applied to claim verification, which can help detect and classify misinformation. The goal of 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.

3. 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 et al., 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 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 BERT** 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 et al., 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 et al. (2018) to combine this knowledge. Our contributions include:

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



4. RELATED WORK:

Both **Soleimani et. al. (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 et al. (2018)** 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 et al. (2019), as well as Longpre et al. (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 et al., as well as Yu et al. (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 et al., 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 et al. in 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 2). Given the FEVER task's extremely high computational requirements, DistilBERT was a good fit for our improvement experiments.

5. APPROACH:

5.1 | 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 (Figure 1). "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.

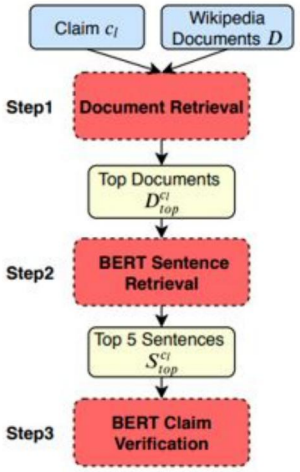


Figure 1: three-step pipeline evidence extraction and claim verification (Soleimani et al., 2019)

5.2 | Baseline Model - BERT: The FEVER dataset provides N_D Wikipedia documents: $D = \{d_i : i = 1, ..., N_D\}$. A document d_i consist of sentences $S^{d_i} = \{s_j{}^i : j = 1, ..., N_{S^{d_i}}\}$. The model's goal is two-fold: first, it has to classify the claim c_l for $I = 1, ..., N_C$ (N_C = 145,000 for the FEVER benchmark) as "SUPPORTS", "REFUTES" or "NOT ENOUGH INFO". Second, to consider a prediction true, a complete set of evidence $E_{c_l} = \{s_{ij}\}$ has to be retrieved from claim c_l. Claims labelled with "not enough info" do not have an evidence set.

Document Retrieval: for this task we use code from Hanselowski et. al. (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 (Gardner et al., 2017) constituency parser 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 et al. (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 et al. (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 $D^{c_l}{}_{top}$ for claim c_l.

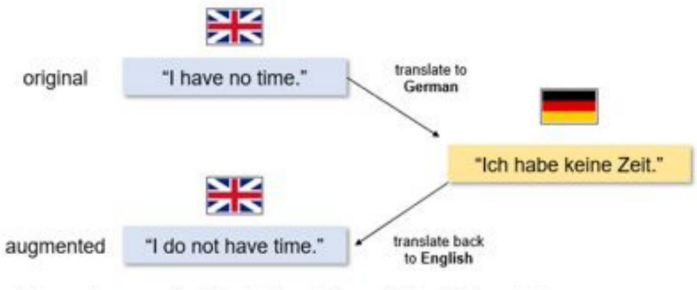
Sentence Retrieval and Claim Verification: here, we use code from Primarosa (2020), with both of these steps using a BERT model each. The architecture of the BERT model follows that of Soleimani et al. (2019) and is illustrated in Appendix 3. In the **sentence retrieval** step, for each claim c_l, all sentences S_{d_i} retrieved from the documents ($D^{c_l}{}_{top}$) in the document retrieval step that match the claim c_l ($S^{c_l}{}_{all} = \{S_{d_i} | \, d_i \in D^{c_l}{}_{top}\}$) are scored, and the top five potential evidence sentences $S^{c_l}{}_{top}$ by this sentence score are retrieved. The training set consists of ~145,000 claims for which this is done. Here, $S^{c_l}{}_{all}$ 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 $S^{c_l}{}_{top}$ 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).

5.3 | 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 et al., 2018) due to its large size, text-only claims (no metadata), and live public leaderboard (Cocarascu, O., 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.

5.4 | 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 et al, 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.

5.5 | Improvement 2: Data Augmentation via Addition of Synonyms: As per Wei et al. (2019), 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 c. two days.

5.6 | 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 et al. (2019) and Yu et al. (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 et. al., 2018).

5.7 | Improvement 4: BERT Training over One Epoch + Improvement 5: BERT Training over Three Epochs: "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 4). As our baseline BERT model trains over two epochs, we experimented with one epoch and three epochs, respectively.

5.8 | 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 et al. (2018) 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 (Figure 2). As the input in the baseline model from Primarosa (2020) did not have the granularity of label scores like Yoneda (2018) did, we added code to make these scores available for use in the neural network's input.

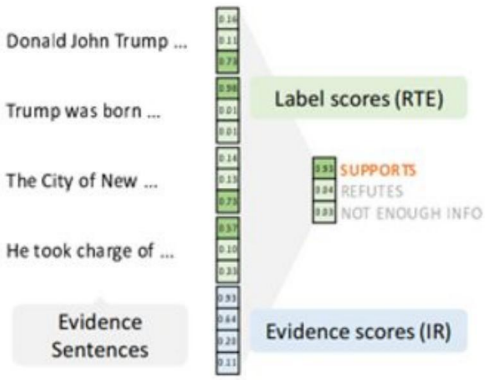


Figure 2: overview of the aggregation network (Yoneda et al., 2018)

5.9 | Codebase: we used the following code / files from existing papers:

File / repository	Link	Associated paper
Baseline BERT	https://github.com/simonepri/fever-transformers	(No paper - see readme file at the link)
Synonym replacement	https://github.com/jasonwei20/eda_nlp/blob/5d54d4369fa8db40b2cae7d490186c057d8697f8/experiments/nlp_aug.py	'EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks' (https://arxiv.org/pdf/1901.11196.pdf)
Aggregator for claim verification labelling	https://github.com/takuma-ynd/fever-uclmr-system/blob/interactive/neural_aggregator.py	'Four Factor Framework For Fact Finding (HexaF), '(https://www.aclweb.org/anthology/W18-5515.pdf)

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

Experiments	File	Summary of changes made
Synonym replacem. & 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

5.10 | Scripts to run to reproduce results: instead of running this line in the original instructions (Primarosa, 2020): "bash scripts/pipeline.sh claim_verification --model-type bert --model-name bert-base-cased", run the following commands instead:

Experiment	Commands
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

6. EXPERIMENTS:

6.1 | Data: the format of the data at the claim verification step for a single claim is illustrated in Appendix 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.

6.2 | 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.

The table compares these metrics on the claim verification task for all seven experiments:

	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

6.3 | Experimental Details: the following neural models with the following specifications were used.

Model	Training Time (h)	Optimizer	# Parameters	# Training Epochs	Learning Rate
Baseline BERT	19	AdamW	~110M	2	2e-5*
DistilBERT	10	AdamW	~66M	2	2e-5*
Aggregator	0.1	Adam	24.6K	5	1e-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.

6.4 | Results: 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.

7. ANALYSIS:

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 (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.

8. 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 backtralsation, 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 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.

9. REFERENCES:

Cocarascu, O. (2018). Fact Extraction and VERification (FEVER) Challenge. *CodaLab - Competition.* https://competitions.codalab.org/competitions/18814#results

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Hanselowski, A., Zhang, H., Li, Z., Sorokin, D., Schiller, B., Schulz, C., & Gurevych, I. (2018). UKP-Athene: Multi-Sentence Textual Entailment for Claim Verification. *arXiv preprint arXiv:1809.01479.*

Kou, X., Yang, Y., Wang, Y., Zhang, C., Chen, Y., Tong, Y., ... & Bai, J. (2020). Improving BERT with Self-Supervised Attention. *arXiv preprint arXiv:2004.03808*.

Li, Z., Ding, X., Liu, T. (2019). Story Ending Prediction by Transferable BERT. *arXiv preprint arXiv:1905.07504v2*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Longpre, S., Lu, Y., Tu, Z., DuBois, C. (2019). An Exploration of Data Augmentation and Sampling Techniques for Domain-Agnostic Question Answering. *arXiv preprint arXiv:1912.02145v1.*

Primarosa, S. (2020). FEVER Transformers. *GitHub. https://github.com/simonepri/fever-transformers*

Sanh, V., Debut, L., Chaumond, J., Wolf, T. (2020). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108v4.*

Sefara, J. (2020). TextAugment: Improving Short Text Classification through Global Augmentation Methods. *GitHub: https://github.com/dsfsi/textaugment*

Soleimani, A., Monz, C., and Worring, M. (2019). BERT for Evidence Retrieval and Claim Verification. *arXiv preprint arXiv:1910.02655v1.*

Thorne, J., Vlachos, A., Christodoulopoulos, C., & Mittal, A. (2018). Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355.*

Wei, J., Zou, K. (2019). EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. *arXiv preprint arXiv:1901.11196v2.*

Yoneda, T., Mitchell, J., Welbl, J., Stenetorp, P., Riedel, S. (2018). UCL Machine Reading Group: Four Factor Framework For Fact Finding (HexaF). *Association for Computational Linguistics. https://www.aclweb.org/anthology/W18-5515.pdf.*
GitHub: https://github.com/takuma-ynd/fever-uclmr-system/blob/interactive/README.md

Yu, A., Dohan, D., Luong, M. (2018). Qanet: Combining Local Convolution With Global Self-attention For Reading Comprehension. *arXiv preprint arXiv:1804.09541v1.*

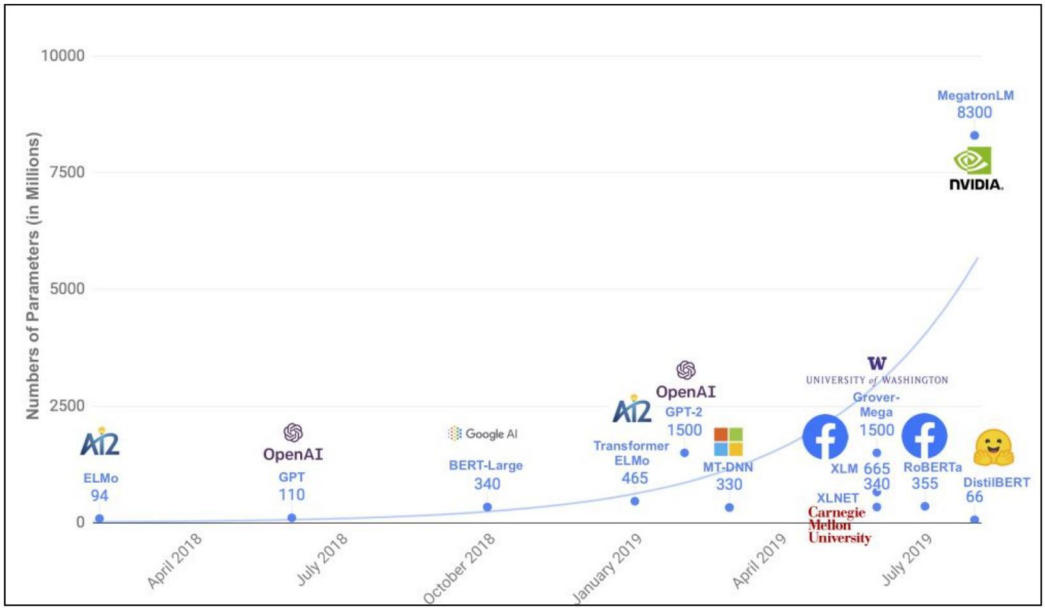
Zhang, W., Huang, L., Feng, Y., Shen, L., & Liu, Q. (2018). Speeding up neural machine translation decoding by cube pruning. *arXiv preprint arXiv:1809.02992.*

10. APPENDIX:

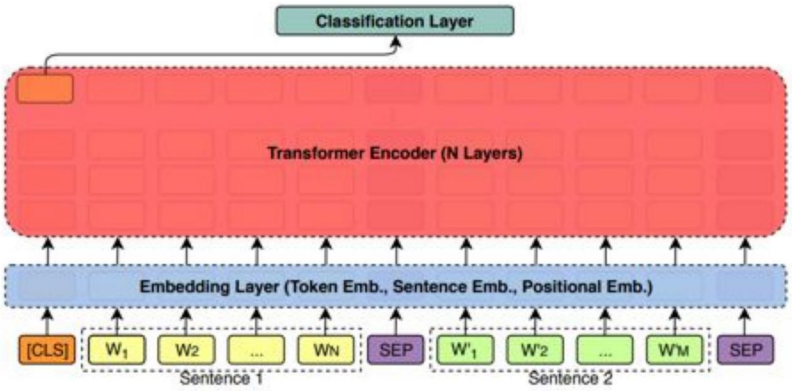
Appendix 1: current FEVER challenge results (as of March 12th, 2021)

#	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

Appendix 2: parameter counts of recently released and pretrained language models (Sanh et al., 2020)

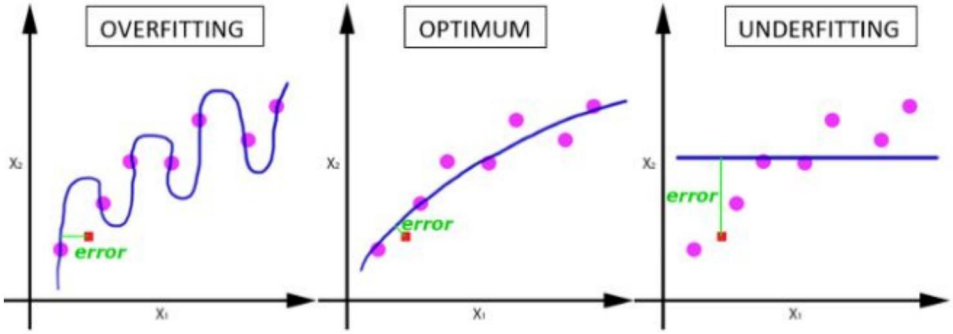


Appendix 3: BERT model architecture (illustration from Soleimani et al., 2019)



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.

Appendix 4: impact of different epoch numbers on model results (Sharma, 2017)



Appendix 5: format of the data at the claim verification step for a single claim

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